Keyword Querying and Ranking in Databases

Surajit Chaudhuri
Microsoft Research
Redmond, WA 98052
+1-425-703-1938
surajitc@microsoft.com

Gautam Das
Dept of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, USA 76019
+1-817-272-7595
gdas@uta.edu

1. INTRODUCTION

With the proliferation of data sources exposed through web interfaces to consumers, simple ways of exploring contents of such databases are of increasing importance. Examples include users wishing to search catalogs of homes, cars, cameras, restaurants, and photographs. One approach that has been explored is to allow users to query such databases in the same ways as they explore web documents. Thus, it is desirable to be able to use the paradigm of keyword querying and automated result ranking over contents of databases. However, the rich relationships and schema information present in databases makes a direct adaptation of information retrieval techniques inappropriate. This problem has attracted much attention in research as it presents a rich set of challenges from defining semantics of such querying model to developing algorithms that ensure adequate performance. In this tutorial, we focus on the highlights of research progress in this field.

2. OUTLINE OF TUTORIAL

The following is a brief outline of the topics to be covered.

2.1 Overview

The tutorial will begin by arguing why adaptation of querying paradigms from information retrieval is attractive for database systems. We focus on two key challenges in adapting such a querying paradigm:

Mapping of keyword queries to SQL queries: This step requires translating the keyword query into a set of candidate SQL queries taking into account the content as well as the schema of the database systems.

Automated Result Ranking: This is the task of automatically determining an order to the result tuples of any SQL query identified in the previous step. This assists users to effectively browse large sets of returned results by helping them focus on the most relevant tuples.

Together, appropriate solutions to these two problems provide an information retrieval style ad-hoc search and retrieval system for databases.

VLDB '09, August 24-28, 2009, Lyon, France.

2.2 Challenges

2.2.1 Semantics Challenges

Unlike the classical "bag of words" IR data model, the data model for structured databases is more complex, consisting of data spread across multiple tables connected by key and foreign keys. Thus, answer to a keyword query may require result tuples to be assembled via joins. Likewise, direct adaptations of automated ranking functions from IR, such as TF-IDF, may not apply, as correlations often exist among data elements. Hence naïve adaptation of the "bag of words" model is a poor fit for structured databases. This part of the lecture will bring home some of the novel challenges of supporting exploratory search interface over databases.

2.2.2 Query Processing Challenges

A direct adaptation of query processing techniques from information retrieval to databases is challenging as well. Most IR query answering engines rely on the *inverted list* index structure. In contrast, any keyword querying and ranking implementations on databases have to co-exist and leverage B+ and full-text search (FTS) indexes available in SQL database systems.

2.3 Semantics

We will briefly review the semantics of fundamental IR ranking functions, in particular TF-IDF, Probabilistic IR models, and Language models and then discuss the semantic challenges in keyword querying and automated ranking in databases.

2.3.1 Keyword Querying Semantics

There have been various proposals for solutions for keyword querying in databases. We will discuss motivating scenarios for different semantics and the different formal frameworks they build upon. Our discussions shall cover early systems such as DBXPLORER, BANKS, DISCOVER, as well as more recent proposals such as BLINKS, EKSO, SPARK, KITE. In our tutorial, we will analyze how these systems differ semantically from each other as well as from traditional IR keyword search engines. Although not the key focus of this tutorial, we briefly cover keyword querying systems for relational data streams and semi-structured data.

2.3.2 Ranking Functions Semantics

We then motivate the need for automatic ranking in databases by discussing the *empty answers* and *many answers* phenomena. While some ranking functions are specifically appropriate for keyword querying systems, most of our discussions apply to automated ranking functions that are broadly applicable to all SQL selection queries. We study different ranking functions in

detail, including the minimal-sized-tree based ranking functions, the authority-based ranking functions, the Vector Space and Probabilistic IR inspired models. We compare and contrast these ranking functions from the semantic aspect, and specifically discuss how the nature of structured data impacted their formulation.

2.4 Query Processing

We begin by discussing the core evaluation technique for keyword queries in information retrieval based on inverted lists. We then discuss how today's database systems allow text columns integrated as part of their SQL querying environment (Full Text Search Indexes), and point out their key differences from traditional B+ indexes. Finally, we will review the essence of database algorithms for Top-K queries such as the Threshold Algorithm (TA). We then discuss in details how recent research has tried to leverage these access methods to offer efficient implementation of keyword querying and ranking in databases, as further detailed below.

2.4.1 Keyword Querying

The techniques for efficient keyword querying in database systems can be broadly classified as follows:

Graph Based Systems: In these systems (e.g., BANKS) the database is transformed into an instance level data graph where edges indicate the various ways in which tuples (vertices) are related.

SQL Query Based Systems: In this approach, each keyword query is translated into a set of SQL queries. Our discussion shall include early systems such as DBXPLORER, DISCOVER, and more recent systems.

Composite Systems: These systems attempt to leverage advantages of both graph based and SQL based approaches, e.g., the ESKO system.

We will compare and contrast the query processing techniques employed for each of the three approaches outlined above. Their relationship to formal problems (e.g., Steiner tree, Group Steiner trees) will be analyzed. Their dependence on any underlying query engines (e.g., for SQL or SPARQL) will be discussed.

2.4.2 Automated Ranking

In this part of the tutorial, we will survey query processing implications for various ranking schemes that have been proposed for keyword querying systems, ranging from those based on short join sequences (or compact join trees) to those based on PageRank style authority transfer techniques. In many of these proposals, a key ingredient has been TA-style Top-K algorithms that support early termination. Several of the automated ranking proposals also rely on Full Text Search (FTS) support in databases. Implications of using TA and using FTS in database systems for automated ranking will be critically examined. If time permits, we will briefly comment on keyword query processing in semi-structured data and relational data streams, and point out novel challenges posed by these scenarios for query processing.

2.5 Conclusions

We will emphasize how the challenging problems of keyword querying and ranking in databases leverage information retrieval, traditional relational query processing, as well as more recent innovations in database algorithms. We will conclude by identifying open problems in supporting keyword search and ranked retrieval over database systems.

3. TARGET AUDIENCE

Researchers in the area of database systems, information retrieval, data mining, database algorithms, and web services will benefit from this cross-disciplinary tutorial.

4. BIOGRAPHICAL SKETCH

Surajit Chaudhuri is a Principal Researcher and a Research Area Manager at Microsoft Research, Redmond. He has worked in the areas of query optimization, physical database design, data cleaning, and text search. He is an ACM Fellow. He was awarded the ACM SIGMOD Contributions award in 2004 and a 10 year VLDB Best paper Award in 2007.

Gautam Das is an Associate Professor at the Computer Science and Engineering department of the University of Texas at Arlington. His research interests span data mining, information retrieval, databases, algorithms and computational geometry. His research has been supported by grants from NSF, ONR, Microsoft Research, and Nokia Research.