Abstract—Keyword search is the most effective information discovery method in documents. The large volume of data is stored in databases. Plain text coexists with structured data, unstructured data for this type of data efficient processing of top-k queries is a crucial requirement. This paper describes fundamental characteristics including relational database, top-k queries, and Steiner trees. Recently, Tuple units are used to improve the keyword search by joining the multiple related tuple units and indexes are used for structural relationships. In this paper various existing techniques for developing search system are described in the literature survey. This survey also describes the Ranking. Ranking queries are dominant in many emerging applications for finding top-k answers. The research strategy used to resolve is top-k query processing.

Keywords—Keyword Search, Top-k Query Processing, Relational Databases, Tuple Units

I. INTRODUCTION

Data mining is the process that attempts to discover patterns in large data sets. The methods used by it are at the intersection of machine learning, statistics, artificial intelligence and database systems. In general the aim of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.

The relational databases are commonly searched using structured query languages. The internet user wants to get the correct answers quickly and efficiently. The users are not willing to browse through the complete answer set. Keyword search is an extensively accepted mechanism for querying in textual document systems and World Wide Web. The database research community has recently recognized the benefits of keyword search and has been introducing keyword search capability into relational databases [11], [8], [3], [4], XML databases [5], graph databases [3], [6], [7], [8] and heterogeneous data sources [9].

The advantages of the keyword search on databases are: It is easy to use and enables the interesting or unexpected discoveries to access the data in web and scientific applications. That is relevant data are scattered but are rightly relevant to query should be automatically assembles in the result.

II. FUNDAMENTAL CONCEPTS

In this section we try to provide the understanding of some basic notions of the keyword search.

A. Relational Database

The relational model is the most widely used data model and a vast majority of current database systems are based on this model. In general a Relational Database Management Systems (RDBMS) provides users with an abstract view of the underlying data. In many approaches, a search system which enables keyword search in relational databases is implemented on top of the RDBMS [18].

B. Keyword Search

The term Keyword Search is that a user submits a query using a finite set of keyword to find the data that satisfies his/her information needs [1].

C. Top-k Query

The meaning of term query is a finite set of keywords. To make clear the meaning of top-k query define as- Given a database D of m objects, each of which is to describe the character by n attributes, a scoring function f, in agreement with to which we rank the object the database D. Then a top-k query Q returns the k objects with the highest rank in f efficiently. Instead of all answers, a top-k query returns the subset of most related answers [1], [33].
D. Tuple Units

Given a database with m connected tables, the tuples (records) that can be integrated together through the primary/foreign keys must be very relevant with each other. Based on observation Li.et.al [1], [14]. Tuple units are efficient to answer keyword queries and can represent meaningful and integral information units [1]. The database is modeled as a directed graph. Tuples → nodes. Foreign-key, Primary-key link → edge. Result is a rooted directed tree containing at least one node having each query keyword.

E. Steiner-tree-based search

A relational database can be modeled as a database graph $G = (V, E)$ such that there is a one-to-one mapping between a tuple in the database and a node in $V$. $G$ supposed to be as a directed graph with two a edge: a forward edge $(u, v) \in E$ if and only if there is a foreign key from $u$ to $v$, and a back edge $(v, u)$ if and only if $(u, v)$ is a forward edge in $E$. An edge $(u, v)$ suggests a close relationship between tuples $u$ and $v$ [26], [16], [6].

Most of existing methods of keyword search over relational databases find the Steiner trees composed of relevant tuples in the process of answers. The Steiner trees are identified by discovering the rich structural relationships between tuples, and ignore the fact that such structural relationships can be pre-computed and indexed. Tuple units that are composed of most applicable tuples are proposed to address this problem. Tuple units can be computed and indexed [1].

III. EFFICIENT PROCESSING

In this section the terminologies or the research dimensions are described as follows

A. Top-k Query Processing Techniques

The taxonomy to classify top-k query processing techniques based on many design dimensions are shown in fig. 1 [33]

Top-k processing in domains such as the web, multimedia search and distributed systems has shown a great impact on performance [33]. From this classification we discuss only the Ranking function (scoring).

A. Ranking

Ranking function returns a ranking value for each row in a partition depending on the function that is used. Ranking functions are nondeterministic in nature. Several ranking functions are used to rank the tuple units. The ranking is important for keyword search and it is the simplest way to score and rank virtual documents and tuples is by the use of traditional TF*IDF metric to score the tuple units or the documents. Tuple (Document) frequency and length can also be obtained.

To access a group of tuples as an answer, the ranking function includes many operations like aggregation, grouping, choosing to these operations in the search model will complicate the problems related to the existing techniques.[1]

B. Indexing

Indexing is the term which reduces the scan time and also reduces the computation time. Indexes are used to efficiently search the related answers. To structure the answer queries and also for finding the structural relationship between tuples that is keyword structure aware index for single keyword and also for multiple keywords. In which the inverted files which has a attribute that points to a list of documents [1], [17], [15].

IV. LITERATURE SURVEY/RELATED WORK

A number of previous researchers have proposed systems for the keyword search in relational databases using various approaches.
I) Keyword Searching and Browsing in Databases Using

This is a system which enables keyword based search on the relational databases altogether with data and schema browsing. It describes a backward edge to enable the search system backward; an answer to a query is a subgraph by connecting the nodes matching the keywords. BANKS recognizes the connected trees say, Steiner trees, in a labeled graph by using an approximation to the Steiner tree problem, which is proved to be an NP-Complete problem. Finding the most relevant answer is alike to searching for a minimum group steiner tree having all of the query keywords. [2]

II) DBXplorer: A System for Keyword-Based Search over Relational Databases By S. Agrawal, S. Chaudhuri

In this system DBXplorer, an efficient and scalable keyword search utility for relational databases, is described. DBXplorer returns all rows either from single tables, or by joining tables connected by foreign-key joins such that the each row includes all keywords. It also provides an indexing structure, symbol table which find out the exact position of query keywords in the relational entities. Query answers are constructed through breadth-first enumeration and then join each join tree is mapped. The join trees are ranked by the number of joins. [26]

III) Finding Approximating Top-k Answers in Keyword Proximity Search By B. Kimelfeld and Y. Sagiv

Proximity is an appropriate measure for the relevance of different data objects when a specific query is not provided. When pair of keyword queries are as a input, Find set and Near set, are expressed, the system finds objects specified in the Find set that have the shortest path connecting them to objects specified in the Near set. The purpose of this system is of textual proximity in IR is applied to an arbitrary database which measures some pairs of objects are closely related. The objects are related by a distance function and the score of an object is computed using its proximity to all objects. For efficient distance computation, all pair’s distances are recomputed in a k-neighbourhoods distance lookup table [16].

IV) Discover: Keyword Search in Relational Databases and Efficient Ir-Style Keyword Search over Relational Databases By V. Hristidis and Y. Papakonstantinou

This system produces a minimum number of efficient SQL queries, Candidate Networks, which find all of the connections between the tuples that contain the query keywords.

Candidate networks are built as total minimal joining networks, which are trees of tuples. DISCOVER II assigns each tuple into the result tree and finds the individual score after this it combines the scores of all tuples in the result tree. The top-k processing technique, avoids the creation of all the queries by discarding low-scoring candidate networks in advance is done by using the Pipelined algorithm. [3] [10]

V) Objectrank: Authority-Based Keyword Search in Databases By A. Balmin, V. Hristidis, and Y. Papakonstantinou.

ObjectRank system applies a qualified PageRank algorithm to keyword search in a database for which there is a natural flow of authority between the objects. It considers a schema graph as an authority flow reference and uses a data graph for the results. The authority potentially flows in both directions. Instead of the tuple trees as the answers this system results individual tuples. Global ObjectRank is query-independent and is obtained by placing all of the nodes of the data graph in the base set. The InverseObjectRank [27] measure is a that thing which rests on the assumption that specific results should be ranked higher than general ones with topics that span across many topics. This InverseObjectRank solves the problem by exploiting the specificity of a node. [27]

VI) Spark: Top-k Keyword Query in Relational Databases
By Y. Luo, X. Lin, W. Wang, and X. Zhou

This SPARK system propose three algorithms, skyline sweeping, block pipelining and tree pipeline, which provide efficient query processing framework based on the new ranking method. Efficiency and effectiveness of answering top-k keyword query in RDBMS is also described. They proposed a ranking formula by accommodating IR techniques based on the concept of virtual documents. This system reduces the number of database accesses by using monotonic, upper bound function for the ranking and the Skyline sweeping algorithm minimizes the amount of join checking. [11]
In this paper we also focused on the various research dimensions and its approaches which are used to improve processing of the keyword search in relational databases.

REFERENCES


