Keyword-based Search over Databases: A Roadmap for a Reference Architecture Paired with an Evaluation Framework

Sonia Bergamaschi[1], Nicola Ferro[2], Francesco Guerra[1], and Gianmaria Silvello[2]

[1] Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Italy
{sonia.bergamaschi,francesco.guerra}@unimore.it
[2] Department of Information Engineering, University of Padua, Italy
{ferro,silvello}@dei.unipd.it

Abstract. Structured data sources promise to be the next driver of a significant socio-economic impact for both people and companies. Nevertheless, accessing them through formal languages, such as SQL or SPARQL, can become cumbersome and frustrating for end-users. To overcome this issue, keyword search in databases is becoming the technology of choice, even if it suffers from efficiency and effectiveness problems that prevent it from being adopted at Web scale.
In this paper, we motivate the need for a reference architecture for keyword search in databases to favor the development of scalable and effective components, also borrowing methods from neighbor fields, such as information retrieval and natural language processing. Moreover, we point out the need for a companion evaluation framework, able to assess the efficiency and the effectiveness of such new systems and in the light of real and compelling use cases.

1 Introduction

Since the last decade, we have been observing a continuous increment of structured data available in the Web. In a first stage, structured data was only indirectly available as "embedded" in Web pages. This is the case of data driven Web applications, where user information needs are constrained by pre-defined Web forms which limit the range of the queries that can be performed in favour of a simpler and more intuitive interaction. Based on the user inputs, the Web application retrieves and publishes data extracted from a private (i.e., not directly accessible by the external users) source. The collection of all these sources constitutes the so-called Deep or hidden Web, terms denoting its inaccessible nature. Therefore, the Web application constitutes the entry point for selecting and filtering the access to the data which are then available only by manually filling-up Web forms on application-specific search interfaces [43]. Moreover, tables directly inserted in Web pages have been usually adopted for publishing structured data on the Web. Several studies [15] tried to estimate the dimension of the information contained in such tables and to develop techniques for their automatic discovering, extraction and re-use as autonomous structured data sources. This legacy represents still an open issue when it comes to unleashing the full potential of such data sources by allowing users to seamlessly access them in a fashion not tied to pre-defined paths.

More recently, data have been recognized as an extremely valuable asset also from the socio-economical point-of-view; the Economist magazine recently wrote that "data is the new raw material of business" and the European Commission stated that "Big data technology and services are expected to grow worldwide to USD 16.9 billion in 2015 at a compound annual growth rate of 40% – about seven times that of the information and communications technology (ICT) market overall" [23]. The principal driver of this evolution is the Web of Data, the size of which is estimated to have exceeded 100 billion facts (i.e. semantically connected entities). The actual paradigm realizing the Web of Data is the Linked (Open) Data [36], which by exploiting Web technologies allows public data in machine-readable formats to be opened up ready for consumption and reuse. In this emerging scenario, huge quantities of structured data are published on the Web and they are readily available to end-users for direct consumption. Furthermore, advanced services (e.g. Web and mobile applications) are increasingly making use of these data by exploiting the outcomes achieved in the semantic Web and Linked Data research fields.

In both cases described above, the tasks of finding data sources well-suited for specific information needs and selecting relevant data within a given data source are crucial. In the following, we will focus on relational databases which are the key and most widespread structured data sources baking the above mentioned scenarios, but we make our analysis general enough to be used also with other sources.

Keyword search is the foremost approach for information searching and it has been extensively studied in the field of Information Retrieval (IR) [14]. Nevertheless, retrieving information from (unstructured or semi-structured) documents is intrinsically different from querying databases, and consequently this model has left out the structured data sources which are typically accessed through structured queries, e.g. Structured Query Language (SQL) queries over relational databases or SPARQL Protocol and RDF Query Language (SPARQL) queries over Linked Data graphs.

Structured queries are not end-user oriented, given that their formulation is based on a quite complex syntax and requires some knowledge about the structure of the data to be queried. Furthermore, structured queries are issued assuming that a correct specification of the user information need exists and that answers are perfect – i.e. they follow the "exact match" search paradigm. On the other hand, end-users are more oriented towards a "best match" search paradigm given that their information needs are often vague and subjected to a progressive and gradual process of refinement enabled by the search activity itself [7]. Indeed, according to Marchionini [45], information seeking activities can be aimed at lookup (e.g. fact checking), learn (e.g. comprehend a phenomena, developing new knowledge) and investigate (e.g. support planning and forecasting) where learning and investigative searching require strong human involvement in a continuous and interactive way. As a consequence, search can also be seen as a multi-stage activity that helps the user to clarify her/his information needs and to subsequently tune her/his queries for finding better suited results.

In the last fifteen years, these facts triggered the research community to put a lot of effort in developing new approaches for keyword search over structured databases [17, 18, 54]. Nevertheless, despite of the research work, there are not yet keyword search prototypes able to scale up to industrial-grade applications. Two main issues are hampering the design and development of next generation systems for keyword search over structured data: (i) the lack of systemic approaches considering all of the issues of keyword search from the interpretation of the user needs, to the computation, retrieval, ranking and presentation of the results; and (ii) the absence of a shared and complete evaluation methodology measuring user satisfaction, achieved utility and required effort for carrying out informative tasks.

In particular, with respect to the first issue, we claim that a conceptual architecture pivoting around keyword search and structured data needs to couple system- and user-oriented components. The former ones aim at augmenting the performances (i.e. efficiency) of the search, whereas the latter ones aim at improving the quality of the search (i.e. effectiveness) from the user perspective. Such system- and user-oriented components already exist and have been demonstrated to be effective for their specific purpose. Nevertheless, their integration into a unique framework for keyword search is still lacking. Moreover, it should be considered that a search process is part of a wider user task [38] from which the information need arises and, in turn, makes the user resort to issuing queries to satisfy it. This makes the whole process quite complex and brings in the accomplishment of the user information need several degrees of uncertainty. The main focus of keyword search, i.e. getting out the most from the relational data starting from keywords instead of a structured query, is certainly something that helps users in carrying out their tasks. Nevertheless, many factors, beyond algorithmic correctness and completeness, may impair the impact of keyword search and prevent users to fully exploit its potentialities. Indeed, as also discussed above, even if a database is designed for answering exact queries, the search process and the user intents are usually defined in a vague way which calls for best match approaches and for ranking the results of a query (even exact) by the estimation of how much they may fit the actual information needs of the user. Therefore, we need to put keyword search into a broader context and envision innovative architectures where keyword search is one of the components, paired with other building blocks to better take into account the variability and uncertainty entailed by the whole search process.

Concerning the second issue, experimental evaluation [32] – both laboratory and interactive – estimates how much systems are adherent to the user information needs, provides the desired effectiveness and efficiency, guarantees the required robustness and reliability and operates with the necessary scalability. Measuring these abilities provides insights about the features of a system thus indirectly becoming a key means for supporting and fostering the development of new systems with improved performances.

In light of this, we claim that the current frameworks for the evaluation of keyword search in relational databases [20] need to be re-thought, by moving beyond the evaluation of keyword search components in isolation or not related to the actual user needs, and, instead, by considering the whole system, its constituents, and their inter-relations with the ultimate goal of supporting actual user search tasks [9, 25]. Furthermore, we outline some guidelines for defining a fair and complete evaluation of keyword search approaches building on the well-established IR evaluation methodologies and tuning it for taking into account the intrinsic peculiarities of search over structured data.

The paper is organized as follows: Section 2 discusses the common approaches for querying structured data, proposes a conceptual architecture for keyword-based search systems and presents a use case. Section 3 points out the main limits of current benchmarks for evaluating keyword search systems over structured data and Section 4 outlines some guidelines for defining a fair and complete evaluation methodology. Finally, Section 5 reports final remarks and future works.

2 Querying structured data

Issues related to keyword search over documents have been addressed by the IR community since its inception, back in the mid of last century. A generic IR system comprises three main components [14]: an index and a retrieval and ranking model. The typical search process starts by considering the user's *information need* from which a query composed of one or more keywords (i.e. *terms*) is then derived. Although search is typically based on simple keywords, IR systems usually support also richer syntax allowing for the use of boolean and pattern matching operators, compound terms, phrases, alternative weighting of terms, and so on. A major task of an IR system is to build and maintain the *index* which is a data structure containing the terms in the documents associated with the locations where they appear; for enhancing the search process IR systems also maintain statistics about the indexed documents such as the number of occurrences of a term in a document and in the whole collection. By exploiting the index, the statistics and other available data, the system processes the user query (i.e. parse it into terms to be matched against the index) and by using a retrieval model – e.g. boolean model, vector space model, language model – returns a list of documents ranked accordingly to their estimated relevance to the user need. Finally, evaluation is conducted off line and it is aimed at measuring the effectiveness of an IR system for further understanding its functioning in a real environment and giving insights about how it can be improved. As stated in [14], in order to be successfully carry out, IR evaluation must have: (i) a characterization of the system purpose; (ii) a measure that quantifies how well this purpose is met; and, (iii) an accurate and economical measurement technique.

The problem of ranking documents (i.e. assign *scores*) accordingly to the user's query is one of the most studied problems in the field.

IR can be considered now as a mature technology and its research activity outcomes are implemented in commercial systems such as web search engines and domain-specific IR systems (e.g. desktop search, patent search and medical search). Furthermore, IR can count on a well-defined and shared specification of the main components of a search system, a thorough and fair evaluation methodology and a cooperative research community addressing specific aspects of IR systems – e.g. information needs modeling, indexing, retrieval and ranking and results presentation – while maintaining the common aim of improving search efficiency and effectiveness.

The research in keyword search over relational databases is still young and far from providing similar outcomes. In particular, the approaches proposed in the literature typically focus on developing some functionalities without envisioning a holistic solution. A study and a reference architecture defining required functional components and interfaces among them is still missing. As witnessed by the foremost role of the ANSI-SPARC model [51] in the development of DataBase Management System (DBMS),

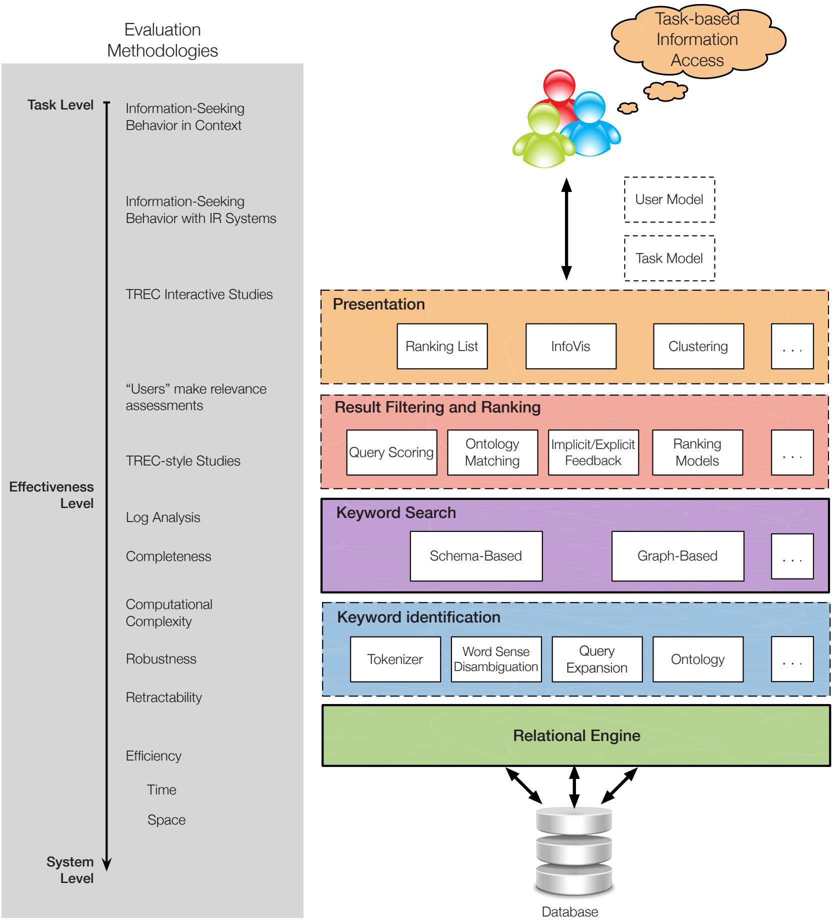


Fig. 1. Reference Architecture and Evaluation Methodologies for a Keyword-based Search System.

the specification of a standard architecture can provide several advantages, making it possible to define best practices as well as to enable interoperability among different component implementations and a fair evaluation process.

Leveraging on the base of knowledge provided by the IR and DBMS areas, we envisaged possible reference architecture for an information access system pivoting on keyword search techniques, as shown in Figure 1. Thick solid lines frame modules which are the focus of current keyword search systems, whereas dotted lines frame modules which are typically not exploited today and come from neighboring fields, such as information retrieval, information extraction, data mining, and natural language processing. The main keyword search and relational database layers are surrounded by a Keyword Identification layer, a Results Filtering and Ranking layer, and a Presentation layer. Furthermore, for each constituting module of the reference architecture, Figure

1 shows examples of possib le components and technologies that could be adopted for implementing it. Indeed, an ideal search system has to consider the search task a user desires to conduct, to perform user queries knowing that they may not exactly correspond to the real user information need, to disambiguate search terms, to rank the results of the search process on the basis of relevance for the user, and to visualize these results in the most proper way for the considered search task.

2.1 Understanding the User Input: the Keyword Identification module

The final goal of the *Keyword Identification* layer is to understand what the user had in mind when s/he formulated the query. This task requires the development of techniques for understanding the data structures involved in the query and their correlations. Such layer is not required in the current DBMSs which are typically queried by a structured query language, e.g. SQL for relational DB. In these systems, the user is in charge of specifying in his/her query which are the data structures containing the data of interest, and how these structures are linked with each other for the computation of an answer. This way, users can formulate accurate and exact queries. Nevertheless, the expressive power of the structured query languages comes with a price: it requires users to be expert, since they have to know the syntax of the language and the data source structures. This limitation makes this kind of language not suitable for users who do not know the language adopted or the source structures storing the data.

A querying system based on keywords does not require the knowledge of any language and data structure, thus allowing people to formulate queries in an easier way. Nevertheless, keyword queries are inherently ambiguous: the same keyword may refer to terms belonging to different database structures. This problem is also common to IR systems, where similarly the same keyword may be used with different meanings in the same / different document(s). In relational databases, this ambiguity generates more issues, mainly due to the fragmentation of the information in different tables. In case of multiple keywords, for example, we have not only to deal with the uncertainty about the meaning of each term, but also about the paths connecting the selected data structures to be followed. The choice of one path instead of another can generate different results since they are based on different interpretations of the intended meaning of the keyword query (see Section 2.4).

Recently, some formal keyword-based languages have been proposed to combine simplicity and intuitiveness of expression with the richness of more expressive languages, such as the Contextual Query Language (CQL). Nevertheless, these languages are not commonly adopted and their expressiveness is still low compared what the SQL language allows users to formulate in databases.

The last difference between IR systems and keyword search over structured databases is that in IR user keywords are directly retrieved in some specific indexes over the data, whereas in the so-called schema-based keyword search systems the keywords provide the information for formulating structured queries, but they are not directly used by the system for retrieving data.

It is evident that the Keyword Identification layer relies on different components for managing the different approaches.

Summarizing, the Keyword Identification is a strategic layer in keyword search over structured database, since it is in charge of understanding what the user had in mind when formulating the query.

The final goal is freeing keyword search from an exact match with the keywords present in the relational data and introducing the possibility of matching in multiple ways the keywords expressed by user in order to compensate for possible imprecisions or errors in the choice of the keywords.

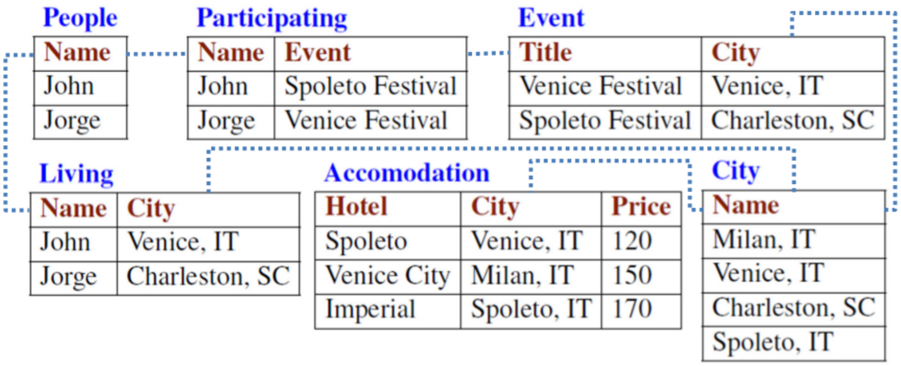
2.2 The Business Logic: Keyword search, Result Filter, and Rank

The *Keyword search* layer aims at matching the user keywords (or their interpretations) with data structures and domains of selected attributes. Keyword search aims at retrieving the database tuples matching the user keywords (or their interpretations). Two main techniques are typically adopted [54]: graph-based and schema-based. Graph-based techniques (e.g., BANKS [1], BLINKS [34], PRECIS [48], DPBF [22] and STAR [40]) model relational databases as graphs, where nodes are tuples, edges foreign-primary key relationships between those tuples. Their main aim is to optimize the computation of specific structures over the graphs (e.g., Steiner trees, rooted trees, etc.) to find the most relevant top-k connected tuples. Their challenge is to handle the large and complex graphs induced by the database instance, as it could make the problem hardly tractable. Schema-based techniques (e.g., DISCOVER [37], DBXplorer [4], SPARK [44], and SQAK [50]) exploit the schema information to formulate SQL queries determined starting from the user keyword queries. In this case, the system has to discover the structures containing the keywords and how these structures may be joined in order to formulate a set of queries capturing the intended meaning of the user keyword query, expressed in the native structured query language of the source. Most of the existing approaches rely on indexes and functions over the data values to select the most prominent tuples as results of queries.

Recently metadata-based approaches have been developed [12, 8, 11]. These approaches are useful when there is not direct access to the database instance or when frequent updates make the process of building and updating indexes too expensive.

Finally, some hybrid approaches, combining the features of graph and schema-based systems have been proposed. Among them, QUEST [10] couples by means of a probabilistic framework a schema-based approach for matching keywords with data structures and domains (thus retrieving solutions according to the "user perspective", i.e., to what the user had in mind when s/he formulated the query), and a graph-based approach to connect the tables that better represent the meaning of the user query (thus retrieving solutions according to the "database perspective", i.e., to the way data is stored in the database).

The *Results Filtering and Ranking* layer accounts for the need of adopting alternative strategies for ranking and selecting the results to be presented to the user by the system. This layer plays a fundamental role when the "exact match" search paradigm is left in favor of a "best match" approach where the results returned to the user are not always "correct", but need to be ranked on the basis of the relevance to the user information need. It may concern weighting the results on the basis of the process which



People

Name
John
Jorge

Participating

Name	Event
John	Spoleto Festival
Jorge	Venice Festival

Event

Title	City
Venice Festival	Venice, IT
Spoleto Festival	Charleston, SC

Living

Name	City
John	Venice, IT
Jorge	Charleston, SC

Accomodation

Hotel	City	Price
Spoleto	Venice, IT	120
Venice City	Milan, IT	150
Imperial	Spoleto, IT	170

City

Name
Milan, IT
Venice, IT
Charleston, SC
Spoleto, IT

Fig. 2. A fraction of a database schema with its data.

generated relational queries from user keywords, or relying on implicit/explicit feedback from the user to filter out some results, or using rank aggregation and data fusion techniques to merge alternative ranking strategies. In some of the approaches this layer is embedded and joint with the keyword search layer.

2.3 Presentation of the Results

The *Presentation* layer regards how the outputs of a system are presented to the user; for instance, we can have traditional ranking lists, results presentation based on advanced information visualization techniques and user interfaces [35], presentations of clusters of results.

The presentation, conversely to what happens in IR, is not standardized. In IR, it is clear that users are querying over a set of documents and they are expecting as output a ranked list of them on the basis of their information needs. In databases, most of the keyword search approaches work over a single database and not over a set of databases (the granularity of the result is different). In this case, it is not clear which answer should be returned to the user. It could be a boolean value (i.e., the database contains the user keywords), or tuples with different level of granularity (from few columns to the universal tuple). The definition of which is the best “dimension” of the results for a specific query is still an open challenge in current systems.

2.4 Use Case

To better illustrate the need for a reference architecture, let us introduce a simple relational database containing tourism information about people, events planned in some cities and possible accommodations for tourists. In Figure 2, we report a fraction of the database schema with some sample data. Some descriptive attributes are defined for each table and some foreign key - primary key relationship (represented with dashed lines) exists between the tables.

As mentioned, some of the issues addressed by keyword search systems in databases are common to IR systems searching in documents. The process for tokenizing keywords, or recognizing synonym terms, for example, as most of the components included in the Keyword Identification layer in Figure 1 can indeed be the same in both the scenarios given that the approaches share the same goal: satisfying the user information need. Nevertheless, relational databases differ from documents because they comprise a schema. This has two major implications: (i) the schema can provide useful information for solving the query; and (ii) the schema elements can be used for augmenting the search space given that keywords can match also with schema elements (i.e. metadata) and not only with the data contained in the database.

In the following, we consider queries composed of two keywords that we assume to be conjunct by a logical AND operator; we can point-out three possible matches between the keywords and databases:

- *data-oriented matches*: both the keywords match into data instances;
- *schema-oriented matches*: both the keywords match into schema elements;
- *mixed matches*: a keyword matches into a schema element, the other into a data instance.

Data-oriented matches. Let us consider the keyword query "Spoleto Charleston": the process for retrieving its possible answers can be schematized in two main steps.

In the *first step*, an index-based approach can find several matches for the keyword "Spoleto" with instances of the table `Accommodation` (i.e., Spoleto is the name of a `Hotel`, or Spoleto is a value of the attribute `City` in the same table) with an instance of the table `Event` (i.e Spoleto is contained in the title of an event) or with an instance of the table `City` (i.e Spoleto is the name of the city). This is due to the fact that the user query is ambiguous and it is unclear whether the user is looking for hotels called "Spoleto", hotels which are in "Spoleto", or the event "Spoleto Festival".

All the three answers are valid and possibly relevant to the user that issued such a query; indeed, one of the tasks of the search engine is to rank them by the estimated relevance to the user information need. The very same scenario can be outlined by considering the second keyword – i.e. "Charleston" – which can match with an element of the table `City` or of the table `Event`.

In the *second step*, the possible paths – that need to be formalized into structured queries when working with schema-based approaches – connecting the elements identified in the previous step have to be computed and ranked. For example, if we consider a keyword query "John Charleston", we can individuate two possible paths linking the tables `People` and `City`: one through the table `Living`, meaning that the user is interested in some people called "John" living in "Charleston" and the second through the tables `Participating` and `Event`, meaning that the user is interested in some people called "John" participating in an event organized in "Charleston".

This problem can be even more complex if some keywords match with foreign key - primary key values. This occurs in our example if the user formulates the query "Spoleto", where it is unspecified if the user is interested in accommodations or cities. From an algorithmic perspective an answer showing all the hotels in Venice provides an answer as good as the one that reports all the information about the city. Nevertheless, from the user perspective, only one of these two sets of results is relevant to her/his information needs, e.g. the city one, while providing both of them would reduce the effectiveness of the system and hampers performance.

Finally, note that our assumption that keywords are connected with AND operators can have not trivial implications in databases, in particular when both the keywords match with the same attribute domain. One of the possible answers of the query *"Spoleto Charleston"* matches both the keywords into elements of the table `City`. In this case, the simplest interpretation of the query – i.e. "the user is looking for cities called Spoleto and Charleston" – would return no results because of the first Normal Form in databases which imposes atomic values in the attribute domains. Possible not empty interpretations of the query are "people living in Spoleto and Charleston", or "people participating in events organized in Spoleto and in Charleston" that requires, in schema-based approaches, the non trivial operation of understanding it and the consequent formulation of complex join SQL queries involving other tables.

Schema-oriented matches. Several existing keyword search approaches do not consider database metadata as possible targets for user queries and consequently they cannot solve schema-oriented keywords. In these cases, the simple query "Hotel Price" does not find any result. Otherwise, a simple index built on the schema elements can provide the results given that the table `Accommodation` has the attributes `Hotel` and `Price`.

Mixed matches. Let us consider the query "Hotel Venice", where one keyword refers to a schema element and the second to a data element. Firstly, an index-based approach can find matches between the keyword "Venice" and several possible instances: it could match occurrences of the table `Accommodation` (i.e., Venice is the name of a `Hotel`, or Venice is a value of the attribute `City`), of the table `Event` (i.e Venice is the `title` of an event), and of the table `City` (i.e Venice is the `name` of the city). The user query is ambiguous and it is unclear whether the user is looking for hotels called "Venice", hotels which are in "Venice", or hotel related in some way to the event "Venice Festival". All the listed answers are possible, it is one of the tasks of the search engine to rank them. Moreover, if an index built on the schema element exists (i.e., the system considers database metadata as possible targets for user queries), the second keyword "Hotel" can be exploited to rank the answers. For example, some heuristic rule can prioritize the associations of keywords in the same tables (according with the idea that a query search for something that is "close" in the database representation) and consequently provide higher rank to the results where Venice is an accommodation than the ones where Venice is an event. Note that some keywords can match with schema and data elements at the same time, as it occurs for the keyword "City". In this case, multiple combinations are possible.

Since users do not know the information in the data source, this situation frequently occurs and is typically not managed by most of the existing systems.

A classical keyword search engine may not find any correspondences between the keyword "Hotel" and an element in the database. In this case, the problem is two-fold: firstly, the chosen keyword may better match metadata instead of a value in a table (as in the case in Figure) – indeed, several existing keyword search engines do not consider database metadata as possible targets for user queries; secondly, analogously to what happens IR systems, the chosen keyword may actually refer to a concept represented in the table with a value which is a synonym of the chosen keyword.

Even in this toy example, it becomes clear that query ambiguity as well as the choice between graph or schema based techniques impacts the system performance. Therefore, the complexity of real scenarios may take even more advantage from the architecture proposed in Figure 1, which complements keyword search with additional components that, in this specific example, analyze the user keywords and disambiguate their meaning. Similarly, the evaluation methodology must be able to detect these different issues in order to properly assess how systems tackle them.

Finally, as a further example, it is not clear from any of the queries proposed which is the information that the user would like to receive as a result. Several options are possible: in some cases the user would like to receive all the data about the tuples satisfying the criteria defined by the keyword query (i.e., the "universal relation"), in other cases only the values of some attributes (i.e., a projection of the "universal relation"), or even a boolean value (i.e., the existence of at least an instance). On the other hand, she/he also would like to receive results ordered by the system estimation of their relevance and how much they satisfy her/his information need. The task of understanding the granularity, the ordering and aggregation of the expected results is not typically managed by the existing systems and requires a specific module in the architecture of a complete keyword search system. As above, evaluation must be able to take into account these aspects and assess the systems accordingly.

3 Evaluating keyword-based search approaches for structured data sources

Innovative proposals for pushing the boundaries of keyword search cannot set aside a proper and shared evaluation methodology which helps in progressing towards the envisioned goals, ensures the soundness and quality of the proposed solutions, and guarantees the repeatability and comparability of the experiments.

As shown on the left hand side of Figure 1, evaluation can be carried out at three levels: at a "task level" for instance by means of user studies [42]; at an "effectiveness level" by means of the test collection methodology [32]; and, at a "system level" by means of benchmarking queries per second, memory and CPU load, correctness and completeness [20], thus progressively moving from a human to a system focus.

Nevertheless, experimental evaluation is hampered by fragmentation – different tasks, different collections, different perspectives from interactive to laboratory evaluation which are usually dealt with in separated ways, without sharing resources and the produced data [53]. This will be even more true for the multidisciplinary approach to the system architecture proposed in Figure 1; for this reason, a unified and holistic approach to evaluation would be needed to assess the different facets of such a complex system and to reconcile the experimental outcomes.

To the best of our knowledge, only few papers addressing issues related to the evaluation of keyword search systems over relational databases have been published in the literature. In [52], it is observed that the existing keyword search approaches have been evaluated against different databases with different query sets. This fact prevents their direct comparison based on their original experimental results, since cross-collection comparisons are often not feasible or, at least, extremely difficult. Moreover, in some cases, the evaluation framework adopted appears to be inadequate, mainly due to the employment of a small number of self-authored queries, thus leading to biased results. Only recently a benchmark [20] proposed some metrics and a query set for evaluating the approaches against three data sources (Mondial, IMDB and Wikipedia). Even if it represents an important step towards a fair evaluation of keyword search approaches, the benchmark suffers from some limitations. Firstly, the metrics adopted (precision and recall compared to a gold standard, and time needed for returning the results) cannot be suitable when applied to a schema-based keyword search system, which transforms keyword queries into SQL queries. In this case, the evaluation of the time performance is biased by the time required for the execution of the SQL queries by the DBMS underlying the application. Different underlying DBMSs can largely influence the time performance, but they are typically independent of the keyword search technique under evaluation. Secondly, the benchmark computes the effectiveness of the approaches by analyzing the results (instances) retrieved with specific keyword queries whereas schema-based search approaches provide SQL queries as a primary result. Note that all the tuples resulting from the same SQL query have intrinsically the same score, and that the same result can be obtained by different queries. Thirdly, most of the queries in the dataset are composed of only one element. In several approaches the evaluation of this kind of queries is performed only by analyzing the indexes over the data implemented in the DBMS. Even in this case, the evaluation task risks to evaluate the DBMS storing the data instead of the keyword search technique. Fourthly, the benchmark does not discuss what is a “correct” result in terms of granularity. It would be useful to discuss if the system has to retrieve the “universal tuple” or a subset of it.

4 Towards a holistic evaluation approach

A systematic and comparable experimental evaluation is a very demanding activity, in terms of both time and effort needed to perform it. For this reason, in the IR field, it is usually carried out in publicly open and large-scale evaluation campaigns at international level, which allow for sharing the effort, producing large experimental collections, and comparing state-of-the-art systems and algorithms. Relevant and long-lived examples are Text REtrieval Conference (TREC)[3] [33] in the United States, the Conference and Labs of the Evaluation Forum (CLEF) initiative[4] [26] in Europe, and NII Testbeds and Community for Information access Research (NTCIR)[5] in Japan and Asia. All these international evaluation activities rely on the Cranfield methodology [19] – i.e. the *de-facto* standard for experimental evaluation of IR systems – which makes use of shared experimental collections expressed as a triple composed by a *dataset*, a set of *topics*, which simulates actual user information needs and the *ground-truth* or the set

[3] http://trec.nist.gov/

[4] http://www.clef-initiative.eu/

[5] http://research.nii.ac.jp/ntcir/

relevance judgments, i.e. a kind of "correct" answers, where for each topic the data relevant for the topic are determined.

This well-established methodology and these international experiences would help in moving evaluation of keyword search in databases forward and fostering the development of next generation systems in this field. Indeed, by sharing resources and providing open fora to compare and discuss approaches, they ease the design of shared evaluation tasks, geared into concrete and compelling use cases as those discussed in the previous sections, which drive the design and development of next-generation systems [2]. Evaluation campaigns support the systematic exploration and deep understanding of the system behaviour via failure analysis [5, 30, 31], especially when systems are challenged by realistic datasets and compelling use cases. Moreover, they stimulate the creation of multidisciplinary communities, embracing all the competencies needed to embody the reference architecture of Figure 1. Finally, they foster the re-use of the experimental data and the acquired knowledge [3], giving the possibility to conduct longitudinal studies to track the improvement on performances [6, 27], to reproduce the obtained results [28] and lowering the barriers for comparing to and progressing beyond the state-of-the-art [21].

Moreover, as reported by [46], for every $1 that NIST and its partners invested in TREC, at least $3.35 to $5.07 in benefits accrued to researchers and industry. During their life-span, large-scale evaluation campaigns have produced huge amounts of scientific data which are extremely valuable for research and development but also from an economic point of view: [46] estimates that the overall investment in TREC of about 30 million dollars in its first 20 years which, as discussed above, produced an estimated return on investment between 90 and 150 million dollars. Therefore, applying experimental evaluation to vision represented in Figure 1 gives the promise not only to advance state-of-the-art techniques, but also to have a concrete economic impact.

Therefore, the evaluation of keyword search approaches assumes a paramount importance. In Section 3, some criticisms of the current evaluation frameworks have been highlighted. In the following, we try to go beyond them, by introducing and discussing some guidelines for the definition of an environment for experimenting the keyword search proposals.

Let us focus on scenarios where the search engine is coupled with a single database: this is notthere are only few proposals in literature of systems that rank databases on the basis of user keywords).

First of all, it is necessary to define the kind of evaluation we need to conduct because there is a clear distinction between evaluating the *formal correctness* of an algorithm implemented by a system and its *effectiveness* in addressing a search task. In the former case we are evaluating the formal *correctness* of the results returned by an algorithm or a system. The evaluation process assumes the existence of exact results and checks if, given certain inputs (e.g. a query), the actual outputs (e.g. a set of tuples) are the expected ones. Most of the evaluation conducted on keyword-based systems working over structured data has been based on the correctness of the results: in this case, the main difference between algorithms is given by their execution times (i.e. their efficiency) or whether they complete at all, during to their computational complexity. In the latter case, we are considering a *functional* and *holistic* evaluation process that checks if the system being tested is well-suited for addressing a given search task. In this case, we cannot assume the existence of formal correct answers and we are not evaluating the correctness of the implementation of the employed algorithms, but their effectiveness with respect to the search task the system has to address. This switch from correctness to effectiveness should also be accompanied by the adoption of proper evaluation measures [47], capable of grasping different aspects such as the utility delivered to the user [39], the impact of the time the user spends in interacting with the system [49], the effort required to the user [29], or the different ways in which the user scans the result list [24].

Overall, we believe that this evaluation process can lead to substantial improvement of keyword systems over structured data as it happened for document-oriented IR systems. Therefore, we propose an evaluation framework based on the Cranfield methodology which has to be tailored for keyword search over structured data; in the following, we outline the main components (dataset, topics, ground-truth) of this methodology and the issues to be faced.

The dataset. In general, the dataset has to be representative of the domain of interest both in terms of kinds of data and size. For example, if we consider the case cultural heritage search we need to use actual cultural heritage data as provided by The Library of Congress or Europeana and their size should be close to the actual one – e.g. for search tasks over Europeana, a dataset size of about 100 GB is considered a reasonable choice. This means that the kind of data and its size cannot be fixed a-priori, but have to be decided on the basis of the search task the system has to address.

From the keyword search point-of-view, the dataset must also have a complex structure made of inter-connected tables, with multiple paths joining the same tables, because retrieving data from flat datasets is mainly an index-based process and data indexes are provided by the DBMS. Thus, using a flat dataset would lead to the evaluation of the DBMS and not of the algorithm performing keyword search.

The topics. One of the main limitations of the current evaluation frameworks is that they propose a set of specific queries to be solved and not a set of information needs from which to derive the queries. This could make the system under evaluation prone to "overfitting" – i.e. the developers may try to adapt their systems with the main goal of optimizing the systems for the specific benchmark queries and not for addressing a general task. In this way, we may evaluate a system specifically tailored on the benchmark, which will not perform accordingly on a different data and query set.

Our idea is to base the evaluation on a set of information needs (i.e. the topics) in the form of short descriptions of what a user is looking for. The set of topics simulates actual user information needs and they could be prepared from real system logs, gathered by means of task-based analysis, or through a deep interaction with the involved stakeholders. As a consequence, the evaluation is conducted starting from the information need and not from ready-to-use query. For instance, a possible information need behind query "Hotel Venice" described in Section 2.4 could be "which are the hotels in Venice?".

Moreover, it is necessary to define information needs that can be translated into queries composed of more than one keyword because queries composed of only one term do not test the business logic behind the keyword search system, but only the data indexes, which typically are provided by the DBMS. With reference to the use case presented above, a query like "Venice" requires only to look up a term in a database index. A different situation occurs when keywords match tables connected by multiple paths, where it is not clear which one among all possible paths is the best match for the user information need.

The ground-truth. When we evaluate the efficacy of a system, the correctness of the returned results is measured in terms of relevance with respect to user information need. For this reason the possible results to a query have to be judged by a pool of domain users that decide if a result is *relevant* for a given information need – i.e. we have to form the set relevance judgments or the ground-truth against which the system output is evaluated. Note that the relevance judgments can be binary, i.e., relevant or not relevant, or multi-graded, e.g., highly relevant, partially relevant, not relevant and so on [41].

For keyword search systems, the ground-truth creation activity is related to the definition of the expected results from a user keyword query.

A schema-based approach generates (a set of) SQL queries, and they could be a good candidate for the evaluation. Nevertheless, to consider SQL queries as result of a search system makes graph-based approaches (that directly retrieve the data) not comparable with schema-based approaches. Vice versa, even though we consider the tuples returned by the queries, the comparison with graph-based approaches is not straightforward since all the values resulting from the same SQL query have the same rank. For example, the "Hotel Venice" query in schema-based approaches provides a list of hotels, all with the same rank. On the contrary, in graph-based approaches, in principle, each hotel in the list can have a different rank.

However, if we consider tuples as the natural result of user queries we have to define its boundaries. For instance, let us consider the query "Hotel Venice" of the use case, where the user is looking for the hotels located in Venice. A possible result is constituted only by the names of the Hotels (the values of the attribute `Hotel` in table `Accommodation`). Another valid result is the entire tuple composed of the values for the attributes `Hotel` and `City`. Other results, involving a different number of tables until the reaching of the universal relation, are possible (for example we can join the values of table `Accommodation` with the ones of `City` which is connected via foreign key).

5 Conclusion and future work

In this paper we discussed the need for considering keyword search over relational databases in the light of broader systems, where keyword search is just one of the components and which are aimed at better supporting users in their search tasks. These more complex systems call for appropriate evaluation methodologies which go beyond what is typically done today, i.e. measuring performances of components mostly in isolation or not related to the actual user needs, and, instead, able to consider the system as a whole, its constituent components, and their inter-relations with the ultimate goal of supporting actual user search tasks.

Future work will be devoted to develop a benchmark that follows the guidelines proposed in the paper. The benchmark will be experimented within the KEYSTONE COST Action[6], which is a network of researchers from 28 countries that aims to study topics related to keyword search in structured databases, and in the context of the CLEF initiative, which is the European forum for the evaluation of information access systems with an emphasis on multilingual and multimodal information with various levels of structure.

References

1. B. Aditya, G. Bhalotia, S. Chakrabarti, A. Hulgeri, C. Nakhe, Parag, and S. Sudarshan. BANKS: Browsing and Keyword Searching in Relational Databases. In Bressan et al. [13], pages 1083–1086.
2. M. Agosti, R. Berendsen, T. Bogers, M. Braschler, P. Buitelaar, K. Choukri, G. M. Di Nunzio, N. Ferro, P. Forner, A. Hanbury, K. Friberg Heppin, P. Hansen, A. Järvelin, B. Larsen, M. Lupu, I. Masiero, H. Müller, S. Peruzzo, V. Petras, F. Piroi, M. de Rijke, G. Santucci, G. Silvello, and E. Toms. PROMISE Retreat Report – Prospects and Opportunities for Information Access Evaluation. *SIGIR Forum*, 46(2), December 2012.
3. M. Agosti, N. Ferro, and C. Thanos. DESIRE 2011: First International Workshop on Data infrastructurEs for Supporting Information Retrieval Evaluation. In I. Ounis, I. Ruthven, B. Berendt, A. P. de Vries, and F. Wenfei, editors, *Proc. 20th International Conference on Information and Knowledge Management (CIKM 2011)*, pages 2631–2632. ACM Press, New York, USA, 2011.
4. S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A System for Keyword-Based Search over Relational Databases. pages 5–16, 2002.
5. M. Angelini, N. Ferro, G. Santucci, and G. Silvello. VIRTUE: A visual tool for information retrieval performance evaluation and failure analysis. *Journal of Visual Languages & Computing (JVLC)*, 25(4):394–413, August 2014.
6. T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements That Don't Add Up: Ad-Hoc Retrieval Results Since 1998. In D. W.-L. Cheung, I.-Y. Song, W. W. Chu, X. Hu, and J. J. Lin, editors, *Proc. 18th International Conference on Information and Knowledge Management (CIKM 2009)*, pages 601–610. ACM Press, New York, USA, 2009.
7. N. J. Belkin, R. Oddy, and H. M. Brooks. SK For Information Retrieval: Part I. Background and Theory. *Journal of Documentation*, 38(2):61–71, 1982.
8. S. Bergamaschi, E. Domnori, F. Guerra, R. Trillo-Lado, and Y. Velegrakis. Keyword Search over Relational Databases: A Metadata Approach. pages 565–576, 2011.
9. S. Bergamaschi, N. Ferro, F. Guerra, and G. Silvello. Keyword Search and Evaluation over Relational Databases: an Outlook to the Future. In *Proc. 7th International Workshop on Ranking in Databases (DBRank 2013) with VLDB 2013*, pages 8:1–8:3, 2013.
10. S. Bergamaschi, F. Guerra, M. Interlandi, R. Trillo-Lado, and Y. Velegrakis. QUEST: A Keyword Search System for Relational Data based on Semantic and Machine Learning Techniques. *PVLDB*, 6(12):1222–1225, 2013.
11. S. Bergamaschi, F. Guerra, S. Rota, and Y. Velegrakis. A Hidden Markov Model Approach to Keyword-Based Search over Relational Databases. In M. A. Jeusfeld, L. M. L. Delcambre,

[6] http://www.keystone-cost.eu/

and T. W. Ling, editors, *Proc. of the 30th International Conference on Conceptual Modeling (ER 2011)*, pages 411–420. Lecture Notes in Computer Science (LNCS) 6998, Springer, Heidelberg, Germany, 2011.

12. L. Blunschi, C. Jossen, D. Kossmann, M. Mori, and K. Stockinger. SODA: Generating SQL for Business Users. *PVLDB*, 5(10):932–943, 2012.
13. S. Bressan, A. B. Chaudhri, M.-L. Lee, J. Y. Yu, and Z. Lacroix, editors. *Proc. 28th International Conference on Very Large Data Bases (VLDB 2002)*. Morgan Kaufmann, 2002.
14. S. Buettcher, C. L. A. Clarke, and G. V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press, Cambridge (MA), USA, 2010.
15. M. J. Cafarella, A. Y. Halevy, and J. Madhavan. Structured data on the web. *Commun. ACM*, 54(2):72–79, 2011.
16. C. Y. Chan, B. C. Ooi, and A. Zhou, editors. *Proc. ACM SIGMOD International Conference on Management of Data (SIGMOD 2007)*. ACM Press, New York, USA, 2007.
17. S. Chaudhuri and G. Das. Keyword Querying and Ranking in Databases. *PVLDB*, 2(2):1658–1659, 2009.
18. E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton. Combining Keyword Search and Forms for Ad Hoc Querying of Databases. In U. Çetintemel, S. B. Zdonik, D. Kossmann, and N. Tatbul, editors, *Proc. ACM SIGMOD International Conference on Management of Data (SIGMOD 2009)*, pages 349–360. ACM Press, New York, USA, 2009.
19. C. W. Cleverdon. The Cranfield Tests on Index Languages Devices. In K. Spärck Jones and P. Willett, editors, *Readings in Information Retrieval*, pages 47–60. Morgan Kaufmann Publisher, Inc., San Francisco, CA, USA, 1997.
20. J. Coffman and A. C. Weaver. An empirical performance evaluation of relational keyword search techniques. *IEEE Trans. Knowl. Data Eng.*, 26(1):30–42, 2014.
21. E. Di Buccio, G. M. Di Nunzio, N. Ferro, D. K. Harman, M. Maistro, and G. Silvello. Unfolding Off-the-shelf IR Systems for Reproducibility. In J. Arguello, F. Diaz, J. Lin, and A. Trotman, editors, *Proc. SIGIR Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR 2015)*, 2015.
22. B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin. Finding Top-k Min-Cost Connected Trees in Databases. pages 836–845.
23. European Commission. Communication from the Coommission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions – Towards a thriving data-driven economy. COM(2014) 442 final, `http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52014DC0442&from=EN`, 2014.
24. M. Ferrante, N. Ferro, and M. Maistro. Injecting User Models and Time into Precision via Markov Chains. In S. Geva, A. Trotman, P. Bruza, C. L. A. Clarke, and K. Järvelin, editors, *Proc. 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2014)*, pages 597–606. ACM Press, New York, USA, 2014.
25. N. Ferro, editor. *Bridging Between Information Retrieval and Databases - PROMISE Winter School 2013, Revised Tutorial Lectures*. Lecture Notes in Computer Science (LNCS) 8173, Springer, Heidelberg, Germany, 2014.
26. N. Ferro. CLEF 15th Birthday: Past, Present, and Future. *SIGIR Forum*, 48(2):31–55, December 2014.
27. N. Ferro and G. Silvello. CLEF 15th Birthday: What Can We Learn From Ad Hoc Retrieval? In E. Kanoulas, M. Lupu, P. Clough, M. Sanderson, M. Hall, A. Hanbury, and E. Toms, editors, *Information Access Evaluation – Multilinguality, Multimodality, and Interaction. Proceedings of the Fifth International Conference of the CLEF Initiative (CLEF 2014)*, pages 31–43. Lecture Notes in Computer Science (LNCS) 8685, Springer, Heidelberg, Germany, 2014.

28. N. Ferro and G. Silvello. Rank-Biased Precision Reloaded: Reproducibility and Generalization. In N. Fuhr, A. Rauber, G. Kazai, and A. Hanbury, editors, *Advances in Information Retrieval. Proc. 37th European Conference on IR Research (ECIR 2015)*, pages 768–780. Lecture Notes in Computer Science (LNCS) 9022, Springer International Publishing Switzerland, 2015.
29. N. Ferro, G. Silvello, H. Keskustalo, A. Pirkola, and K. Järvelin. The Twist Measure for IR Evaluation: Taking User's Effort Into Account. *Journal of the American Society for Information Science and Technology (JASIST)*, (in print).
30. D. Harman and C. Buckley. SIGIR 2004 Workshop: RIA and "Where can IR go from here?". *ACM SIGIR Forum*, 38(2):45–49, 2004.
31. D. Harman and C. Buckley. Overview of the Reliable Information Access Workshop. *Information Retrieval*, 12(6):615–641, 2009.
32. D. K. Harman. *Information Retrieval Evaluation*. Morgan & Claypool Publishers, USA, 2011.
33. D. K. Harman and E. M. Voorhees, editors. *TREC. Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge (MA), USA, 2005.
34. H. He, H. Wang, J. Yang, and P. S. Yu. BLINKS: Ranked Keyword Searches on Graphs. In Chan et al. [16], pages 305–316.
35. M. A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
36. T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011.
37. V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In Bressan et al. [13], pages 670–681.
38. P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer, Heidelberg, Germany, 2005.
39. K. Järvelin and J. Kekäläinen. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, October 2002.
40. G. Kasneci, M. Ramanath, M. Sozio, F. M. Suchanek, and G. Weikum. STAR: Steiner-Tree Approximation in Relationship Graphs. In Y. E. Ioannidis, D. L. Lee, and R. T. Ng, editors, *Proceedings of the 25th International Conference on Data Engineering*, pages 868–879. IEEE Computer Society, 2009.
41. J. Kekäläinen and K. Järvelin. Using Graded Relevance Assessments in IR Evaluation. *Journal of the American Society for Information Science and Technology (JASIST)*, 53(13):1120—1129, November 2002.
42. D. Kelly. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval (FnTIR)*, 3(1–2):1–224, Jan. 2009.
43. R. Khare, Y. An, and I.-Y. Song. Understanding Deep Web Search Interfaces: A Survey. *SIGMOD Rec.*, 39(1):33–40, 2010.
44. Y. Luo, X. Lin, w. Wang, and X. Zhou. SPARK: Top-K Keyword Query in Relational Databases. In Chan et al. [16], pages 115–126.
45. G. Marchionini. Exploratory Search: From Finding to Understanding. *Commun. ACM*, 49(4):41–46, 2006.
46. B. R. Rowe, D. W. Wood, A. L. Link, and D. A. Simoni. *Economic Impact Assessment of NIST's Text REtrieval Conference (TREC) Program.* RTI Project Number 0211875, RTI International, USA. `http://trec.nist.gov/pubs/2010.economic.impact.pdf`, July 2010.
47. T. Sakai. Metrics, Statistics, Tests. In Ferro [25], pages 116–163.
48. A. Simitsis, G. Koutrika, and Y. E. Ioannidis. Précis: from unstructured keywords as queries to structured databases as answers. *VLDB Journal*, 17(1):117–149, 2008.

49. M. D. Smucker and C. L. A. Clarke. Time-Based Calibration of Effectiveness Measures. In W. Hersh, J. Callan, Y. Maarek, and M. Sanderson, editors, *Proc. 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*, pages 95–104. ACM Press, New York, USA, 2012.
50. S. Tata and G. M. Lohman. SQAK: Doing more with keywords. In J. T.-L. Wang, editor, *Proc. ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*, pages 889–902. ACM Press, New York, USA, 2014.
51. D. Tsichritzis and A. Klug. The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems. *Information Systems*, 3(3):173–191, 1978.
52. W. Webber. Evaluating the effectiveness of keyword search. *IEEE Data Engin. Bull.*, 33(1):55–60, 2010.
53. G. Weikum. Where's the Data in the Big Data Wave? ACM SIGMOD Blog, `http://wp.sigmod.org/?p=786`, March 2013.
54. J. X. Yu, L. Qin, and L. Chang. Keyword Search in Relational Databases: A Survey. *IEEE Data Eng. Bull.*, 33(1):67–78, 2010.