This article provides a technical review of semantic search methods used to support text-based search over formal Semantic Web knowledge bases. Our focus is on ranking methods and auxiliary processes explored by existing semantic search systems, outlined within broad areas of classification. We present reflective examples from the literature in some detail, which should appeal to readers interested in a deeper perspective on the various methods and systems implemented in the outlined literature. The presentation covers graph exploration and propagation methods, adaptations of classic probabilistic retrieval models, and query-independent link analysis via flexible extensions to the PageRank algorithm. Future research directions are discussed, including development of more cohesive retrieval models to unlock further potentials and uses, data indexing schemes, integration with user interfaces, and building community consensus for more systematic evaluation and gradual development.

Introduction

Semantic search (Baeza-Yates, Ciaramita, Mika, & Zaragoza, 2008) is considered by many as the natural evolution of current search technology. Although many conventional retrieval models have been proved to work effectively and efficiently over coarse document collections, there are many inherent obstacles to overcome when focus starts to shift toward items of finer granularity. Arguably, current search technologies are hindered by their limited understanding of user queries and ability to reason with more complex information requests requiring restrictions and finer specifications at the level of objects. Product search is a typically cited example of this realm. Traditional approaches to information retrieval (IR) often treat documents as collections or bags of individual words, and their correspondence to a similar representation of user queries generally determines their level of similarity. This notion has often been coupled with simple forms of natural language processing (Baeza-Yates, 2004) and features based on links, such as popularity and usage, when search is conducted over web-accessible documents. More elaborate retrieval models have also evolved in an effort to include information related to the classification of content inside documents, to prioritize selections based on where query terms are found within the documents (whether part of a title, body, anchor text, etc.). The idea of semantic search is to diverge from this coarse view and sometimes monotonic treatment of documents to a finer perspective, one that is able to exploit and reason intelligently with granular data items, such as people, products, organizations, or locations, whether that is to complement document retrieval or to facilitate different forms of search.

The advent of the Semantic Web (Berners-Lee, Hendler, & Lassila, 2001) is seen as an appealing vision for achieving deeper and better integration of data and information and, consequently, better understanding of the constructs and challenges for working in a semantic environment. Making searches semantic is about operating in an environment in which symbols, documents, and other resources are given well-defined meaning. The Semantic Web is about exposing structured information on the web in a way that its semantics are grounded on well-defined and agreed-upon vocabularies. Through the established efforts of a number of online communities, there is now a large corpus of structured data in various formats (Resource Description Framework
Semantic search is a dynamic area of research. The application area and realization of different approaches have been very diverse and sometimes even lacking a common set of ideas. Information search utilizing Semantic Web and other data graphs raises many challenging issues, including the modeling of queries and the definition of “documents” in response to queries. To these ends, there is widespread research covering developments across several distinct areas that do not necessarily coincide, although they can rightfully be classified under the overall realm of semantic search. Some of the more established areas have received considerable attention and have been the focus of several academic conferences and workshops. We can identify mainstream research associated with a number of areas based on the orientation and focus of different developments, including:

- Document-oriented search, in which the focus is on retrieval of documents but using various ontological techniques to enhance document retrieval, such as works that explore the combination of semantic metadata and other document features to improve retrieval performance or augment document lists with relevant data pulled from the Semantic Web (Fernandez et al., 2008; Guha, McCool, & Miller, 2003; Han & Chen, 2006; Vallet-Weadon, Fernández-Sánchez, & Castells-Azpilicueta, 2005).
- Multimedia search, in which formal representations of domain ontologies and semantic annotations are used for indexing and searching digital multimedia content, such as audio recordings, images, and movies (Celino, Valle, Cerizza, & Turati, 2006; Ding, Yang, Li, Wang, & Wenyin, 2004; Linckels, Repp, Karam, & Meinel, 2007; Wei & Barnaghi, 2007). (Multimedia search may be thought of as a special case of entity search, except that indexable features are usually the product of special processing peculiar to digital content, such as speech recognition, collaborative tagging, or segment detection.)
- Association search, in which the focus is on discovery and interpretation of direct and indirect associations between resources (Sheth, Arpinar, & Kashyap, 2004). (The motivation here is that complex relationships can capture the meaning of resources and being able to extract the most obscured relations can provide essential information. Potential uses have been realized in a number of areas, including national security applications, such as being able to determine whether a flight passenger is known to be associated with an organization on the watch list [Sheth et al., 2005].)
- Entity-oriented search, in which the focus is on direct retrieval of resources at the granularity of objects, such as products, people, and organizations, which is a very active area of research, capturing developments that span a wide range of activities, from simple keyword and parameterized query algorithmic solutions to more elaborate design models for iterative and exploratory search.

Entity search is a well-documented theme in the literature and is being actively addressed by both the IR and Semantic Web communities. The Semantic Web community has recently organized the Semantic Search Challenge,2 aiming to prioritize and evaluate research into “ad-hoc object retrieval” utilizing Semantic Web graphs (Halpin et al., 2010; Pound, Mika, & Zaragoza, 2010). The outcome from the series has been a standard reference collection for conducting and evaluating semantic search experiments. The collection focuses largely on conventional web queries and provides assessments over a sizable data set of 1 billion triples. Outside mainstream Semantic Web research, the theme has appeared in a number of research tracks at the celebrated TREC and INEX IR conference series. The expert search task of the TREC Enterprise Track, initiated in 2005 (Balog, Fang, de Rijke, Serdyukov, & Si, 2012), and the more recent TREC Entity Track (Balog, Serdyukov, & Vries, 2010) and INEX Entity Ranking Track (Demartini, Iofciu, & de Vries, 2010) deal with searches at the entity level. These focus largely on entities represented as “pseudodocuments” composed of virtual organizations of content from Wikipedia and other homepages. In the database community, keyword- and natural language-based

---

1Linked Data: http://linkeddata.org

2Semantic Search Challenge: http://semsearch.yahoo.com
search in databases is again a historic theme in the literature (Chen, Wang, Liu, & Lin, 2009).

Article Scope and Motivation

The present article provides a technical review of entity search methods applied over formal Semantic Web knowledge bases. Our focus is on ranking approaches and methodologies explored in the literature, with particular emphasis on methods that make use of the graph structure of Semantic Web data. Ranking models have been an integral part of IR research and remain an active and challenging dimension in modern frameworks and data models. Throughout this review, we seek to obtain a deeper understanding of the architectural choices that play a role in supporting text-based search over Semantic Web data. For this reason, we focus on presenting a few topics in some detail. The survey is not intended to be an exhaustive list of available architectures, but rather a detailed outline of reflective examples from the literature.

Ranked keyword search over graph-structured data has attracted much attention recently for a number of reasons. Keyword-based search tools generally do not require users to master a complex query language or understand the underlying data schema to be able to interact. In effect, they are a very attractive frontier for research into scalable semantic search engines that can cope with multiple heterogeneous data collections. Furthermore, ranked keyword search can generally function as the starting point for further exploration and search, and users have grown accustomed to this setting. Even complex systems based on articulated interfaces often require an initial starting point for users to engage in further interaction. Keywords can be used to pinpoint objects of interest, after which a system can provide additional menus and filters to reduce incrementally the size of the results or to construct more expressive queries. It remains vital, therefore, that an effective and flexible retrieval model is available for the core functionality of a system, whether that is to be treated as a stand-alone facility or part of a larger complex of tools.

This review maintains a strong Semantic Web orientation; it is prevalent throughout the material selected for review. However, the techniques outlined are conceptually, and sometimes pragmatically, applicable to any type of data that pertains to a graph structure, particularly directed labeled graphs, as is outlined in the next section. The paper takes on a holistic view of developments in this area, both across the Semantic Web and also as supported by works in similarly related fields, for example, relational database and XML search. The selection of works has been driven by the availability of detailed and complete descriptions and the need to capture a wide spectrum of techniques and architectural frameworks. The presentations follow a common outline: a detailed description of a characteristic operation is presented, followed by reflective examples of individual systems that explore or implement the operation in a given context. We give special emphasis to the evaluation procedures followed to demonstrate the performance of individual systems and any coupling involved with other methods to facilitate overall retrieval.

Search Context: Data and Challenges

The Semantic Web

The Semantic Web is an extension of the current web that aims to underpin web resources with machine-understandable data in order to optimize sharing, reuse, and general handling of information. The infrastructure of the Semantic Web is a proposed set of standardized technologies to handle effectively the global identification, modeling, and querying of semistructured data resources. The RDF forms the foundation of data modeling languages on the Semantic Web and provides a syntax that allows the use of uniform resource identifiers (URIs) to name resources. RDF is a flexible, graph-based data model and provides a foundation for more advanced and expressive assertional languages.

Semantic Web data are maintained within special information repositories known as knowledge bases and are made publicly available either in the form of raw data files or via triple stores, which provide functionality similar to that of ordinary RDBMSs. The underlying building block of a knowledge base is a subject–predicate–object triple. A subject is the identifier of a resource (an entity), a predicate the identifier of a relation, and the object either the identifier of another resource or a concrete value, such as a string literal or some other primitive data value. One can conceptualize a knowledge base as a loosely coupled directed labeled graph (DLG), in which subjects and objects are treated as nodes and predicates as labeled edges (relations) between them. DLGs are a common and generic model for describing possibly any type of semantic network or association graph. On the Semantic Web, relations are first-class URI resources and can be defined locally or reused from existing vocabularies. A knowledge base is formally divided between a definition schema, comprising the terminological basis of the data, and the actual instance data providing an instantiation of the conceptual schema.

Intrinsic Technical Problems

Dealing with searches over Semantic Web data raises many issues. The definition of a document in conventional IR is now projected onto entities that may be connected to a multitude, and possibly a nondeterministic set, of object- and data-type relations. In addition to the semantic matching of keywords to ontology concepts and other RDF literals (a topic central to IR research in general, requiring disambiguation and expansion of polysemous words and phrases), a search process has to interpret and utilize the graph structure. Even for simple queries, a sparsely distributed network may require evidence to be traversed in the graph until an association with candidate resources can somehow be established. Query evidence may be connected to relevant objects
but not directly to the resources sought. Additionally, the presence of explicit semantics in the data uncovers functionality that can lead to more expressive query construction, essentially allowing queries of more complex graph patterns to match, for example, queries pertaining to multiple triple patterns with variable restrictions on types and attributes, such as “Mel Brooks movies starring J. Silberman.” Systems may opt to exploit this potential for hybrid or semistructured query capabilities.

Key to success when dealing with ambiguous keyword queries remains the effectiveness of the ranking produced by an algorithm and its degree of portability, irrespective of any further processing incurred by a system. For example, an effective model restricted to a specific domain will face deficiencies when ported onto new data sets from the ever-increasing web of data. Similarly, a very efficient and portable algorithm with deficiencies in its ranking cannot satisfy the expected utility of end users. In addition, a large-scale semantic search engine will have to cope with a very large and complex space of distributed knowledge bases on the web, imposing hard scalability and performance restrictions. A good balance between effectiveness, efficiency, and portability across domains is a necessary commitment for successful implementations.

Ranking Approaches

Query Graph Construction and Exploration Methods

Keyword query processing over graph-structured data has emerged as an important research topic in the wider field of database research. A considerable amount of research reported in the literature focuses on adapting keyword search to relational and XML databases, which can also be portrayed as graphs or trees. In this section, we look at various techniques that interpret keyword queries as substructures of a graph and apply various heuristics to estimate the relevance of each substructure. Our focus is on methods applied to Semantic Web data, although we start by looking at earlier works dealing with conventional databases. The two are in fact very similar, and the former may appear as extensions to earlier works.

Conceptually, databases can be regarded as graphs or trees, with nodes resembling tuples or XML elements and edges resembling foreign-key relations (w.r.t. relational databases) or element contains and IDREF/ID links (w.r.t. XML databases). Techniques that operate directly on XML data are very popular in the literature, although most depend on tree-structured data (Cohen, Mamou, Kanza, & Sagiv, 2003; Florescu, Kossmann, & Manolescu, 2000; Guo, Shao, Botev, & Shammugasundaram, 2003). In a typical scenario, an algorithm computes minimal-cost connected trees as answers to a query. Techniques that focus on relational databases consider a graph orientation and are thus more related to the Semantic Web, which is inherently graph based.

Database techniques. There is a large body of work dealing with keyword searches inside databases. These are generally divided between schema-agnostic techniques that operate directly on data and database extensions that require a database schema. Popular methods focus on finding a minimal subgraph/tree in the network that connects all the nodes matching the keyword elements. BANKS (Bhalotia, Hulgeri, Nakhe, Chakrabarti, & Sudarshan, 2002), for instance, is a popular schema-agnostic architecture that employs a backward search algorithm starting from the nodes containing at least one query keyword and iteratively traverses incoming edges until a connecting answer root is reached. The answer to a query becomes a rooted directed Steiner tree (Dreyfus & Wagner, 1971) containing a directed path from the root to each keyword node. The model comprises a combination of relevance clues from nodes to edges, including heuristics to measure the prestige of nodes as a function of their in-degree and edge weights reflecting the strength of relationships (proximity) between tuples. Kacholia et al. (2005) propose an extension to BANKS considering bidirectional propagation factors, for example, methods to traverse the graph both backward from keyword nodes and forward from potential roots. This has the effect of finding more efficiently potential roots in the network; it was proved that fewer iterations were needed, so the model can deal with situations when query keywords match a very large number of nodes. As an extension to this, He, Wang, Yang, and Yu (2007) introduced a novel indexing scheme using block-based partitioning to improve the efficiency of bidirectional graph exploration. Similar popular approaches are presented as database extensions in DBXplorer (Agrawal, Chaudhuri, & Das, 2002) and DISCOVER (Hristidis & Papakonstantinou, 2002); these operate on the schema graph of databases and hence rely heavily on the database schema and the infrastructure of the underlying RDBMS.

Semantic Web techniques. Recent studies on the Semantic Web have been motivated by similar ideas. The general focus is on the computation of conjunctive queries from keywords using Semantic Web data. Zhou, Wang, Xiong, Wang, and Yu (2007) explore a process for automatically translating keyword queries into formal logic queries via a prototype system known as SPARK. Given a keyword query, SPARK maps the keywords to various knowledge base constructs and outputs a ranked list of SPARQL equivalents, which the user can choose to execute. The process (illustrated in Figure 1) starts with keywords being enumerated into several combinations and mapped to resources in the knowledge base; a series of morphological and semantic processing steps (string comparisons and synonym expansion using the WordNet electronic lexicon) facilitates the mapping and assigns a confidence value to each mapped keyword. The graph construction phase takes as input the mapped resources, splits them into different query sets via further enumeration, and applies a minimum spanning tree algorithm to construct possible query graphs from each
query set. The output query graphs are essentially a set of candidate SPARQL queries to be ranked before being presented to the user.

 Ranking in SPARK is driven by a combination of diagnostic probability estimates for each candidate formal query. Precisely, it is defined as:

\[ p(F|D, Q) = p(F|Q) p(F|D) \]

assuming independence between the relevance of formal query, \( F \); to the knowledge base, \( D \); and the keyword query, \( Q \). The function \( p(F|Q) \) incorporates the confidence values of each mapped keyword and the overlap of \( F \) with the original query. The likelihood \( p(F|Q) \) considers the information content of a formal query as a measure of the relative frequency of its relations as occurring in \( D \). The model is flexible to parameterization, offering users the option to adjust the ranking via a slider on a sigmod function, such as to favor frequent versus infrequent relations.

SPARK was evaluated on a set of manually constructed knowledge bases and translated queries from the Mooney Natural Language Learning Data. This involved a set of 250 translated keyword queries and SPARQL equivalents for each of the three ontologies involved. Evaluation focused on recall and mean reciprocal rank, which are sensible measures given the system’s orientation to few results per query. An additional user study was conducted over a 2-month period involving 50 test users. Results indicate that the model works best with medium-sized or short queries (2–6 terms); more complex natural language queries proved too ambiguous to understand and translate, for example, queries involving negation, superlative forms, and other value constraints.

In a similar study, Tran, Wang, Rudolph, and Cimiano (2009) also extend the notion of query graph construction to answer sets that are not restricted to trees but that can be graphs in general. In this approach, keywords are interpreted as both vertices and edges to allow better reasoning with more complex queries, for example, “authors working at Stanford University that have won a Turing Award.” The knowledge base is pre-indexed into an inverted index of keyword-element mappings and a summary graph, which captures relations between classes and instances into a graph index via type and subsumption information. The aim of a summary graph is to reduce the solution space to a more concise equivalent for more efficient exploration. The rest of the process for top-k query computation is summarized in the following steps.

1. Map query keywords to elements of the data graph (literals associated with nodes and edges).
2. Explore the data graph by traversing paths from the keywords to potential connecting elements.
3. Merge paths that meet at connecting elements to construct a set of matching minimal subgraphs.
4. Rank matching subgraphs to produce a top-k query answer set.

The computation process can result in multiple subgraphs corresponding to several possible interpretations of the keywords. Results from the process are effectively a set of matching structured queries, which the user can choose to execute and retrieve the answers individually. The relevance of computed queries is assessed via a combination of cost functions, defined as a monotonic aggregation of scores derived from the paths in a graph. Precisely, a cost function has the form:

\[ \sum_{p \in P} \sum_{n \in P} c(n) \]

where \( P \) is the set of paths in the answer graph, and \( n \) is an element to be associated with a specific cost. The authors experiment with path lengths (favoring graphs with entities closer together), popularity scores (simple metrics to favor larger graphs), and keyword-matching scores (incorporating both syntactic and semantic similarities using WordNet). The precise implementation is based on the threshold algorithm, except that lower bounds correspond to highest costs and upper bounds to lowest costs. Experiments over the DBLP\(^3\) data set and the TAP ontology by Stanford University concluded that keyword-matching scores were the most prevalent factor, with superior results in all cases. The evaluation focused on the mean reciprocal rank over a set of 39 queries constructed and assessed by 12 participating users. It remains unclear, however, whether combinations of cost functions were indeed assessed and what the best combination would be.

---

\(^3\)DBLP Computer Science Bibliography: http://www.informatik.uni-trier.de/~ley/db

FIG. 1. Query graph construction process. From simple keywords to a set of candidate query graphs. Simplified reproduction from Zhou et al. (2007).
A closely related work, though still in its initial architectural stages, has been described by Parthasarathy, Kumar, and Damien (2011). The authors experiment with type and subsumption information, but this time exploited to traverse the data graph and construct an initial set of matching subgraphs. Then, a set of pruning and hooking heuristics is introduced to merge subgraphs together. Pruning eliminates loosely hanging nodes, and everything that remains can potentially be mapped or merged across pairs of graphs. The outcome may be multiple answer graphs, and ranking becomes essential to order the results. The authors consider heuristics to estimate the structural compactness of the elements in the output graphs, the textual relevance of keywords to the nodes mapped, and the relevance of nodes and edges. We refrain from giving further details, because the method has not yet been evaluated.

**Spreading Activation**

Spreading activation is a popular technique used traditionally in psychology to study human memory phenomena and operations, such as retention and recall of cognitive units of memory (Anderson, 1983). The framework has been widely adopted in other fields in which semantic or associative networks are the primary form of knowledge representation, with several applications in IR (Crestani, 1997). The algorithm provides a basic inference solution to network data structures in which concepts are treated as nodes and relationships as weighted or labeled arcs between them. The intuition is a fairly simple one: Given an initial activation value for a set of nodes, spreading activation will traverse the network iteratively and spread the activation values to neighboring nodes. There are possibly many different processing techniques, restrictions, and decay conditions that can be applied, but the general idea is that, when propagation halts, each and every node in the network will be activated with a certain value.

In its basic form, we may define the input $I_j(t_i+1)$ of node $j$ at time $t_i+1$ to be the sum of the outputs of the nodes that connect to it, weighted by the type of relation that holds between them.

$$I_j(t_{i+1}) = \sum_k O_k(t_i) \times w_{k,j} \times (1-a), \quad a \in (0,1)$$

(3)

where $O_k(t_i)$ is the output of node $k$ and $w_{k,j}$ is the weight of the relation. It is common to associate a loss function, $a$, with the propagation, such that it gives preference to shorter paths in the network. Spreading activation conveys an attractive formalism for processing query evidence across Semantic Web graphs. The following systems have both used spreading activation in a similar form to develop their inference processes. Different techniques are applied to associate weights with relations.

**Examples of Spreading Activation Use**

Jiang and Tan (2006) present a unique prototype solution, OntoSearch, that combines ontology-based inference with classical keyword-based methods at query time for retrieval. The work focuses on a collection of semantically enriched documents, although the algorithm is conceptualized at the entity level and so is presented here as an entity type of ranking model.

Resource URLs in OntoSearch correspond to instance entities and are treated as compound vectors of keywords and concepts. Keywords constitute the textual descriptions of resources (what would be equivalent to a label), and concepts assume taxonomical ontology classes (concepts related to resources via some type of instantiation edge). The method uses a TF-IDF measure to assign weights to keywords and binary values to indicate a concept’s association with the corresponding resource. Upon arrival of a query, the system uses the submitted query terms to retrieve an initial list of resources via a keyword-based search method. The concepts associated with the retrieved resources are then seeded into a spreading activation algorithm to infer more concepts that are semantically related to the original set. The outcome of the algorithm formulates a compound query vector with keywords and weighted concepts (concepts activated by spreading activation). OntoSearch utilizes the relative frequency of properties to determine the weights used in spreading activation. Ranking involves a straightforward dot product of the resource vectors (which remain intact) and the extended query. OntoSearch extends spreading activation with personalized views of a domain in the form of user ontologies encoding relevance feedback provided for past queries. These are factored into the concept weights, assuming a time decay factor based on the interval between queries.

OntoSearch was empirically evaluated on a small collection of academic publications from the ACM Digital Library. The ACM Computing Classification System\(^4\) terms were used to index the documents with taxonomic information, assuming the data set’s underlying ontology for the experiments. The evaluation involved a user study to extract relevance assessments for the retrieved documents and compare the system against a conventional keyword-based search engine (Lucene). Although no strong indication of the statistical significance of the tests is apparent, OntoSearch outperformed Lucene in terms of average precision on a set of 30 test queries. Performance appeared to be considerably higher at low recall levels, although the two approaches naturally decreased and converged at high recall levels. Usage of a user ontology indicated improvements over the baseline method for three of five users.

A very similar approach, combining spreading activation with traditional keyword processing, has been described by Rocha, Schwabe, and Aragao (2004). One of the main ideas explored was how to extract information from the link structure of knowledge bases to associate weights with object relations. The authors combine two measures, namely, cluster and specificity, and use a hybrid spreading activation technique that combines numerical weights with the labels of properties.

\(^4\)ACM Computing Classification System: http://www.acm.org/class
The cluster measure is treated as an asymmetric estimate and attempts to establish the degree of similarity between two related instances. The algorithm is a straight adaptation of the clustering function developed by H. Chen and Ng (1995) for constructing association networks from term co-occurrence rates in documents. The measure interprets the similarity of two entities, \( D_i \) and \( D_k \), as the ratio of their intersection with other entities in a knowledge base, relative to the event space of either of the two entities. Let \( N_{i,j} \) denote the event that \( D_i \) is related to \( D_j \), taking on values from the set \{1,0\} (indicating whether the event is true or false), and \( N_{i,k} \) denote the event that \( D_i \) and \( D_k \) are both related to concept \( D_j \), again taking on values from the set \{1,0\}. Considering a knowledge base with \( n \) entities, the similarity of \( D_i \) and \( D_k \) relative to \( D_j \) is given as:

\[
W(D_i, D_k) = \frac{\sum_{i=1}^{n} N_{i,j} \cdot N_{i,k}}{\sum_{i=1}^{n} N_{i,j}}
\]

which is a probability estimate between 0 and 1. Note that the equation is asymmetric: switching the denominator to sum over \( N_{i,k} \) instead of \( N_{i,j} \), the intersection becomes relative to \( D_k \). This is the main characteristic of the cluster function, with implications for semantic or associative networks in which directed arcs establish connections between nodes.

The specificity measure is similar to the IDF convention and is used for discriminating against very common relations. The measure is inspired by the work of Stojanovic, Studer, and Stojanovic (2003) on differentiating property instances based on their utility in knowledge bases. The specificity of a relation, \( r \), between two instances \( D_i \) and \( D_k \) is given as:

\[
W_r(D_i, D_k) = \frac{1}{\sqrt{n_k}}
\]

which is inversely proportional to the number of instances \( n_k \) that link to \( D_k \) via the given relation. The measure is asymmetric and interprets how specific the destination concept is. The result saturates over increasing values of \( n_k \).

Combining weights with labeled arcs involves assigning additional manual weights to properties; hence, spreading activation is extended with an extra weighted factor in each propagation. These can be relative weights for fine-tuning propagation in a network; for example, zero-weighted properties can clamp a network and not allow propagation to flow through the edge, whereas higher weights can be associated with more important properties.

The ranking process is similar to OntoSearch. Results from an initial keyword-based search using Lucene are supplied to the spreading activation algorithm, and the initial ranking defines the activation values of nodes. The outcome may be a reordering or expansion of the initial results list or a new set of results altogether. There is no refactoring of query input after the spreading activation process halts (as is done in OntoSearch). The proposed algorithm lacks empirical evaluation with a baseline method, but a qualitative analysis from domain experts indicated promising results on two separate implementations. It was observed that many relevant results would have been possible only through an otherwise complicated manual chaining of queries.

### Classic Probabilistic Retrieval Models

Probabilistic models in IR have been integral for reasoning with uncertainty in a wide range of tasks. Some of the earliest and pioneering techniques in the field were designed around models that base their core assumptions on rudimentary probabilistic and Bayesian principles, such as the binary independence and language modeling approaches (Croft, Metzler, & Strohman, 2009). Uncertainty is an intrinsic problem in IR. A major difference between IR systems and other information systems is the lack of query formulation that can represent uniquely an information need and a clear procedure to decide whether an object from a knowledge base is a correct answer. Probability theory has been the most well-studied paradigm for modeling solutions to IR, with the more successful frameworks serving as extensible solutions on which more complex models have evolved.

A modern textbook on IR typically offers extensive coverage of probabilistic models, which can range from early principled approaches (dating from the early 1960s) to more abstract inference network models serving as generalization frameworks. This section presents coverage of two widely adopted models that motivated recent experimental developments in Semantic Web search.

#### Language Model

Language models are a general formal approach to IR, with many variant realizations (Croft et al., 2009; Zhai, 2008). In their most common use, they are known as query likelihood models; the definition stems from the use of probabilistic reasoning to measure the likelihood that a query can materialize given a document specification. Effectively, the method associates a probability distribution over the occurrence of words in the index vocabulary of a collection. A document specification becomes a sampling of words from the distribution, and the goal is to measure how likely it is that a document is about the same topic as the query. Language models provide a generic Bayesian interpretation to the relevance of queries and documents, with the general form:

\[
p(D|Q) \propto p(Q|D)p(D)
\]

for a query, \( Q \), and a document \( D \). The likelihood that a query is relevant to a document is usually treated with naïve term independence assumptions, as in:

\[
p(Q|D) = \prod_{w \in Q} p(w|D)
\]

whereas the document prior is seen as a useful parameter for introducing additional criteria to favor documents with special features. The diagnostic support accorded to a
where the probability distributions are over the set $P_D$ where

$$p(w|D) = (1 - \lambda) p(w|C) + \lambda p(w|D)$$  \tag{8}$$

where $p(w|D)$ and $p(w|C)$ usually translate to the relative frequency of term $w_i$ in document $D$ and across the entire collection, $C$. The smoothing parameters $\lambda \in (0,1)$ are usually constant to the current document. Language models typically associate separate probability distributions with queries and documents, and the Kullback-Leibler (KL) divergence is used to compare the two models in terms of how close they are to each other, that is, the relative entropy or information gain from one to the other. Documents (or result graphs) can then be ranked in increasing order of the KL divergence. Assuming $P_Q$ and $P_D$ to be the probability (likelihood) distributions associated with a query and a document, respectively, the KL divergence is given as:

$$KL(Q||D) = \sum_T P_Q(T) \log \frac{P_Q(T)}{P_D(T)}$$  \tag{9}$$

where the probability distributions are over the set $T_i = \{t_1, \ldots, t_n\}$ of $n$ terms in the corpus.

Language models are in general based on a very intuitive and extensible framework. The specification of a document is masked by a simple aggregation of individual scores (Equation 7, avoiding complex term codependencies), which are adjusted on a finer scale by weighting different sources of information (Equation 8, collection-wide and document-centric information). This serves as an interesting formalism for wider adoption of the model.

Elbashassiouni, Ramanath, Schenkel, Sydow, and Weikum (2009) investigate the use of a language model to rank results to triple-based query patterns, whereby queries are treated as either purely structured or keyword-augmented triple patterns. The method fundamentally extends the notion of documents in traditional IR to a large, all-encompassing graph of triples. A query, $Q$, is treated as an $n$-triple pattern (or relaxed pattern with variable predicate matches) and any subgraph of $n$ triples from the knowledge base is considered a potential result graph to the query (essentially assuming the role of a document in traditional IR). The method uses the relative frequency of individual triples (as opposed to terms) to approximate their marginal likelihood contribution in both exact triple matches and keyword-augmented queries. The authors refer to this as the relative “witness count” of triples. They do not appear to account for the within-triple frequency of terms, which in turn is surprising, given that frequency values are taken into account in the outlined keyword indexes. Realistically, data type relations can be associated with more verbose literal values, such as the case of labels and descriptions. Term frequency (e.g., in terms of TF-IDF) is an important property of the extent to which a value treats the subject referred to by a term.

The proposed method was suitably evaluated on two data sets and benchmarked against three other approaches. The experiments utilized a subset of IMDb\(^5\) and LibraryThing\(^6\) (a catalogue and forum of books), and the competitors included the web object retrieval (WOR; Nie, Ma, Shi, Wen, & Ma, 2007), BANKS (Kacholia et al., 2005), and NAGA (Kasneci et al., 2008) methods. These are similar methods that operate on structured data at the entity level and use different types of graph analytics to rank results. The evaluation involved a user study to estimate the relevance of results produced by each of the contestants. The outlined methodology outperformed the other methods on both data sets in terms of nondiscounted cumulative gain. It remains unclear, however, whether the proposed strategy can operate effectively over crisp RDF-centric knowledge bases. The authors used a customary search engine to approximate the witness counts of triples, as required by their model. Consequently, a ranking procedure tied to an external search engine may not be portable or effective over self-contained RDF knowledge bases in which triples are expected to be distinct.

Balog, Ciglan, Neumayer, Wei, and Nørvaag (2011) used the language model as part of their competing system at the Semantic Search Challenge in 2011. Their main experiments involved an extension to the model to contribute field-level scores to the representation of the entities being evaluated. This was fairly straightforward to achieve, given the vague specification of term probabilities in the model. The individual scores of terms were projected onto field-specific dependencies adjusted by a prior score reflecting the importance of each field considered ($f$).

$$p(w|D) = \sum_{f \in F} p(w|D_f) p(f)$$  \tag{10}$$

The individual term probabilities, $p(w|D_i)$, were then smoothed by Dirichlet priors as normal, except using field-specific and entity-level information, more specifically, functions to incorporate the length of each field being considered and field/entity-specific background models. The authors further explored propagation heuristics to communicate the individual scores of entities to connected entities via sameAs relations extracted from DBPedia.\(^7\)

Similarly, the authors of WOR (Nie et al., 2007) applied the language model at the level of web objects, whereby an object was defined as a collection of database records of multiple attributes or fields aggregated from multiple web sources. The authors experimented with variations of the model based on different levels of granularity of objects. In their best approach, individual term probabilities were extended by an additional dimension, incorporating the various possible object representations from multiple sources or records.

\(^5\)Internet Movie Database: http://imdb.org

\(^6\)LibraryThing: http://www.librarything.com

\(^7\)DBPedia: http://dbpedia.org
where the prior of individual fields, \( p(f) \), was treated as a smoothing function incorporating the importance of the field and the accuracy of the field extraction phase. Similarly, the prior of a record representation, \( p(r) \), was used to incorporate the accuracy of record detection.

**BM25F**

BM25F (Robertson & Zaragoza, 2009) is a state-of-the-art technique for structured document retrieval. The method was originally conceived in 1976 as a simple probabilistic model, known as the binary independence retrieval (BIR) model and was designed chiefly to integrate user feedback information into a ranking formalism. The original assumption was that documents can be classified between relevant and nonrelevant sets and that terms are distributed differently within the two sets. In the absence of relevance information, the model encloses a ranking function that works similarly to a TF-IDF hybrid, in the sense of adopting collection-wide and document-centric term occurrence statistics. The BIR model, also known as Okapi BM25, was later extended to manage structured document retrieval (in particular, S. Robertson formalized the method in 2004 [Robertson, Zaragoza, & Taylor, 2004]) by extending its ranking functions to multiple weighted fields, as opposed to flat documents, for example, by weighting occurrences of terms in the title, body, or anchor text of web pages. In general, the newest version of the model, BM25F, is known to improve retrieval effectiveness by using nonlinear frequency saturation functions, document and field length normalization, and field weights for structured IR. This entails a rather lengthy list of tuning parameters. In particular, \( 2k + 1 \) parameters for \( k \) fields must be estimated per collection for the model to reach its optimum potential. Parameter optimization in BM25F is a heavy experimental process, requiring training data sets with possibly large volumes of queries and assessments.

BM25F is most common in the literature as a precise ranking function and not an extensible model, as the case would more naturally be for the language model. A few recent studies on semantic search have used BM25F for entity-oriented search, whereby Semantic Web resources are explored primarily at the level of data type information.

Pérez-Agüera, Arroyo, Greenberg, Iglesias, and Fresno (2010) designed an experiment in which entity resources are generalized as structured documents consisting of five fields: all text from property values, words from the URI of the entity, words from the URIs of objects (associated entities), words from predicates used to link to the entity, and words from the URI of associated classes via \( rdf:type \) relations. The categories were weighted with individual field boost factors, and the remaining parameters were assigned values guided by the authors’ judgement. The experiments aimed primarily to recap shortcomings of techniques that fail to implement correctly saturation effects and field weighting, demonstrating how BM25F can address the context correctly. The authors used the 2009 INEX Wikipedia collection for evaluation, in turn transposed to RDF by mapping to equivalent DBPedia entries. A series of precision metrics was employed to compare BM25 and BM25F with corresponding variants of the Lucene engine. The results exhibited quality improvements over all the test beds using the BM25 variants. Lucene appeared to perform considerably worse when structure was taken into account. This is indicative of the method’s shortcomings in dealing with document structure. The results are profoundly remarkable, seeing how Lucene is widely adapted indiscriminately for such purposes.

In a similar study, Blanco, Mika, and Vigna (2011) designed an experiment with BM25F over the Billion Triples Challenge 2009 data set, the data set used as part of the Semantic Search Challenge in 2010/2011. In this experiment, the authors capture data type information from the top-300 data type properties in the collection and assign different weights to different categories of predicates. Properties were manually classified into three classes (important, unimportant, neutral), with weights assigned to each class. Domain names were also classified between important and unimportant, with dbpedia.org and netflix.com constituting the important category. The experiments focus on a simplified version of BM25F, with individual field lengths projected onto a higher dimension as the size of the enclosing entity, effectively reducing the index space required to store individual field lengths for a potentially very large set of entities and property values. The method appears to be a revised version of the winning team’s submission at the 2010 Semantic Search Challenge (Blanco et al., 2010). Results from the experiments indicated 42% improvement in average precision over the best run at the 2010 competition.

**Link-Analysis Inspired Methods**

The hypertextual structure of the web has been one of the richest sources of information for developing reliable ranking heuristics. There are conceivably many applications that can benefit from analysis of hypertext links, including document classification and clustering, deciding what pages to crawl, prioritizing documents in vast posting lists, and composite scoring of web pages on any given query. The two most popular contributions in this area with important implications for web search have been the HITS algorithm by Kleinberg (1999) and the PageRank algorithm by Brin and Page (1998). The former is typically treated as a query-dependent algorithm, useful for such cases as finding communities of practice on a given topic or postquery processing and sorting of documents. PageRank is most commonly known for query-independent or prior scoring of documents, providing a static score element for web pages on which to base a notion of importance or popularity. Both algorithms are iterative algorithms whose values are expected to converge after a certain number of iterations.
Seminar Web data are in many ways similar to the hypertext web, in that links constitute a fundamental notion of relevancy. However, resources on the Semantic Web can be related via a multitude of heterogeneous links, each indicating a different type of association. For this, static scoring via conventional link analysis to derive scores of popularity or importance demands deeper elucidation of what is actually being conferred across web resources. The PageRank algorithm, primarily because of its popular pose in the literature and as part of the Google search engine, has served as a common baseline for link analysis on Semantic Web graphs.

**PageRank**

PageRank assumes a homogeneous structure of the web, in which links carry a uniform endorsement to the analysis of pages. PageRank has a simple, intuitive, probabilistic interpretation that tries to emulate the likelihood of a person randomly surfing the web to arrive at a particular page. The PageRank of a page is derived from its back-links and is proportional to the sum of the ranks of all the pages that link to it. If we assume $x$ to be a page on the web, $B_x$ to be the set of all pages that link to $x$, and $N_x$ the total outgoing links of $x$, PageRank is computed as follows:

$$R(x) = c \sum_{y \in B_x} \frac{R(y)}{N_y} + E(x)$$

(12)

where $c$ and $E(x)$ are treated as normalizing constants ranging between 0 and 1 and are used to balance the equation. $c$ indicates the maximum rank contribution of the set of pages $B_x$, and $E(x)$ adjusts the score to an upper limit of 1, while setting a uniform initial value across all pages. Given the algebraic relation of the two parameters, they are often expressed as $d$ and $(1 - d)$, respectively. Given this formulation, the importance that a page confers to $x$ is determined by the importance of the page itself and is inversely proportional to the number of pages to which it links.

Extensions to PageRank for weighted link analysis are a common scenario in the IR and database literature. A reflective example is Microsoft’s PopRank model (Nie, Zhang, Wen, & Ma, 2005), which adopts the algorithm to “popularity propagation factors” learned from partial ranking lists via a machine learning approach. The method emulates PageRank’s “random surfer” model to a “random object finder” and has been applied successfully on large document collections. ObjectRank (Balmin, Hristidis, & Papakonstantinou, 2004), another example, applies PageRank in a query-dependent fashion to satisfy keyword searches in databases. The technique assumes a weighted schema graph with links assigned different authority transfer rates. XRank (Guo et al., 2003) is a similar approach for XML classification. The following examples constitute reflective uses of PageRank for ranking Semantic Web data for search.

**Uses of PageRank for Semantic Web Search**

Some of the earliest retrieval techniques applied on the Semantic Web focused on finding relevant ontologies, or Semantic Web documents (SWD), as potential matches to a customary set of keywords. Effectively, a general methodology for ranking SWDs can work for ranking RDF instances or entities, but usually invested approaches are not always so generic. Swoogle\(^8\) (Ding et al., 2005) dominated this area of development, maintaining a robust index to ontologies across a wide range of domains. The main construct of Swoogle’s ranking is based on a modular weighted PageRank (OntoRank) that aims to assess the popularity of documents by exploring different interdocument relations. These take the form of axiomatic referral links, such as when an SWD uses or extends vocabulary terms defined in another (for example, via rdfs:subClassOf or rdf:type relations). The main extension to the original algorithm involves the inclusion of manually specified navigation preferences, which take the form of weights assigned to the semantic links between documents. Considering link($y$,l,$x$) to denote a relation, $l$, between $x$ and $y$, and weight($l$) to be a user-specified weight for the given relation, PageRank is adjusted as follows:

$$R'(x) = d \sum_{y \in B_x} R'(y) \times f(y, x) + (1 - d)$$

(13)

where $f(y,x)$ is the aggregated weight over all the relations from $y$ to $x$. OntoRank further accumulates a document’s final score with the ranks of all the documents that import the given ontology via owl:imports. Swoogle’s ranking is inclusive, and OntoRank is applied to provide ranking for ontology terms in a knowledge base, for example, to facilitate retrieval of properties and classes based on how often they are used and the popularity of the documents that use them. The main pivot of the approach is whether the underlying documents are well connected or cross-referenced, which is not necessarily the case. Evidently, autonomous documents may end up receiving poor OntoRank scores.

**SWRank.** SWRank (Wu & Li, 2007) is a prototype entity-ranking method that, like Swoogle’s OntoRank, exploits the use of multiple relations between resources to implement PageRank-like analysis. SWRank considers overall hub score to be the popularity of an entity, which is the reverse of conventional PageRank. The approach works by reversing the direction of all the edges in an RDF graph and applying weighted PageRank (as with Equation 13) on the reversed graph. The outcome is a shift of orientation, but with relative consistency to the original algorithm. Reverse PageRank is a speculative technique for hypertext browsing, investigated

---

\(^8\)Swoogle: http://swoogle.umbc.edu
previously by Fogaras (2003). SWRank works consistently across the schema and data levels of a knowledge base and hence involves no pragmatic differentiation between schema and assertional semantics.

The system outlined by Wu and Li (2007) combines SWRank with classic vector-based ranking for overall retrieval of entities. The vector-based scheme emulates traditional TF-IDF on all the literal values associated with resources. A resource is effectively treated as a bag of words, without further processing of data type relations. Experiments on data sets generated from SourceForge9 and SchemaWeb10 revealed comparable convergence speeds between SWRank and plain PageRank. SWRank coincided more with the project web hits statistics from SourceForge, a rather promising outcome. The main caveat that we observe with reversing the algorithm is that the orientation shifts from distilling authorities to focusing on hubs in the network. Traditional PageRank would classify a resource as popular if many other resources link to it and not many others, and many resources link to them and not many others, which is a reasonable assumption. With SWRank, it appears that resources are classified as popular if they link to very few resources that link to very few others; such implies a “close” community finder rather than a popularity estimate. The motivation for using Reverse PageRank requires deeper justification, especially when employed as a general algorithm for enhancing the ranks of resources.

Sindice. Sindice11 (Oren et al., 2008) is an end-to-end search engine for linked data on the web, offering a suite of API tools for querying the indexed sources (at the time limited to keyword, URI, and inverse functional property lookups). The engine underlying the keyword lookup processor (SIREn12) extends on the Apache Lucene project and supports full-text and semistructured queries. Sindice employs a two-layer hierarchical link analysis model to rank resources, known as DING (for data set ranking; Delbru, Toupikov, Catasta, Tummarello, & Decker, 2010), that distinguishes between entity and data set information (as illustrated in Figure 2). Links are aggregated from the entire graph and weighted as bundles of links and link sets via a linear TF-IDF-inspired unsupervised method. The weighting scheme assigns a higher degree of importance to links with a high frequency in individual data sets and lower frequency across the entire data set collection. The DING algorithm is an extension to PageRank (works exactly like a weighted PageRank when applicable) and defuses the weights into data set and entity ranks by traversing the weighted graphs. The aim is to estimate the importance of data sets across the entire collection and that of entities on a per-data-set level. The final score is a linear combination of the two weights after normalizing the ranks by the size of the data sets.

Sindice employs a variety of interesting methods to rank resources, but very little evaluation exists to demonstrate the quality of the approach, especially at different granularity levels of the algorithm. Experiments were conducted to evaluate individual parts of DING against a baseline method (operating on the full data graph). These revealed close correlation between the different methods, demonstrating that a global entity rank can possibly be interpolated via less expensive local computations. User studies also assessed the performance of the ranking on different data sets, using a similar methodology. Yet again, close correlation was found between the different components of DING and the baseline method. There is scepticism about these results. The user studies compared results produced by different blends of the same algorithm, indicating correlation to a baseline subjective estimate. For example, whether popular (w.r.t. PageRank) and larger data sets should receive higher ranks on the Web of Data is not precisely understood. Secluded knowledge bases may contain very important information. The Sindice engine was used in the 2010 Semantic Search Challenge (Delbru, Rakhmawati, & Tummarello, 2010) and scored fourth of the six contestants. In 2011, the competing system was extended with a BM25F variant (Campinas, Delbru, & Rakhmawati, 2011) and scored first on the entity search track and second on the list search track.

SWSE. SWSE13 (Hogan et al., 2011) is another prototype data aggregation project that indexes Semantic Web data for searching. SWSE crawls and bundles RDF data with non-RDF sources (HTML documents, RSS feeds) and arranges the content into canonical bundles after analyzing owl:sameAs and inverse functional property relations. TBox reasoning is also adopted to infer new statements about the data. Ranking in SWSE is based chiefly on the notion of a naming authority (Harth, Kinsella, & Decker, 2009), which aims to distinguish and establish a connection between an

FIG. 2. Sindice’s two-layer model. Data set layer made of interdata-set link sets, and entity layer made of inter- and intradata-set links. Simplified reproduction from Delbru et al. (2010).
entity identifier (URI) and the source with the authority to assign the identifier, also referred to as pay level domain, for example, example.com for foo.example.com. In the case of HTTP 303 redirections (a common scenario in publication of linked data) the naming authority is extracted from the redirected URI. Having constructed a naming authority graph, PageRank is applied to derive scores for each source-level identifier. Property and \texttt{rdfs:} object-position URIs are subject to overinflating the ranks and therefore are excluded from the derivation of the graphs. The rank of individual entity identifiers subsumes the ranks of the sources in which the identifier occurs. The intuition is that, the more highly ranked the source mentioning a URI, the higher the rank of the term should be. SWSE combines the PageRank scores of URIs with simple TF-IDF query-dependent scores for overall ranking. There is no evidence of data type property demarcation, although some indication is given that labels (literals linked to \texttt{rdfs:}label) are preferred over other primitive data values.

Evaluation of the naming authority strategy has focused mainly on evaluating different variants of the algorithm (differing according to the level of the naming authority) and contrasting the results with a baseline method. This is similar to the Sindice experiments. The baseline method included a naive version of PageRank operating directly on entities by not taking sources into account. Experiments were conducted on several data sets, including a stripped version of the 2008 Billion Triples Challenge data set. Quality evaluation was driven by a user study, in which 15 participants were asked to rate results from different top-10 ranked lists. The proposed algorithm exhibited improvements over the baseline method. Performance evaluations indicated runtime properties similar to those of the baseline method. As with the previous experiments, there is some scepticism about the results. The evaluation involved a small number of test users and queries and focused entirely on the top-10 results. These did not involve appropriate effectiveness metrics (precision/recall) or significance tests.

Conclusions and Future Research Directions

This article introduces the area of semantic search from a broad point of view and subsequently narrows its focus to key techniques from the literature involving ranked keyword search over Semantic Web data. We present important concepts and common techniques in some detail, which should appeal to readers interested in a deeper perspective on the various methods and systems implemented. Naturally, this comes at the expense of a more complete survey over work in this area. The following sections revisit highlights from the survey and outline key areas toward which future research may be directed. The material presented is but a small subset of a much broader theme, so many of the longitudinal challenges in IR are effectively carried over. Directions for future work are based on aspects that mostly reflect the course of the survey, but the topics are by no means exhaustive.

Unifying Ranking Models

Ranking is among the most functional issues in search engines. Several approaches have been described in this article, but none of them stands out as the definitive solution. The different orientations of the techniques presented aim to address different aspects of a retrieval process. For example, link analysis techniques, as in the popularity-based measures presented, give us insight on the linkage and density of the graphs surrounding entities. Propagation and graph exploration techniques are useful for distributing query evidence to the graphs. At the same time, probabilistic models, as in BM25F, have proved very successful in modeling and reasoning with different frames of contextual content in knowledge bases. The question remains whether an overall view can be synthesized from these and other vertical approaches. Research on combining multiple models of relevance, therefore, seems highly relevant. Frameworks that can blend together query-independent and query-dependent techniques to prioritize query evidence across clusters of high-proximity nodes or describe both probabilistic and logical processes (e.g., restrictions on types and predicates) to allow more complex constraint queries may be one way forward.

Several systems outlined in this survey also make effective use of precise axiomatic relations to enhance the solution space during or prior to query processing. For example, Tran et al. (2009) explore type and subsumption information to develop summary graphs for more efficient graph exploration, and Hogan et al. (2011) explore OWL semantics to expand the solution space with additional explicit semantics prior to query processing. These are interesting operations that make effective use of some of the unique characteristics of Semantic Web data. Class and identity correspondences are among the most common forms of mappings on the Web of Data. We expect that the exploitation of these and other emerging common constructs will remain key to demonstrating how consensus and improved ranking can be achieved across heterogeneous data.

Indexing Schemes

Retrieval efficiency is a major consideration when thinking about functional models for a wide range of data collections. Although search engines generally do not have the costs associated with relational and RDF databases, there are significant obstacles in terms of fast response, because query terms may appear in a very large number of documents/entities that are associated with many other terms. The efficiency of a ranking model is largely dependent on the choice of an appropriate scheme to store and retrieve the necessary information. In conventional IR, inverted indexes (Zobel & Moffat, 2006) have been the most common structure explored and implemented across a number of standard search engine libraries. In an inverted index, vocabulary terms are typically the basic index unit stored in a dictionary (either a hash map or a tree), with
pointers to associated lists of document identifiers and relevant frequency information. Inverted indexes are flexible data structures with common extensions in the literature, such as parametric zone indexes for models that distinguish between various parts of a document or positional indexes for models that prioritize phrases in text.

With respect to the semantic search methods reviewed in this article, several considerations make the process rather different from more conventional retrieval practices. First is the ability to retrieve resources based on words that appear in the values of properties (as in models that group property values into different weighted zones or expect restrictions on the names of properties) and the need to reason with object-level semantics for graph exploration and propagation techniques. Inverted indexes remain a natural course for modeling these types of associations. Most systems use or extend a standard search engine library (e.g., Lucene, MG4J, Lemur) to associate keyword-level indexes with entities, although, beyond a few exceptions, not much information is given on the precise implementation details.

Blanco et al. (2011) explore MG4J’s positional indexes to expand terms with field information corresponding to the top-300 data type properties from the Billion Triples Challenge data set. The authors focus on an efficient implementation by exploring an additional index to group properties into three broader, weighted classes, effectively leaving them with only three fields to parameterize each individual term. Tran et al. (2009) focus on an expressive keyword index for graph exploration. The authors use inverted indexes to associate terms with lists of connected nodes via specific predicates/edges and to the labels of edges and classes. To reduce time and space complexity, object-level semantics are captured between classes of entities as a summary graph index, so instance-level relations are aggregated at a higher dimension. In a similar context, the authors of BLINKS (He et al., 2007) present a block-based partitioning scheme that divides the graph into several subgraphs and captures keyword–node, node–keyword, and block-level proximity information into a set of inverted indexes for use in bidirectional graph exploration. Both of the aforementioned experiments were carried out on single data sets (the largest of approximately 26M triples) and illustrate affordable use for practical implementations.

Indexing schemes to support models over large and possibly multiple data set environments will remain a key factor in future implementations. That standard libraries can be developed or extended to provide basic means for graph partitioning, propagation and various levels of parametric indexes is a highly desirable prospect. Costs associated with index maintenance, combination of models (e.g., BM25F with graph exploration), and support for extended queries are interesting areas to explore as well.

**Tasks, Data Sets, and Evaluation**

For many years, research in IR has been driven by careful and thorough evaluation of the quality of proposed innovations. Conference series such as TREC and INEX have contributed to a community consensus on a portfolio of principled evaluation measures for assessing the performance of search algorithms. Methodical evaluation is key to making progress in the field. It is also essential to understanding whether a search engine is being used effectively and whether it provides the functionality for which it was conceived.

Starting an evaluation campaign for semantic search is, however, far from trivial. The community will have to agree on a precise perimeter of queries to assess and a set of data sets that are mostly reflective of the context of search. The Semantic Web community has recently organized an “ad-hoc object retrieval” task (Halpin et al., 2010; Pound et al., 2010), which is a step in the right direction, providing a general reference RDF collection for entity-oriented searches. The collection focuses on web queries (simple keyword queries) and a general, sizeable corpus representative of real-world data crawled from multiple sources on the web. An issue that may require further consideration is the precedence of selected domains in the collection (data and assessments), with dbpedia.org taking approximately 50% of the distribution. Some of the systems competing at the Semantic Search Challenge appear to have exploited the distribution for a better chance of winning (Halpin et al., 2010).

Results from the two consecutive runs of the Semantic Search Challenge are a valuable point of reference for comparison against a baseline of methods over the billion triples data set. Arguably, some of the competing systems suffered from a rather conservative perspective, but a few systems (some of which are reviewed in this survey) are interesting assimilations of popular techniques. An interesting next frontier for future competitions may be the proliferation of different tasks to direct focus on specific application needs and enduring trends, for example, a task focusing on semantic-oriented queries (e.g., queries involving variable matching and restrictions on attributes) as opposed to plain keywords or a task focusing on statistical and geographical data as found in the abundance of government-released data sets. Platforms that demonstrate good performance across a variety of domains will without doubt be key indicators of successful implementations, but a more gradual evolution from microexperiments to macrosettings may be a more appropriate path. The community can then look forward to unifying the most competent solutions, those most appropriate to deal with the unique characteristics of each task.

**Integration with User Interfaces**

This review focuses on a single mode of user interaction and presents in detail several forms of algorithmic approaches for distilling information from knowledge bases to satisfy user queries. From a broader perspective, however, semantic search is very commonly viewed as an iterative and exploratory process in which the user can actively engage with the system via various forms of interaction (Hildebrand, Ossenbruggen, & Hardman, 2007; Uren et al.,
Agrawal, S., Chaudhuri, S., & Das, G. (2002). DBXplorer: A system for
keyword-based search over relational databases. In R. Agrawal, K.
Dittrich, & A. Ngu (Eds.), Proceedings of the 18th International Confer-
ence on Data Engineering (ICDE '02) (pp. 5–16). Washington, DC: IEEE
Computer Society.

Anderson, J.R. (1983). A spreading activation theory of memory. Journal of
Verbal Learning and Verbal Behavior, 22(3), 261–295.

Baeza-Yates, R. (2004). Challenges in the interaction of information
retrieval and natural language processing. In A. Gelbukh (Ed.), Proceed-
ings of the 5th International Conference on Computational Linguistics
and Intelligent Text Processing (CICLing '04) (Vol. 2945, pp. 445–456).
Berlin: Springer-Verlag.

Baeza-Yates, R., Ciaramita, M., Mika, P., & Zaragoza, H. (2008). Towards
semantic search. In E. Kapetanios, V. Sugumaran, & M. Spiliopoulos
(Eds.), Proceedings of the 13th International Conference on Natural
Language and Information Systems: Applications of Natural Language
Information to Information Systems (NLDB '08) (Vol. 5039, pp. 4–11). Berlin:
Springer-Verlag.

Balmin, A., Hristidis, V., & Papakonstantinou, Y. (2004). Objecttrank:
authority-based keyword search in databases. In M.A. Nascimento, M.T.
Oszu, D. Kossmann, R.J. Miller, J.A. Blakeley, & B. Schieber (Eds.),
Proceedings of the 30th International Conference on Very Large Data
Bases (VLDB '04) (Vol. 30, pp. 564–575). San Francisco: Morgan
Kaufmann.

Balog, K., Serydyukov, P., & Vries, A.P. (2010). Overview of the TREC
2010 entity track. In E.M. Voorhees & L.P. Buckland (Eds.), Proceedings of the 19th
Text Retrieval Conference (TREC '10). Gaithersburg, Maryland: NIST.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web.
Scientific American, 284(5), 28–37.

Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., & Sudarshan, S.
(2002). Keyword searching and browsing in databases using BANKS.
In R. Agrawal, K. Dittrich, & A. Ngu (Eds.), Proceedings of the 18th
International Conference on Data Engineering (ICDE '02) (pp. 431–
440). Washington, DC: IEEE Computer Society.

Blanco, R., Mika, P., & Zaragoza, H. (2010). Entity search track submission
by yahoo! research barcelona. Paper presented at the 3rd International
Semantic Search Workshop (SemSearch '10). WWW2010, New York.

Balog, K., Mika, P., & Vigna, S. (2011). Effective and efficient entity
search in RDF data. In L. Aroyo, C. Welty, H. Alani, J. Taylor, & A.
Bernstein (Eds.), Proceedings of the 10th International Conference on
The Semantic Web (ISWC '11) (Vol. 7031, pp. 83–97). Berlin: Springer-
Verlag.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web
search engine. Computer Networks and ISDN Systems, 30(1–7), 107–117.

Campinas, S., Delbru, R., & Rakhmawati, N.A. (2011). Sindice BM25MF
at SemSearch 2011. Paper presented at the 4th International Semantic
Search Workshop (SemSearch '11). WWW2011, Hyderabad, India.

Celino, I., Valle, E.D., Cerizza, D., & Turati, A. (2006). Squggle: A
semantic search engine for indexing and retrieval of multimedia content.
In L.J.B. Nixon, L. Hardman, A. Scherp, & S. Boll (Eds.), Proceedings
of the 1st International Workshop on Semantic-Enhanced Multimedia
Presentation Systems (SEMS '06) (Vol. 228, pp. 20–34). Berlin: CEUR-
WS.org.

Chen, H., & Ng, T. (1995). An algorithmic approach to concept exploration
in a large knowledge network (automatic thesaurus consultation); sym-
biotic branch-and-bound search vs. connectionist Hopfield net activation.
Journal of the American Society for Information Science, 46, 348–369.

Chen, Y., Wang, W., Liu, Z., & Lin, X. (2009). Keyword search on struc-
tured and semi-structured data. In U. Cetintemel, B.S. Zdonik, D.
Kossmann, & N. Taitbul (Eds.), Proceedings of the 2009 ACM SIGMOD
International Conference on Management of Data (SIGMOD '09) (pp.
1005–1010). New York: ACM Press.

Cohen, S., Mamou, J., Kanza, Y., & Sagiv, Y. (2003). XSEarch: A semantic
search engine for XML. In J.C. Freytag, P.C. Lockemann, S. Abiteboul,
M.J. Carey, P.G. Selinger, & A. Heuer (Eds.), Proceedings of the 29th
International Conference on Very Large Data Bases (VLDB '03) (Vol.
29, pp. 45–56). San Francisco: Morgan Kaufmann.

Crestani, F. (1997). Application of spreading activation techniques in
information retrieval: Artificial Intelligence Review, 11(6), 453–482.

Croft, B., Metzler, D., & Strohman, T. (2009). Search engines: Information
retrieval in practice. Upper Saddle River, New Jersey: Pearson Addison-
Wesley.

Delbru, R., Rakhmawati, N.A., & Tummarello, G. (2010). Sindice at Sem-
Search 2010. Paper presented at the 3rd International Semantic Search
Workshop (SemSearch '10). WWW2010, New York.

Delbru, R., Toupikov, N., Catala, M., Tummarello, G., & Decker, S.
(2010). Hierarchical link analysis for ranking web data. In L. Aroyo,
G. Antoniou, E. Hyvönen, A. Teije, & H. Stuckenschmidt (Eds.), Proceed-
ings of the 7th International Conference on The Semantic Web: Research
and Applications (ESWC '10) (pp. 225–239). Berlin: Springer-Verlag.

Demartini, G., Iofciu, T., & de Vries, A. (2010). Overview of the INEX
2009 Entity Ranking Track. In S. Geva, J. Kamps, & A. Trotman (Eds.),
Focused Retrieval and Evaluation, 8th International Workshop of the
Initiative for the Evaluation of XML Retrieval (INEX '09) (Vol. 6203, pp.
254–264). Berlin: Springer-Verlag.

Ding, D., Yang, J., Li, Q., Wang, L., & Wenyin, L. (2004). Towards a flash
search engine based on expressive semantics. In S. Feldman, M. Uretsky,
M. Najork, & C. Wills (Eds.), Proceedings of the 13th International
Conference on Very Large Data Bases (VLDB '03) (Vol. 1005–1010).
Washington, DC: IEEE Computer Society.

Domingue, J., Fensel, D., & Hendler, J.A. (2011). Handbook of Semantic
Web Technologies. Berlin: Springer-Verlag.
Vallet-Weadon, D., Fernández-Sánchez, M., & Castells-Azpilicueta, P. (2005). The quest for information retrieval on the Semantic Web. UPGRADE: The European Journal for the Informatics Professional. Monograph: The Semantic Web, 2005(6), 19–23.

Wei, W., & Barnaghi, P.M. (2007). Semantic support for medical image search and retrieval. In J.W. Gardner (Ed.), Proceedings of the 5th IASTED International Conference: Biomedical Engineering (BioMED '07) (pp. 315–319). Innsbruck, Austria: ACTA Press.

Wu, G., & Li, J. (2007). SWRank: An approach for ranking Semantic Web reversely and consistently. In J. Liu & H. Zhuge (Eds.), Proceedings of the 3rd International Conference on Semantics, Knowledge and Grid (SKG '07) (pp. 116–121). Washington, DC: IEEE Computer Society.

Zhai, C. (2008). Statistical language models for information retrieval: A critical review. Foundations and Trends in Information Retrieval, 2(3), 137–213.

Zhou, Q., Wang, C., Xiong, M., Wang, H., & Yu, Y. (2007). SPARK: Adapting keyword query to semantic search. In K. Aberer, K.-S. Choi, N.F. Noy, D. Allemang, K.-I. Lee, L.J.B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, & P. Cudré-Mauroux (Eds.), Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC ’07, ASWC ’07) (Vol. 4825, pp. 694–707). Berlin: Springer-Verlag.

Zobel, J., & Moffat, A. (2006). Inverted files for text search engines. ACM Computing Surveys (CSUR), 38(2), 6.