A Framework for Evaluating Database Keyword Search Strategies

Joel Coffman
University of Virginia
Charlottesville, VA
jcoffman@cs.virginia.edu

Alfred C. Weaver
University of Virginia
Charlottesville, VA
weaver@cs.virginia.edu

ABSTRACT

With regard to keyword search systems for structured data, research during the past decade has largely focused on performance. Researchers have validated their work using ad hoc experiments that may not reflect real-world workloads. We illustrate the wide deviation in existing evaluations and present an evaluation framework designed to validate the next decade of research in this field. Our comparison of 9 state-of-the-art keyword search systems contradicts the retrieval effectiveness purported by existing evaluations and reinforces the need for standardized evaluation. Our results also suggest that there remains considerable room for improvement in this field. We found that many techniques cannot scale to even moderately-sized datasets that contain roughly a million tuples. Given that existing databases are considerably larger than this threshold, our results motivate the creation of new algorithms and indexing techniques that scale to meet both current and future workloads.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation*; H.2.4 [**Database Management**]: Systems—*Relational databases*

General Terms

Experimentation, Standardization

Keywords

Evaluation, keyword search

1. INTRODUCTION

Numerous researchers have proposed keyword search strategies for structured data, which includes semi-structured documents (e.g., XML) and information stored in relational databases. This push reflects Internet users' increasing reliance on keyword search and also reflects a desire to hide underlying data representations and to eliminate complex query languages from end users. To the best of our knowledge, none of these proposed systems has reached mainstream use. One potential barrier to deploying these systems is the ad hoc evaluations performed by researchers.

The history of the information retrieval (IR) community illustrates the importance of standardized evaluation. Singhal [28] states, "A system for experimentation coupled with good evaluation methodology allowed rapid progress in the field and paved [the] way for many critical developments." The Text REtreival Conference (TREC) testifies to the impact of standardized evaluation, for search effectiveness *doubled* within six years of its inception [30].

The Initiative for the Evaluation of XML retrieval (INEX) workshop [10] established standardized evaluation procedures for XML retrieval. Despite the similarity of keyword search in semi-structured data and relational data,[1] relational keyword search systems have not been evaluated at this venue. Perhaps researchers see evaluation forums such as INEX as too expensive to validate experimental system designs, but standardized evaluation is essential for real progress. The strategic step of creating a DB&IR evaluation forum has yet to occur. Without it, progress will not match that of the larger IR community. In the interim (and as also suggested by Webber [32]), the community should coalesce behind a standardized set of datasets and queries for evaluating search systems—we describe such a framework in this paper. According to Chen *et al.* [3], "Contributions from the research community are highly demanded for developing comprehensive frameworks for evaluating the retrieval and ranking strategies of keyword search on various structured data models." Our evaluation framework enables direct comparison of the effectiveness and performance of different search techniques. In the remainder of this section, we illustrate the difficulties inherent to searching structured data and also present the contributions of this work.

1.1 Keyword Search in Structured Data

The ubiquitous search textbox has transformed the way people interact with information. Despite the wide-ranging success of Internet search engines in making information accessible, searching structured data remains a challenge. Both semi-structured and relational data introduce challenges not encountered in unstructured IR. For instance, the correct granularity of search results must be reconsidered. An XML document might contain a single element that is pertinent to a given query along with many unrelated elements. The

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...$10.00.

[1] Keyword search over data graphs generalizes both.

Physical views

Person	
id	name
10	Ford, Harrison
11	Connery, Sean

Character	
id	name
7	Indiana Jones
9	Professor Henry Jones

Movie	
id	title
18	*Raiders of the Lost Ark*
19	*Indiana Jones and the Last Crusade*

Cast		
personId	characterId	movieId
10	7	18
10	7	19
11	9	19

Logical views

Person
Person.name
Ford, Harrison
Connery, Sean

Person ← Cast → Character	
Person.name	Character.name
Ford, Harrison	Indiana Jones
Connery, Sean	Professor Henry Jones

Person ← Cast	**→ Character** **→ Movie**
Person.name	Character.name Movie.title
Ford, Harrison	Indiana Jones *Raiders of the Lost Ark*
Ford, Harrison	Indiana Jones *Indiana Jones and the Last Crusade*
Connery, Sean	Professor Henry Jones *Indiana Jones and the Last Crusade*

Figure 1: Physical and logical views of relational data. Physical views are actual database relations while logical views of the information are not. Logical views form (potentially) relevant search results that present related information as a unified whole.

Digital Bibliography & Library Project (DBLP)[2]'s XML dump currently contains more than 1.3 million publications; searching this repository for a particular paper should return only the information about that paper and not the complete bibliography.

Identifying relevant results is further complicated due to the fact that the data's physical representation often does not match a logical view of the information. For example, relational data is normalized to eliminate redundancy. The schema separates logically connected information, and foreign keys identify related rows. Whenever search queries cross these relationships, the data must be mapped back to a logical view to provide meaningful search results. As shown in Figure 1, answering the question "who played Professor Henry Jones in *Indiana Jones and the Last Crusade*" requires data from all 4 physical relations and is answered by the bottommost logical view (lower right of figure).

Recombining disparate pieces of data into a unified whole makes searching structured data significantly more complex than searching unstructured text. In unstructured IR, each document may be indexed prior to searches. In contrast, indexing all the possible logical (or materialized) views of structured data results in an explosion in the size of the index because the number of possible views is limited only by the data itself. For example, Su and Widom [29] indexed a subset of the logical views over a database and found that the index exceeded the size of the original data between two and eight times.

Searching structured data continues to grow in importance as websites serve increasing amounts of data on-demand in response to user actions and to personalize webpages. This information is normally stored in a relational database, and Bergman [1] estimates that data to be orders of magnitude larger than the static web. The explosive growth of social networking and microblogging websites contributes to the ever increasing amount of data stored relationally. Novel search techniques are required to access this information efficiently and effectively.

1.2 Contributions and Outline

The major contributions of this paper are threefold. Section 2 reviews related work, providing a high-level overview of keyword search systems for structured data. We also present details regarding the evaluations of these systems published at top research venues (e.g., VLDB, SIGMOD, ICDE). Our survey exposes the ad hoc techniques currently being used by researchers and a lack of experimental repeatability given the scanty details provided in the literature. Webber [32] independently considers many of these details and arrives at similar conclusions.

Section 3 describes our evaluation framework. In contrast to many previous evaluations, our evaluation framework covers 3 dramatically different datasets and contains 50 information needs for each dataset. To promote standardized evaluation, we will make our datasets, queries, and relevance assessments available for other researchers to use in their own experiments. Our framework carefully considers the unique requirements of keyword search systems for structured data and follows the traditional definitions of relevance developed by the IR community.

We evaluate 9 state-of-the-art systems in Section 4 using our framework. Our results do not support the claimed effectiveness of existing search techniques. In fact, our results indicate that many strategies perform comparably with regard to search effectiveness despite contrary claims appearing in the literature. Our work is also the first to investigate the correlation among the results produced by each system to determine if faster query processing techniques produce results comparable to more effective ranking schemes. We highlight some of the ranking factors that work particularly well for our datasets and query workloads and also present our conclusions regarding state-of-the-art keyword search systems for structured data.

2. BACKGROUND

Relational keyword search systems tend to follow a strict dichotomy. One approach uses IR techniques to rank results while the other approach considers the problem a form of

[2] http://dblp.uni-trier.de/

proximity search. Proximity search systems minimize the distance between search terms in the data graph. If results are enumerated in increasing order of weight, the problem is an instance of the group Steiner tree problem, which is NP-complete. Hence, a variety of heuristics and specialized data structures have been proposed to make the problem tractable.

2.1 Related Work

DISCOVER [15] proposed the general system architecture that most IR approaches follow. Search results are networks of tuples that collectively contain all search terms. The candidate network generator enumerates all networks of tuples that are potential results. Because the total number of candidate networks is limited only by the actual data, efficient enumeration requires that a maximum candidate network size be specified. Hristidis *et al.* [14] refined DISCOVER by adding support for OR semantics and by including an IR scoring function (pivoted normalization scoring [28]) to rank results. Their monotonic score aggregation function enables efficient execution. Liu *et al.* [21] propose four additional normalizations for pivoted normalization scoring to adapt it to a relational context. SPARK [22] returns to a non-monotonic score aggregation function for ranking results; the non-monotonic function precludes the use of existing query processing algorithms. Qin *et al.* [27] investigate query processing to eliminate the middleware layer between search systems and the underlying relational database.

BANKS [2] introduced the backward expanding search heuristic for enumerating results. Edges in the data graph denote a relationship between the vertices (a foreign key between the relational tuples). For every edge (u, v) induced by a foreign key, a backward edge (v, u) with a larger weight is also added. These backward edges enable edge directionality to be considered when identifying results. Bidirectional search [17] improves the performance of the original system. Ding *et al.* [8] use dynamic programming—the DPBF algorithm—to identify the minimal group Steiner tree in time exponential in the number of search terms. BLINKS [13] uses a bidirectional index to improve query performance. EASE [20] proposes a graph index to support efficient searches on unstructured, semi-structured, and relational data where query results are subgraphs whose radius does not exceed a maximum size. Golenberg *et al.* [11] guarantee a polynomial delay when enumerating results but enumerate results by height rather than weight. Dalvi *et al.* [7] investigate keyword search on external memory graphs using a multi-granular graph representation to reduce I/O costs as compared to data structures managed by virtual memory. STAR [18] approximates the optimal result tree in pseudo-polynomial time, which is shown to outperform several other proximity search heuristics.

2.2 Survey of Existing System Evaluations

In this section, we survey the evaluations of systems published at leading research venues in this field. In our experience, these evaluations exceed the scope and quality of those published in smaller venues yet clearly illustrate the ad hoc evaluation techniques currently practiced by researchers.

Table 1 summarizes the datasets used in previous evaluations. Two of the most popular datasets (DBLP and the Internet Movie Database (IMDb)) lack a canonical schema, which leads to several different schemas appearing in evaluations. As evidenced by the number of table footnotes, the information contained within each dataset is fluid: researchers often use arbitrary subsets in their evaluation, but this practice raises two experimental design questions that have not been addressed. First, do the techniques really scale to the level researchers claim? Approaches that cannot handle the amount of data present in today's data repositories will be unable to cope with tomorrow's infusion of new data. Second, are the properties of the resulting subsets representative of the original dataset? Arbitrary subsets can mask scalability issues and artificially bolster the reported effectiveness of the system.

Table 1: Summary of datasets used by previous researchers. In this and future tables and figures, systems are ordered by date of publication, i.e., BANKS is the oldest system.

System	DBLP	IMDb	Mondial	MovieLens	TPC-H	Other[a]
BANKS [2]	●					○
DISCOVER [15]					●[b]	
Efficient [14]	○[c]					
Bidirectional [17]	●	○				○
Effective [21]						○
DPBF [8]	●			●		
BLINKS [13]	○[d]	○				
SPARK [22]	○	●	●			
EASE [20]	●			●		○
Golenberg *et al.* [11]			●			
Dalvi *et al.* [7]	●	○				
STAR [18]	○[e]	○[f]				○
Qin *et al.* [27]	●	●				

Legend
● identical relational schemas
○ different schemas or schemas not provided

[a] This column denotes the presence of additional datasets in the evaluation. None of these datasets are related.

[b] The database content is varied for the experiments.

[c] A random number of citations is added to each paper.

[d] Most papers not cited by or citing another were removed.

[e] Subset includes 15,000 vertices and 150,000 edges; edge weights are assigned randomly.

[f] Subset includes 30,000 vertices and 80,000 edges; edge weights are assigned randomly.

Table 2 lists more detailed information about previous system evaluations. As evidenced by the table, many evaluations reported in the literature focus exclusively on performance. Few experiments consider the quality of search results. Both should be considered, for it is trivial to quickly return poor quality results. Given the number of different ranking schemes and execution heuristics, search quality should not be ignored.

Many experiments use queries that are created using random keywords that appear in the database. According to Manning *et al.* [23], "using random combinations of query terms as an information need is generally not a good idea because typically they will not resemble the actual distribution of information needs." More recent evaluations use queries constructed by the researchers, but these evaluations suffer

Table 2: Summary of experiments reported in the literature. Empty cells indicate that the information is not relevant or was not provided and could not be determined from the details of the original evaluation.

	Experiment	Type		Queries				
System	Dataset	Perf.	Qual.	Selection	$\lvert Q\rvert$	$[\![q]\!]$	$\overline{[\![q]\!]}$	Rel.
BANKS [2]	DBLP / other	✓		Researchers	7			
DISCOVER [15]	TPC-H	✓		Random		2–5		
	TPC-H	✓		Random	200	2–5		
	TPC-H	✓		Random	100	2	2.0	
Efficient [14]	DBLP	✓		Random	100	2		
	DBLP	✓		Random		2–5		
Bidirectional [17]	DBLP	✓	✓	Random	200	2–7		○
Effective [21]	Lyrics		✓	Search log	50	2–20	6.7	●
DPBF [8]	DBLP[a]	✓		Random	500	2–6	4.0	
	MovieLens[a]	✓		Random	500	2–6	4.0	
	MovieLens	✓		Random	100	4	4.0	
BLINKS [13]	DBLP	✓		Researchers	60	2–4	3.0	
	IMDb[a]	✓		Researchers	40	2–8	5.0	
SPARK [22]	DBLP	✓	✓	Researchers	18	2–4	2.7	●
	IMDb	✓	✓	Researchers	22	2–3	2.4	●
	Mondial	✓	✓	Researchers	35	2–3	2.2	●
EASE [20]	DBLife	✓	✓	Researchers	5	4–5	4.6	○
	DBLP	✓	✓	Researchers	5	2–4	3.2	○
	MovieLens	✓	✓	Researchers	5	3–4	3.4	○
	previous 3	✓	✓	Researchers	5	3–4	3.8	○
Golenberg *et al.* [11]	Mondial[a]	✓		Random	36	2–10	6.0	
Dalvi *et al.* [7]	DBLP	✓	✓	Researchers	8	2–6	3.5	○
	IMDb	✓	✓	Researchers	4	2–3	2.5	○
STAR [18]	DBLP[a]	✓		Random	180	3,5,7	5.0	
	IMDb[a]	✓		Random	180	3,5,7	5.0	
	YAGO[a]	✓		Random	120	3,6	4.5	
Qin *et al.* [27]	DBLP	✓		Researchers	17	3–5	3.1	
	IMDb	✓		Researchers	20	3–5	3.3	

[a] The queries are equally partitioned among the number of query terms.

Legend	
Perf.	performance evaluation
Qual.	search quality evaluation
$\lvert Q\rvert$	total number of queries
$[\![q]\!]$	range in number of query terms
$\overline{[\![q]\!]}$	average number of terms per query
●	explicitly define relevance
○	relevance considered, but criteria unclear

from an insufficient number of queries. When evaluating search systems, 50 information needs is the traditional minimum [23, 31]. In addition, we have noticed that many queries selected by researchers subtly reflect their proposed ranking scheme. For the evaluation of Effective [21], every query contained a "schema" term (a search term that matches the name of a database relation or attribute). Matching search terms to the relational schema was not considered in previous work so naturally the proposed system outperforms competing approaches.

Among the systems that do consider the quality of search results, the definition of relevance is often vague. The developers of EASE [20] state, "Answer relevance is judged from discussions of researchers in our database group". Such a vague definition does not make the assessment process reproducible by a third-party. SPARK [22] used the following definition: relevant results must 1) contain all query keywords and 2) have the smallest size (of any result satisfying the first criterion). In contrast to this definition, the IR community is clear that relevant results must address the underlying information need and not just contain all search terms [23, 32].

Our survey also revealed that systems often perform abnormally well with regard to effectiveness metrics, which we attribute to non-standard relevance definitions coupled with very general queries. Bidirectional [17] claims, "The recall was found to be close to 100% for all the cases with an equally high precision at near full recall." In the evaluation of EASE [20], the queries admit a large number of relevant answers, e.g., "Indiana Jones and the Last Crusade person" where any cast member is relevant. EASE [20] reports a precision of 0.9 for 100 retrieved results, which is considerably better than the best scores reported at TREC (roughly 0.25) [32].

In Table 3, we show the systems that compare against previous work. With the exception of STAR [18], these comparisons are limited to 1–2 other systems. No evaluation appearing at a top-tier conference compares IR ranking schemes with proximity search heuristics. Lack of cross-evaluation makes it difficult to compare the trade-offs between approaches that vary widely with respect to both query processing and ranking results.

In summary, no standardized datasets or query workloads exist for evaluating the performance or effectiveness of existing systems. Even among evaluations that use the same dataset, results are not comparable because researchers create random subsets of the original database (Table 1). The past decade of research primarily focuses on performance (Table 2). Query workloads vary widely across evaluations from large collections of randomly generated queries (these may not reflect real user queries) to small numbers of more representative queries created by researchers (the number of queries does not meet the accepted minimum for evaluating retrieval systems). Finally, comparison among systems is relatively limited (Table 3). The systems we include in our survey have been published in prestigious proceedings (e.g., VLDB, SIGMOD, ICDE), which indicates the unfortunate re-

Table 3: System evaluation comparison matrix. Evaluations that compare against other systems are listed on the left; the systems they compare against appear at the top of the table. Comprehensive evaluations would compare against all previous work (i.e., the lower left entries would all be ●).

	BANKS [2]	DISCOVER [15]	Efficient [14]	Bidirectional [17]	Effective [21]	DPBF [8]	BLINKS [13]	SPARK [22]	EASE [20]	Golenberg *et al.* [11]	Dalvi *et al.* [7]	STAR [18]	Qin *et al.* [27]
BANKS [2]	–												
DISCOVER [15]		–											
Efficient [14]			–										
Bidirectional [17]	●		○	–									
Effective [21]		○	○		–								
DPBF [8]	●			●		–							
BLINKS [13]				●			–						
SPARK [22]			●		○			–					
EASE [20]						●			–				
Golenberg *et al.* [11]										–			
Dalvi *et al.* [7]	●		○								–		
STAR [18]	●			●		●	●					–	
Qin *et al.* [27]								●					–
Our evaluation	●	●	●	●	○	●	●	●					

Legend

●	exact comparison
○	characteristics of system approximated

ality that ad hoc evaluations are an accepted practice rather than aberrations.

3. EVALUATION FRAMEWORK

3.1 Datasets

Two of our datasets are derived from popular websites (IMDb and Wikipedia). The third (Mondial) is an ideal counterpoint due to its smaller size. Table 4 provides detailed statistics regarding all three of our datasets. Even though our datasets are relatively small, they are sufficiently challenging for existing search techniques (as shown in Section 4), and both IMDb and Wikipedia can be scaled up as search techniques improve.

DBLP is one of the more popular datasets included in previous evaluations. We elected not to include it because the content of the DBLP database is similar to IMDb (e.g., names and titles) so results across these two datasets would likely be similar.

3.1.1 Mondial

The Mondial dataset [24] comprises geographical and demographic information from the CIA World Factbook, the *International Atlas*, the TERRA database, and other web sources. We downloaded the relational version from its website. Mondial's cyclic data graph is much more complex than the others included in our evaluation.

3.1.2 IMDb

We downloaded IMDb's plain text files and created a relational database using IMDbPY 4.1. Using a third-party tool eliminates any bias in the creation of the schema, which has the potential to significantly impact search effectiveness and performance. The initial database contained 20 relations with more than 44 million tuples. Because many proximity search systems require an in-memory data graph, our dataset is a subset of the original database. We note that our subset potentially overstates the effectiveness of the various search techniques for this dataset.

3.1.3 Wikipedia

Our final dataset is a selection of articles from Wikipedia. The complete Wikipedia contains more than 3 million articles, which makes including all of them infeasible. Our selection includes more than 5500 articles chosen for the 2008–2009 Wikipedia Schools DVD, a general purpose encyclopedia, which contains content roughly equal to a traditional 20 volume encyclopedia. We deemed general content more desirable than a larger number of articles chosen randomly from the corpus. We drop all the tables unrelated to articles or users and augment the `PageLinks` table with an additional foreign key to explicitly indicate referenced pages.

3.2 Queries

Fifty information needs is the traditional minimum for evaluating retrieval systems [23, 31]. This number of information needs reflects the fact that performance varies widely across queries for the same document collection. Table 2 shows that other evaluations that use representative queries have not included this number of distinct information needs. Liu *et al.* [21] repeat a number of information needs in their queries. All our queries reflect distinct information needs.

We do not use real user queries extracted from a search engine log for three reasons. First, many queries are inherently ambiguous. Given the query "Indiana Jones," it is impossible to determine the underlying information need. Does the user want information about the character Indiana Jones or the films named after that title character? Without knowing the user's intent, it is impossible to judge whether the character or a film is the desired result. In contrast, a synthetic

Table 4: Characteristics and simplified schema of our three evaluation datasets. The reported size includes database indices.

Dataset	Size (MB)	Relations	Tuples
Mondial	9	28	17,115
IMDb	516	6	1,673,074
Movie (**id**, title, year)			181,706
Person (**id**, name)			273,034
Character (**id**, name)			206,951
Role (**id**, type)			11
Cast (***movieId***, ***personId***, ***characterId***, ***roleId***)			812,694
MovieInfo (**id**, *movieId*, info)			198,678
Wikipedia	550	6	206,318
Page (**id**, title)			5,540
Revision (**id**, *pageId*, *textId*, *userId*)			5,540
Text (**id**, text)			5,540
User (**id**, name)			1,745
PageLinks (**id**, *from*, *to*)			187,951
UserGroups (***userId***, **group**)			2

Legend **primary key**, *foreign key*, full text index

Table 5: Query and result statistics.

	Search log [26]	Synthesized			Results	
Dataset	$\overline{[\![q]\!]}$	$\|Q\|$	$[\![q]\!]$	$\overline{[\![q]\!]}$	$[\![R]\!]$	$\overline{[\![R]\!]}$
Mondial		50	1–5	2.04	1–35	5.90
IMDb	2.71	50	1–26	3.88	1–35	4.32
Wikipedia	2.87	50	1–6	2.66	1–13	3.26
Overall	2.37	150	1–26	2.86	1–35	4.49

Legend

$\|Q\|$	total number of queries
$[\![q]\!]$	range in number of query terms
$\overline{[\![q]\!]}$	average number of terms per query
$[\![R]\!]$	range in number of relevant results per query
$[\![R]\!]$	average number of relevant results per query

query workload based on overt information needs avoids this problem. Second, we believe a large number of queries will reflect the limitations of existing search engines—namely, web search engines are not designed to connect disparate pieces of information. Users implicitly adapt to this limitation by submitting few (Nandi and Jagadish [25] report less than 2%) queries that reference multiple database entities. Third, the available search logs provide an insufficient number of user queries for many domain-specific datasets (e.g., DBLP and Mondial).

Ideally, a number of individuals all create candidate information needs for an evaluation, and a subset from this pool is actually included. This procedure is used by established evaluation forums (e.g., TREC and INEX) but is impractical for this work given the lack of incentive for others to participate. Consequently, we independently derived a variety of information needs for each dataset.

Table 5 provides the statistics of our query workload and the relevant results for each dataset. Five IMDb queries are outliers because they include an exact quote from a movie. Omitting these queries reduces the maximum number of terms in any query to 7 and the average number of terms per query to 2.91. The statistics for our queries are similar to those reported for web queries [16] and our independent analysis of query lengths from a commercial search engine log [26], which suggests that our queries are representative of real-world user queries. In contrast, the average length of queries used in previous studies (see Table 2) is almost always greater than the average for web queries.

3.3 Assessing Relevance

Relevance is assessed relative to the original information need. For all our information needs, we identify relevant results by constructing our information needs around a template of database relations. We execute a number of SQL queries to identify all possible results satisfying the information need and judge each of these results for relevance. Thus, careful construction of our information needs allows exhaustive relevance judgments for the collection. As is done at TREC, relevance assessments are carried out by a single individual. While using a single assessment as the gold standard does affect the absolute values of effectiveness metrics, it has not been shown to impact the relative effectiveness of the systems under comparison [23, 31].

We use binary relevance assessments when judging results. In adherence to the Cranfield paradigm [5], TREC traditionally used binary relevance assessments, which also have been used by all the previous evaluations reported in Section 2. In contrast, INEX distinguishes between highly relevant and partially relevant results. We believe this distinction to be good in theory, but it adds considerable complexity to the assessment process and also questions some of the central assumptions of the Cranfield paradigm—namely, all relevant documents are equally desirable. In practice, the notion of relevance, especially for structured data, is extremely subtle, involving novelty and diversity in the results. We refer the reader to Clarke *et al.* [4] for additional details.

4. EXPERIMENTS

In this paper, we do not consider the efficiency of search techniques and instead focus exclusively on search effectiveness. Obviously, performance plays a key factor when assessing system utility. The evaluations reported in the literature already investigate the performance aspect of their systems. Our work complements the evaluations appearing in the literature by comparing systems on the basis of search quality. Omitting a performance comparison also stems from a pragmatic reason: we have not yet had the opportunity to implement many of the optimized query processing techniques proposed by the original researchers.

Our experiments target three questions. First, what is the effectiveness of each system, especially in comparison to each other? For previous evaluations that do consider search effectiveness, we hope to corroborate their claims. Second, what impact does the number of retrieved (top-k) results have when evaluating search quality? Previous experiments at TREC show that retrieving too few results can significantly impact systems' precision-recall curves [12], and some previous evaluations of search effectiveness only include the top-10 or top-20 results. Third, are the systems' results highly correlated with each other? We expect many systems (e.g., BANKS [2] and its successor Bidirectional [17]) to return similar results, which would make performance the only significant difference between these systems.

4.1 Metrics

To measure the effectiveness of search systems, we use four metrics. The number of top-1 relevant results is the number of queries for which the first result is relevant. Reciprocal rank is the reciprocal of the highest ranked relevant result for a given query. Both of these measures tend to be very noisy but indicate the quality of the top-ranked results. Average precision for a query is the average of the precision values calculated after each relevant result is retrieved (and assigning a precision of 0.0 to any relevant results *not* retrieved). Mean average precision (MAP) averages this single value across information needs to derive a single measure of quality across different recall levels and information needs. To summarize the entire precision-recall curve, we use 11-point interpolated average precision. To calculate each metric, we retrieve the top 1000 results for each system.

To measure the correlation between the results returned by the various systems, we use the normalized Kendall distance [19]. The Kendall distance between two permutations is the number of pairwise swaps needed to convert one permutation into the other. Because we consider only the top-k results from each system, we use the generalization proposed by Fagin *et al.* [9].

4.2 Implementations

Our evaluation includes most of the systems described in Section 2.1. Efficient [14], Effective [21], and SPARK [22] all use IR scoring schemes whereas BANKS [2], Bidirectional [17], DPBF [8], and BLINKS [13] are proximity search systems. DISCOVER [15] partially bridges these approaches by ranking results by the number of joins (i.e., edges) in the result tree. We also include our own previous work, structured cover density ranking (CD) [6], which is designed to reflect users' preferences regarding the ranking of search results. Compared to previous evaluations, our work doubles the number of system comparisons (see Table 3). Five systems described in Section 2.1 were not included due to their significant reimplementation effort.

Our reimplementations of systems include a number of enhancements.[3] We generalized DISCOVER's candidate network generation algorithm when we realized it was missing relevant results and modified DPBF to distinguish trees containing a single node (which improved its effectiveness for a number of our topics). Due to space limitations, we do not describe query processing and other aspects of these systems but refer readers to the original papers.

For each system, we set all tuning parameters to the values suggested by the authors. None of the systems—including our own ranking scheme—are tuned using our datasets and queries because such tuning would overstate the effectiveness of the systems [23]. Bidirectional, DPBF, and BLINKS could not handle our IMDb dataset due to excessive ($\geq$ 2.7 GB) memory requirements. In these cases, we use any results output before running out of memory. A system's omission from a table or figure means that *no* query returned even a single result.

4.3 Results

Figures 2 and 3 summarize the effectiveness of each system. The relative rank of each system in comparison to the others is similar in both graphs. We quickly see that effectiveness varies considerably across both datasets and different search techniques. In contrast to previous evaluations, no single system outperforms all the others.

Figure 2 shows the mean reciprocal rank for each system for queries where exactly one database tuple is relevant (20, 20, and 15 topics for the respective datasets). Nandi and Jagadish [25] report that these single entity queries are the most common type of query posed to existing search engines. We expected the proximity search systems (BANKS, Bidirectional, DPBF, and BLINKS) to perform poorly on this task because ranking results by edge weight does not allow these systems to distinguish trees containing a single node. Instead, we see that these systems perform very well on the Mondial dataset (the best 3 systems are all proximity search engines), BANKS significantly outperforms the IR approaches (Efficient, Effective, and SPARK) on the IMDb dataset, and both BANKS and Bidirectional tie for second most effective on Wikipedia. These results counter our original intuition regarding the types of retrieval tasks suited to proximity search techniques. The results for the IR approaches are disappointing in view of the excellent scores of the proximity search systems. Analyzing the results returned for each query sheds some light on the underlying reason: the IR-style ranking schemes prefer larger results that contain additional instances of the search terms instead of the smallest result that satisfies the query.

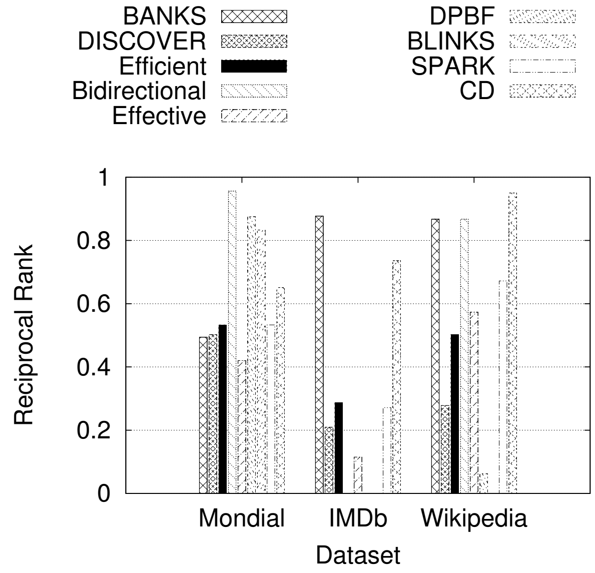

Figure 2: Reciprocal rank for the queries targeting exactly one database entity. Higher bars are better.

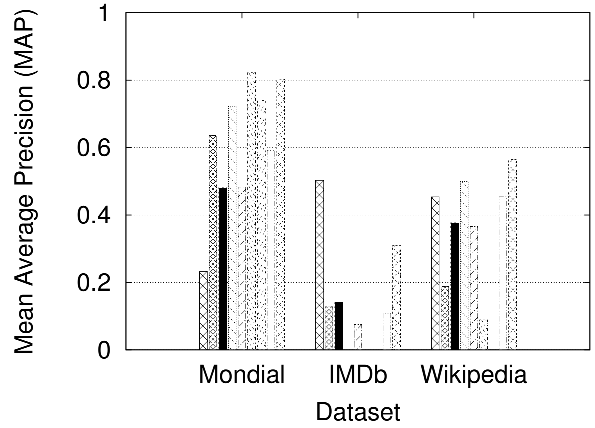

Figure 3: MAP measured across the various systems and datasets. Higher bars are better.

Figure 3 shows the MAP scores for the systems across all queries and datasets and illustrates three interesting trends. First, the IR approaches (Efficient, Effective, and SPARK) all perform comparably due to their common baseline scoring function, pivoted normalization scoring. The different normalizations that each apply to the original scoring function accounts for their minor differences. Even though these three systems are outperformed by proximity search techniques, it does *not* indicate that there is no advantage to IR-style ranking. Cover density ranking (CD) is also based on previous work from the IR community, and it performs much better than the competing IR approaches. In fact, cover density ranking is the second-best system for Mondial (see also Table 7) and is the most effective system for

[3] Our reimplementation of Liu *et al.*'s work [21] does not include phrase-based ranking.

Wikipedia. Second, scalability remains a significant concern for the proximity search systems. BANKS is the only proximity search system that completes *any* IMDb query, and the overhead of BLINKS's bi-level index prevents it from indexing the Wikipedia dataset. Third, both BANKS and Bidirectional include node prestige when scoring result trees. Their node prestige factor accounts for their good scores on the Wikipedia dataset and is contrasted by the poor score of DPBF, which ranks results solely by edge weight.

Table 6 presents 11-point precision and recall for a subset of Wikipedia topics most similar to those encountered at TREC. The query terms are present in many articles, yet most articles containing the search terms are not relevant. Here we see the IR-style scoring functions (particularly Efficient) outperforming the proximity search systems because their scoring functions were designed for lengthy unstructured text. Efficient, which least modifies pivoted normalization scoring, has the most stable performance across the entire precision-recall curve. In contrast, the effectiveness of the other IR-style scoring functions drops precipitously at higher recall levels. BANKS and Bidirectional both perform well due to their consideration of node prestige, which interestingly translates to reasonable effectiveness even for our TREC-style topics.

Table 7 summarizes results for the MONDIAL topics and highlights the differences observed between our evaluation and SPARK's evaluation [22].[4] The left half of the table indicates that SPARK's scoring function significantly improves upon Efficient and Effective. SPARK's purported benefit—more than doubling retrieval effectiveness—is not corroborated by our experiments (the right half of the table), which shows at best 20% improvement over Efficient and Effective. While some variation is natural, the discrepancy between 20% improvement and 100% improvement is not, especially given that the only variation is the query workload. When combined with the above-average score reported for SPARK (0.986 verses 0.8 for mean reciprocal rank by the best systems at TREC [32]), our results question the validity of the previous evaluation and further underscore the need for standardized evaluation.

In Figure 4, we show interpolated-precision curves for a variety of values of k for the same Wikipedia topics used in Table 6. These topics have the most interesting precision-recall curves due to the number of articles that contain each search term. As shown by the graph, a small value of k (like those used in previous evaluations [21, 22]) significantly impact the precision-recall curve, particularly for higher recall levels. In particular, the curves become inaccurate above 40% recall. This result mirrors previous findings at TREC [12]. Space prevents us from presenting additional results, but in general, we found that k must be at least double the number of relevant results to ensure accuracy of MAP and interpolated precision.

Table 7: Mondial results; higher scores are better. The left two columns of results are copied from SPARK's evaluation [22].

MONDIAL	Queries 1–35 [22]		Topics 1–50	
	Top-1	R-rank	Top-1	R-rank
BANKS [2]			16	0.358
DISCOVER [15]			31	0.671
Efficient [14]	2	≤ 0.276	21	0.514
Bidirectional [17]			34	0.730
Effective [21]	10	≤ 0.491	22	0.495
DPBF [8]			37	0.823
BLINKS [13]			36	0.770
SPARK [22]	34	0.986	27	0.607
CD [6]			36	0.804

Table 8 presents the normalized minimizing Kendall distance between each system. Each value is averaged over all the datasets and queries; when comparing the results returned by two different systems for a particular query, we use the top-n results where n is the minimum number of results in the two lists. Limiting the comparison to this (variable) number of results follows the precedent of Fagin *et al.* [9] and is necessary because different systems often return a different number of results even for the same query. The purpose of this analysis is to determine if the only important difference between the various systems is performance. Obviously if both systems return similar results, we would prefer the faster search technique. Unfortunately, our results suggest that the systems are only moderately correlated at best. Consider Bidirectional, which addresses performance bottlenecks of BANKS. The correlation between the two systems is only 0.391; their results are similar but no more than other approaches that share a baseline scoring function (e.g., Efficient, Effective, and SPARK). Because the results of the various systems are not highly correlated, the effectiveness of each system must be independently validated.

4.4 Discussion

In part, our evaluation was designed to corroborate the claims of search effectiveness previously presented in the literature. Across all our datasets, we found that our measurements of search effectiveness are considerably lower than those reported in other evaluations. While it is known that these values cannot be directly compared across different document collections [31], we believe that many previous studies have inflated claims of search quality, perhaps due

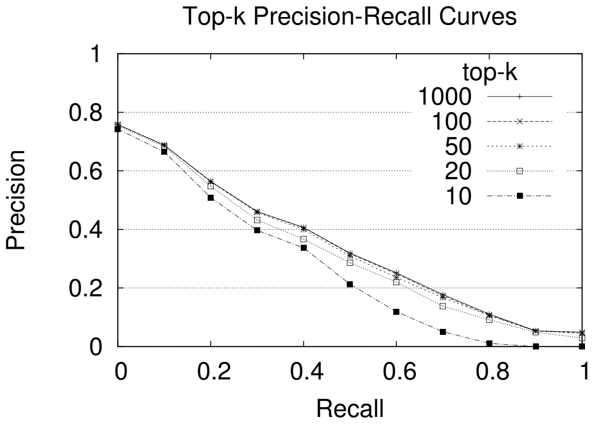

Figure 4: Top-k interpolated precision curves averaged over the systems. The result lists are truncated to contain only k results so the curves for smaller values of k always lie below the curves for larger values of k.

[4] Similar differences may be observed with other evaluations.

Table 6: 11-point interpolated precision and MAP for a subset of the Wikipedia topics. Higher scores are better.

Recall	BANKS [2]	DISCOVER [15]	Efficient [14]	Bidirectional [17]	Effective [21]	DPBF [8]	SPARK [22]	CD [6]
0.0	0.867	0.357	0.925	0.654	1.000	0.492	0.812	0.958
0.1	0.713	0.357	0.870	0.594	0.883	0.377	0.812	0.898
0.2	0.557	0.306	0.834	0.524	0.639	0.296	0.688	0.669
0.3	0.473	0.290	0.739	0.446	0.415	0.267	0.533	0.529
0.4	0.434	0.253	0.696	0.433	0.237	0.234	0.484	0.483
0.5	0.377	0.227	0.523	0.378	0.186	0.179	0.330	0.355
0.6	0.283	0.185	0.475	0.359	0.078	0.157	0.290	0.185
0.7	0.217	0.158	0.363	0.294	0.034	0.081	0.193	0.086
0.8	0.114	0.158	0.227	0.165	0.008	0.060	0.071	0.082
0.9	0.052	0.057	0.116	0.067	0.000	0.002	0.057	0.081
1.0	0.052	0.057	0.112	0.065	0.000	0.002	0.022	0.079
MAP	0.345	0.181	0.518	0.339	0.287	0.179	0.362	0.361

Table 8: Normalized minimizing Kendall distance between each system. Smaller values indicate better correlations. The table is symmetric about the main diagonal.

	BANKS [2]	DISCOVER [15]	Efficient [14]	Bidirectional [17]	Effective [21]	DPBF [8][a]	BLINKS [13]	SPARK [22]	CD [6]
BANKS [2]	0.000	0.507	0.472	0.391	0.695	0.507	0.547	0.524	0.656
DISCOVER [15]	0.507	0.000	0.530	0.576	0.583	0.656	0.530	0.482	0.577
Efficient [14]	0.472	0.530	0.000	0.582	0.374	0.666	0.531	0.366	0.452
Bidirectional [17]	0.391	0.576	0.582	0.000	0.641	0.708	0.605	0.590	0.707
Effective [21]	0.695	0.583	0.374	0.641	0.000	0.727	0.544	0.478	0.478
DPBF [8]	0.507	0.656	0.666	0.708	0.727	0.000	0.581	0.665	0.741
BLINKS [13]	0.547	0.530	0.531	0.605	0.544	0.581	0.000	0.523	0.646
SPARK [22]	0.524	0.476	0.366	0.590	0.478	0.518	0.523	0.000	0.475
CD [6]	0.656	0.577	0.452	0.707	0.478	0.741	0.646	0.475	0.000

to unreported methodological problems such as tuning their systems on the evaluation queries. Webber [32] confirms the trend toward reporting above-average effectiveness scores. The scores of retrieval systems evaluated at TREC and INEX are still considerably lower than ours. Perhaps the size of the collections plays a significant role, for effectiveness on the IMDb dataset lags considerably behind both Mondial and Wikipedia.

Beyond this general characterization of the scores, we see that most of the systems score comparably on each dataset. Overall, there is little that distinguishes any one ranking technique although the IR approaches tend to be less effective than proximity search heuristics. A different system is most effective for each dataset. For the IR-scoring systems, we see no appreciable difference—either in search effectiveness or the set of results retrieved—that would indicate the superiority of any one technique. These results suggest that computationally cheap ranking schemes should be used instead of more complex scoring functions that require completely new query processing algorithms (e.g., those proposed by Luo *et al.* [22]).

Our evaluation illuminates two important issues that should be considered by future work in this field. First, prestige plays an important factor when ranking results. Due to the inclusion of node weights, BANKS [2] and Bidirectional [17] both perform well on the Wikipedia dataset. More recent approaches (as illustrated by DPBF [8]) focus on minimizing the weight of the result tree and perform much more poorly than ranking schemes that incorporate node weights. Given the large number of queries that users make for a specific database entity [25], we consider this approach ill-advised.

Second, our evaluation underscores the scalability issues encountered by systems that require an in-memory data graph to efficiently enumerate results. As stated in Section 2.2, allowing researchers to define arbitrary subsets of datasets for their evaluations may have masked this issue. With the exception of our reimplementation of BANKS, none of the proximity search systems were able to execute any of the IMDb queries. Given that our IMDb dataset is nearly two orders-of-magnitude smaller than the original, scalability issues inherent to the approaches cannot be ignored. Kasneci *et al.* [18] propose storing the complete graph in a database and designing algorithms for this representation, which corrects the immediate problem at the expense of invalidating previous results regarding the algorithms' performance. Dalvi *et al.* [7] use a multi-granular graph representation to alleviate this problem and claim that their technique scales to datasets similar in size to our IMDb subset.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we evaluate the effectiveness of keyword search systems for relational databases. Previous evaluations by researchers have been ad hoc with no standardized datasets or query workloads. The effectiveness of the proposed search techniques is often ignored as researchers focus on the performance aspects of systems. Our evaluation framework is the first designed for this field and provides common workloads for evaluating current and future systems. Standardized evaluation techniques previously enabled rapid progress in the IR community; it is past time for DB&IR researchers to adopt this evaluation paradigm. Such a framework is essential for objectively evaluating many aspects of these systems—including their performance—which depend on the query workload. Our datasets, topics, and relevance assessments are available at `http://www.cs.virginia.edu/~jmc7tp/projects/search/`.

The evaluation presented in this paper is the first to compare a wide variety of IR scoring and proximity search techniques. Our results indicate that no existing scheme is best for search effectiveness, which contradicts previous evaluations that appear in the literature. We also show that the sets of results retrieved by different systems are not highly correlated, which indicates that performance is not the sole factor that differentiates these systems.

In the future, we will expand our evaluation framework to include additional datasets and query workloads. We welcome collaboration with other researchers so evaluation becomes a community effort as it is at TREC and INEX. We also look to reexamine our design decisions for our evaluation (e.g., binary relevance assessments) and to include additional metrics (e.g., normalized discounted cumulative gain (nDCG)) for evaluating search effectiveness.

6. ACKNOWLEDGMENTS

We thank Hristidis *et al.* for providing us with a reference implementation of their system [14]. In addition, we thank Ding *et al.* for providing their implementation of DBPF [8] and He *et al.* for giving us their implementations of bidirectional search [17] and BLINKS [13]. Andrew Jurik, Michelle McDaniel, and the anonymous reviewers all gave helpful comments regarding drafts of this paper.

References

[1] M. K. Bergman. The Deep Web: Surfacing Hidden Value. *The Journal of Electronic Publishing*, 7:1–17, August 2001.

[2] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword Searching and Browsing in Databases using BANKS. In *ICDE '02*, pages 431–440, February 2002.

[3] Y. Chen, W. Wang, Z. Liu, and X. Lin. Keyword Search on Structured and Semi-Structured Data. In *SIGMOD '09*, pages 1005–1010, June 2009.

[4] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and Diversity in Information Retrieval Evaluation. In *SIGIR '08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–666, July 2008.

[5] C. Cleverdon. The Cranfield Tests on Endex Language Devices. In *Readings in Information Retrieval*, pages 47–59. 1997.

[6] J. Coffman and A. C. Weaver. Structured Data Retrieval using Cover Density Ranking. In *KEYS '10*, pages 1–6, June 2010.

[7] B. B. Dalvi, M. Kshirsagar, and S. Sudarshan. Keyword Search on External Memory Data Graphs. *PVLDB*, 1(1):1189–1204, 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 '07*, pages 836–845, April 2007.

[9] R. Fagin, R. Kumar, and D. Sivakumar. Comparing Top k Lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, 2004.

[10] N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas. INEX: Initiative for the Evaluation of XML Retrieval. In *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval*, August 2002.

[11] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword Proximity Search in Complex Data Graphs. In *SIGMOD '08*, pages 927–940, June 2008.

[12] D. Harman. Overview of the Second Text REtrieval Conference (TREC-2). *Information Processing & Management*, 31(3):271–289, 1995.

[13] H. He, H. Wang, J. Yang, and P. S. Yu. BLINKS: Ranked Keyword Searches on Graphs. In *SIGMOD '07*, pages 305–316, June 2007.

[14] V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient IR-style Keyword Search over Relational Databases. In *VLDB '03*, pages 850–861, September 2003.

[15] V. Hristidis and Y. Papakonstantinou. DISCOVER: Keyword Search in Relational Databases. In *VLDB '02*, pages 670–681. VLDB Endowment, August 2002.

[16] B. J. Jansen and A. Spink. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. *Information Processing and Management*, 42(1):248–263, 2006.

[17] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional Expansion For Keyword Search on Graph Databases. In *VLDB '05*, pages 505–516, August 2005.

[18] G. Kasneci, M. Ramanath, M. Sozio, F. M. Suchanek, and G. Weikum. STAR: Steiner-Tree Approximation in Relationship Graphs. In *ICDE '09*, pages 868–879, March 2009.

[19] M. Kendall and J. Gibbons. *Rank Correlation Methods*. Charles Griffin, 1948.

[20] 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 '08*, pages 903–914, June 2008.

[21] F. Liu, C. Yu, W. Meng, and A. Chowdhury. Effective Keyword Search in Relational Databases. In *SIGMOD '06*, pages 563–574, June 2006.

[22] Y. Luo, X. Lin, W. Wang, and X. Zhou. SPARK: Top-k Keyword Query in Relational Databases. In *SIGMOD '07*, pages 115–126, June 2007.

[23] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, 2008.

[24] W. May. Information Extraction and Integration with Florid: The Mondial Case Study. Technical Report 131, Universität Freiburg, Institut für Informatik, 1999.

[25] A. Nandi and H. V. Jagadish. Qunits: queried units for database search. In *CIDR '09*, January 2009.

[26] G. Pass, A. Chowdhury, and C. Torgeson. A Picture of Search. In *InfoScale '06*, May 2006.

[27] L. Qin, J. X. Yu, and L. Chang. Keyword Search in Databases: The Power of RDBMS. In *SIGMOD '09*, pages 681–694, June 2009.

[28] A. Singhal. Modern Information Retrieval: A Brief Overview. *IEEE Data Engineering Bulletin*, 24(4):35–42, December 2001.

[29] Q. Su and J. Widom. Indexing Relational Database Content Offline for Efficient Keyword-Based Search. In *IDEAS '05*, pages 297–306, July 2005.

[30] TREC Overview. `http://trec.nist.gov/overview.html`.

[31] E. M. Voorhees. The Philosophy of Information Retrieval Evaluation. In *CLEF '01*, pages 355–370. Springer-Verlag, 2002.

[32] W. Webber. Evaluating the Effectiveness of Keyword Search. *IEEE Data Engineering Bulletin*, 33(1):54–59, 2010.