Using Document Space For Relational Search

Richard Drake
Faculty of Science
University of Ontario Inst of Technology

Ken Q. Pu
Faculty of Science
University of Ontario Inst of Technology

***Abstract*—In this paper, we present a family of methods and algorithms to efficiently integrate text indexing and keyword search from information retrieval to support search in relational databases. We propose a bi-directional transformation that maps relational database instances to document collections. The transformation is shown to be a homomorphism of keyword search. Thus, any search of tuple networks by a keyword query can be efficiently executed as a search for documents, and vice versa. By this construction, we demonstrate that indexing and search technologies developed for documents can naturally be reduced and integrated into relational database systems.**

I. Motivation

Information retrieval has been an active and fruitful field of research since 1960's. With seminal work by [1] and [2], the IR community has laid the foundation of automatic text indexing and keyword query processing of text documents. The technology for document indexing continues to gain momentum with the growing presence of text data found on the Web and in social media. For instance, new techniques by [3] and [4] improve on the traditional similarity measures by incorporating further (NLP) on the context of phrases and words.

In the last decade, there has been a tremendous interest from the database community to support keyword search queries for structured relational databases. Systems such as *Discover* [5], *DBxplorer* [6] and *BANKS* [7] and many others [8], [9] model relational tuples as documents, and foreign key joins as links. Thus, it's possible to derive IR-style scoring function for document similarity. More recently, semantic information [10], schema and meta data [11] have been incorporated into the search algorithm.

where $\text{SEARCH}_{\text{DB}}$ and $\text{SEARCH}_{\text{DOC}}$ are the search functions for relational databases and documents respectively.

Practical experiences have demonstrated that the state-of-the-art $\text{SEARCH}_{\text{DOC}}$ has more performant implementations ([12], [13]) compared to its relational database counter part. Our interest is to construct efficient transformations h and g such that relational search $\text{SEARCH}_{\text{DB}}$ can be efficiently implemented as a composition of $g \circ \text{SEARCH}_{\text{DOC}} \circ h$, effectively taking advantage of the document search technology.

II. Search problems for relational and document models

In this section, we present the formal definition of relational databases and collections of documents. We also formalize the notion of keyword search entity graphs and join networks of entity graphs in relational databases.

A. Relational entities

A relational database consists of a collection of tables which are interconnected via linkages.

A table, T, has a number of attributes:

$$\text{attr}(T)$$

A tuple in a table $r \in T$ is defined as a mapping from attributes to values: