Towards a Statistically Semantic Web

Gerhard Weikum, Jens Graupmann, Ralf Schenkel, and Martin Theobald

Max-Planck Institute of Computer Science
Saarbruecken, Germany

Abstract. The envisioned Semantic Web aims to provide richly annotated and explicitly structured Web pages in XML, RDF, or description logics, based upon underlying ontologies and thesauri. Ideally, this should enable a wealth of query processing and semantic reasoning capabilities using XQuery and logical inference engines. However, we believe that the diversity and uncertainty of terminologies and schema-like annotations will make precise querying on a Web scale extremely elusive if not hopeless, and the same argument holds for large-scale dynamic federations of Deep Web sources. Therefore, ontology-based reasoning and querying needs to be enhanced by statistical means, leading to relevance-ranked lists as query results.
This paper presents steps towards such a "statistically semantic" Web and outlines technical challenges. We discuss how statistically quantified ontological relations can be exploited in XML retrieval, how statistics can help in making Web-scale search efficient, and how statistical information extracted from users' query logs and click streams can be leveraged for better search result ranking. We believe these are decisive issues for improving the quality of next-generation search engines for intranets, digital libraries, and the Web, and they are crucial also for peer-to-peer collaborative Web search.

1 The Challenge of "Semantic" Information Search

The age of information explosion poses tremendous challenges regarding the intelligent organization of data and the effective search of relevant information in business and industry (e.g., market analyses, logistic chains), society (e.g., health care), and virtually all sciences that are more and more data-driven (e.g., gene expression data analyses and other areas of bioinformatics). The problems arise in intranets of large organizations, in federations of digital libraries and other information sources, and in the most humongous and amorphous of all data collections, the World Wide Web and its underlying numerous databases that reside behind portal pages. The Web bears the potential of being the world's largest encyclopedia and knowledge base, but we are very far from being able to exploit this potential.

Database-system and search-engine technologies provide support for organizing and querying information; but all too often they require excessive manual preprocessing, such as designing a schema and cleaning raw data or manually classifying documents into a taxonomy for a good Web portal, or manual postprocessing such as browsing through large result lists with too many irrelevant items or surfing in the vicinity of promising but not truly satisfactory approximate matches. The following are a few example queries where current Web and intranet search engines fall short or where data

P. Atzeni et al. (Eds.): ER 2004, LNCS 3288, pp. 3–17, 2004.

integration techniques and the use of SQL-like querying face unsurmountable difficulties even on structured, but federated and highly heterogeneous databases:

Q1: Which professors from Saarbruecken in Germany teach information retrieval and do research on XML?
Q2: Which gene expression data from Barrett tissue in the esophagus exhibit high levels of gene A01g? And are there any metabolic models for acid reflux that could be related to the gene expression data?
Q3: What are the most important research results on large deviation theory?
Q4: Which drama has a scene in which a woman makes a prophecy to a Scottish nobleman that he will become king?
Q5: Who was the French woman that I met in a program committee meeting where Paolo Atzeni was the PC chair?
Q6: Are there any published theorems that are equivalent to or subsume my latest mathematical conjecture?

Why are these queries difficult (too difficult for Google-style keyword search unless one invests a huge amount of time to manually explore large result lists with mostly irrelevant and some mediocre matches)? For Q1 no single Web site is a good match; rather one has to look at several pages together within some bounded context: the homepage of a professor with his address, a page with course information linked to by the homepage, and a research project page on semistructured data management that is a few hyperlinks away from the homepage. Q2 would be easy if asked for a single bioinformatics database with a familiar query interface, but searching the answer across the entire Web and Deep Web requires discovering all relevant data sources and unifying their query and result representations on the fly. Q3 is not a query in the traditional sense, but requires gathering a substantial number of key resources with valuable information on the given topic; it would be best served by looking up a well maintained Yahoo-style topic directory, but highly specific expert topics are not covered there. Q4 cannot be easily answered because a good match does not necessarily contain the keywords "woman", "prophecy", "nobleman", etc., but may rather say something like "Third witch: All hail, Macbeth, thou shalt be king hereafter!" and the same document may contain the text "All hail, Macbeth! hail to thee, thane of Glamis!". So this query requires some background knowledge to recognize that a witch is a woman, "shalt be" refers to a prophecy, and thane is a title for a Scottish nobleman. Q5 is similar to Q4 in the sense that it also requires background knowledge, but it is more difficult because it additionally requires putting together various information fragments: conferences on which I served on the PC found in my email archive, PC members of conferences found on Web pages, and detailed information found on researchers' homepages. And after having identified a candidate like Sophie Cluet from Paris, one needs to infer that Sophie is a typical female first name and that Paris most likely denotes the capital of France rather than the 500-inhabitants town of Paris, Texas, that became known through a movie. Q6 finally is what some researchers call "AI-complete", it will remain a challenge for a long time.

For a human expert who is familiar with the corresponding topics, none of these queries is really difficult. With unlimited time, the expert could easily identify relevant pages and combine semantically related information units into query answers. The challenge is to automate or simulate these intellectual capabilities and implement them so that they can handle billions of Web pages and petabytes of data in structured (but schematically highly diverse) Deep-Web databases.

2 The Need for Statistics

What if all Web pages and all Web-accessible data sources were in XML, RDF, or OWL (a description-logic representation) as envisioned in the Semantic Web research direction [25, 1]? Would this enable a search engine to effectively answer the challenging queries of the previous section? And would such an approach scale to billions of Web pages and be efficient enough for interactive use? Or could we even load and integrate all Web data into one gigantic database and use XQuery for searching it?

XML, RDF, and OWL offer ways of more explicitly structuring and richly annotating Web pages. When viewed as logic formulas or labeled graphs, we may think of the pages as having "semantics", at least in terms of model theory or graph isomorphisms[1]. In principle, this opens up a wealth of precise querying and logical inferencing opportunities. However, it is extremely unlikely that all pages will use the very same tag or predicate names when they refer to the same semantic properties and relationships. Making such an assumption would be equivalent to assuming a single global schema: this would be arbitrarly difficult to achieve in a large intranet, and it is completely hopeless for billions of Web pages given the Web's high dynamics, extreme diversity of terminology, and uncertainty of natural language (even if used only for naming tags and predicates). There may be standards (e.g., XML schemas) for certain areas (e.g., for invoices or invoice-processsing Web Services), but these will have limited scope and influence. A terminologically unified and logically consistent Semantic Web with billions of pages is hard to imagine.

So reasoning about diversely annotated pages is a necessity and a challenge. Similarly to the ample research on database schema integration and instance matching (see, e.g., [49] and the references given there), knowledge bases [50], lexicons, thesauri [24], or *ontologies* [58] are considered as the key asset to this end. Here an ontology is understood as a collection of *concepts* with various *semantic relationships* among them; the formal representation may vary from rigorous logics to natural language. The most important relationship types are hyponymy (specialization into narrower concepts) and hypernymy (generalization into broader concepts).

To the best of my knowledge, the most comprehensive, publicly available kind of ontology is the WordNet thesaurus hand-crafted by cognitive scientists at Princeton [24]. For the concept "woman" WordNet lists about 50 immediate hyponyms, which include concepts like "witch" and "lady" which could help to answer queries like Q4 from the previous section. However, regardless of whether one represents these hyponymy relationships in a graph-oriented form or as logical formulas, such a rigid "true-or-false" representation could never discriminate these relevant concepts from the other 48 irrelevant and largely exotic hyponyms of "woman". In information-retrieval (IR) jargon, such an approach would be called Boolean retrieval or Boolean reasoning; and IR almost always favors *ranked retrieval* with some quantitative relevance assessment. In fact, by simply looking at statistical correlations of using words like "woman" and "lady" together in some text neighborhood within large corpora (e.g., the Web or large digital libraries) one can infer that these two concepts are strongly related, as opposed to concepts like "woman" and "siren". Similarly, mere statistics strongly suggests that

[1] Some people may argue that all computer models are mere syntax anyway, but this is in the eye of the beholder.

a city name "Paris" denotes the French capital and not Paris, Texas. Once making a distinction of strong vs. weak relationships and realizing that this is a full spectrum, it becomes evident that the significance of semantic relationships needs to be quantified in some manner, and the by far best known way of doing this (in terms of rigorous foundation and rich body of results) is by using probability theory and statistics.

This concludes my argument for the necessity of a "statistically semantic" Web. The following sections substantiate and illustrate this point by sketching various technical issues where statistical reasoning is key. Most of the discussion addresses how to handle non-schematic XML data; this is certainly still a good distance from the Semantic Web vision, but it is a decent and practically most relevant first step.

3 Towards More "Semantics" in Searching XML and Web Data

Non-schematic XML data that comes from many different sources and inevitably exhibits heterogeneous structures and annotations (i.e., XML tags) cannot be adequately searched using database query languages like XPath or XQuery. Often, queries either return too many or too few results. Rather the ranked-retrieval paradigm is called for, with relaxable search conditions, various forms of similarity predicates on tags and contents, and quantitative relevance scoring. Note that the need for ranking goes beyond adding Boolean text-search predicates to XQuery. In fact, similarity scoring and ranking are orthogonal to data types and would be desirable and beneficial also on structured attributes such as time (e.g., approximately in the year 1790), geographic coordinates (e.g., near Paris), and other numerical and categorical data types (e.g., numerical sensor readings and music style categories).

Research on applying IR techniques to XML data has started five years ago with the work [26, 55, 56, 60] and has meanwhile gained considerable attention. This research avenue includes approaches based on combining ranked text search with XPath-style conditions [4, 13, 35, 11, 31, 38], structural similarities such as tree-editing distances [5, 54, 69, 14], ontology-enhanced content similarities [60, 61, 52], and applying probabilistic IR and statistical language models to XML [28, 2].

Our own approach, the XXL[2] query language and search engine [60, 61, 52], combines a subset of XPath with a similarity operator $\sim$ that can be applied to element or attribute names, on one hand, and element or attribute contents, on the other hand. For example, the queries Q1 and Q4 of Section 1 could be expressed in XXL as follows (and executed on a heterogeneous collection of XML documents):

```
Q1: Select * From Index
    Where ~professor As P
    And P = "Saarbruecken"
    And P//~course = "~IR"
    And P//~research = "~XML"
```

```
Q4: Select * From Index
    Where ~drama//scene As S
    And S//~speaker = "~woman"
    And S//~speech = "king"
    And S//~person = "~nobleman"
```

Here XML data is interpreted as a directed graph, including href or XLink/XPointer links within and across documents that go beyond a merely tree-oriented approach. End nodes of connections that match a path condition such as `drama//scene` are bound to node variables that can be referred to in other search conditions. Content conditions

[2] Flexible XML Search Language.

such as = "∼woman" are interpreted as keyword queries on XML elements, using IR-style measures (based on statistics like term frequencies and inverse element frequencies) for scoring the relevance of an element. In addition and most importantly, we allow expanding the query by adding "semantically" related terms taken from an ontology. In the example, "woman" could be expanded into "woman wife lady girl witch ...". The score of a relaxed match, say for an element containing "witch", is the product of the traditional score for the query "witch" and the *ontological similarity* of the query term and the related term, $sim(woman, witch)$ in the particular example. Element (or attribute) name conditions such as ∼course are analogously relaxed, so that, for example, tag names "teaching", "class", or "seminar" would be considered as approximate matches. Here the score is simply the ontological similarity, for tag names are only single words or short composite words. The result of an entire query is a ranked list of subgraphs of the XML data graph, where each result approximately matches all query conditions with the same binding of all variables (but different results have different bindings). The total score of a result is computed from the scores of the elementary conditions using a simple probabilistic model with independence assumptions, and the result ranking is in descending order of total scores.

Query languages of this kind work nicely on heterogeneous and non-schematic XML data collections, but the Web and also large fractions of intranets are still mostly in HTML, PDF, and other less structured formats. Recently we have started to apply XXL-style queries also to such data by automatically converting Web data into XML format. The COMPASS[3] search engine that we have been building supports XML ranked retrieval on the full suite of Web and intranet data including combined data collections that include both XML documents and Web pages [32]. For example, query Q1 can be executed on an index that is built over all of DBLP (cast into XML) and the crawled homepages of all authors and other Web pages reachable through hyperlinks. Figure 1 depicts the visual formulation of query Q1. Like in the original XXL engine, conditions with the similarity operator ∼ are relaxed using statistically quantified relationships from the ontology.

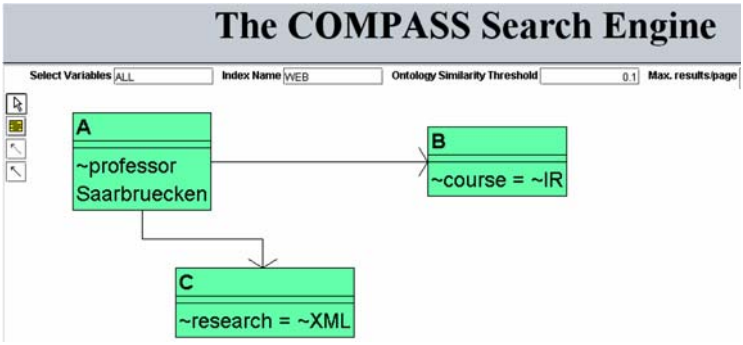

Fig. 1. Visual COMPASS Query

[3] Concept-oriented Multi-format Portal-aware Search System.

The conversion of HTML and other formats into XML is based on relatively simple heuristic rules, for example, casting HTML headings into XML element names. For additional automatic annotation we use the information extraction component ANNIE that is part of the GATE System developed at the University of Sheffield [20]. GATE offers various modules for analyzing, extracting, and annotating text; its capabilities range from part-of-speech tagging (e.g., for noun phrases, temporal adverbial phrases, etc.) and lexicon lookups (e.g., for geographic names) to finite state transducers for annotations based on regular expressions (e.g., for dates or currency amounts). One particularly useful and fairly light-weight component is the Gazetteer Module for named entity recognition based on part-of-speech tagging and a large dictionary containing names of cities, countries, person names (e.g., common first names), etc. This way one can automatically generate tags like `<location>` and `<person>`. For example, we were able to annotate the popular Wikipedia open encyclopdia corpus this way, generating about 2 million person and location tags. And this is the key for more advanced "semantics-aware" search on the current Web. For example, searching for Web pages about the physicist Max Planck would be phrased as `person = "Max Planck"`, and this would eliminate many spurious matches that a Google-style keyword query "Max Planck" would yield about Max Planck Institutes and the Max Planck Society[4].

There is a rich body of research on information extraction from Web pages and wrapper generation. This ranges from purely logic-based or pattern-matching-driven approaches (e.g., [51, 17, 6, 30]) to techniques that employ statistical learning (e.g., Hidden Markov Models) (e.g., [15, 16, 39, 57, 40]) to infer structure and annotations when there is too much diversity and uncertainty in the underlying data. As long as all pages to be wrapped come from the same data source (with some hidden schema), the logic-based approaches work very well. However, when one tries to wrap all homepages of DBLP authors or the course programs of all computer science departments in the world, uncertainty is inevitable and statistics-driven techniques are the only viable ones (unless one is willing to invest a lot of manual work for traditional schema integration, writing customized wrappers and mappers).

Despite advertising our own work and mentioning our competitors, the current research projects on combining IR techniques and statistical learning with XML querying is still in an early stage and there are certainly many open issues and opportunities for further research. These include better theoretical foundations for scoring models on semistructured data, relevance feedback and interactive information search, and, of course, all kinds of efficiency and scalability aspects. Applying XML search techniques to Web data is in its infancy; studying what can be done with named-entity recognition and other automatic annotation techniques and understanding the interplay of queries with such statistics-based techniques for better information organization are widely open fields.

4 Statistically Quantified Ontologies

The important role of ontologies in making information search more "semantics-aware" has already been emphasized. In contrast to most ongoing efforts for Semantic-Web on-

[4] Germany's premier scientific society, which encompasses 80 institutes in all fields of science.

tologies, our work has focused on quantifying the strengths of semantic relationships based on corpus statistics [52, 59] (see also the related work [10, 44, 22, 36] and further references given there). In contrast to early IR work on using thesauri for query expansion (e.g., [64]), the ontology itself plays a much more prominent role in our approach with carefully quantified statistical similarities among concepts.

Consider a graph of concepts, each characterized by a set of synonyms and, optionally, a short textual description, connected by "typed" edges that represent different kinds of relationships: hypernyms and hyponyms (generalization and specialization, aka. is-a relations), holonyms and meronyms (part-of relations), is-instance-of relations (e.g., Cinderella being an instance of a fairytale or IBM Thinkpad being a notebook), to name the most important ones.

The first step in building an ontology is to create the nodes and edges. To this end, existing thesauri, lexicons, and other sources like geographic gazetteers (for names of countries, cities, rivers, etc. and their relationships) can be used. In our work we made use of the WordNet thesaurus [24] and the Alexandria Digital Library Gazetteer [3], and also started extracting concepts from page titles and href anchor texts in the Wikipedia encyclopedia. One of the shortcomings of WordNet is its lack of instances knowledge, for example, brand names and models of cars, cameras, computers, etc. To further enhance the ontology, we crawled Web pages with HTML tables and forms, trying to extract relationships between table-header column and form-field names and the values in table cells and the pulldown menus of form fields. Such approaches are described in the literature (see, e.g., [21, 63, 68]). Our experimental findings confirmed the potential value of these techniques, but also taught us that careful statistical thresholding is needed to eliminate noise and incorrect inferencing, once again a strong argument for the use of statistics.

Once the concepts and relationships of a graph-based ontology are constructed, the next step is to quantify the strengths of semantic relationships based on corpus statistics. To this end we have performed focused Web crawls and use their results to estimate statistical correlations between the characteristic words of related concepts. One of the measures for the similarity of concepts $c1$ and $c2$ that we used is the Dice coefficient

$$Dice(c1, c2) = \frac{2|\{docs\ with\ c1\} \cap \{docs\ with\ c2\}|}{|\{docs\ with\ c1\}| + |\{docs\ with\ c2\}|}$$

In this computation we represent concept c by the terms taken from its set of synonyms and its short textual description (i.e., the WordNet gloss). Optionally, we can add terms from neighbors or siblings in the ontological graph. A document in the corpus is considered to contain concept c if it contains at least one word of the term set for c, and considered to contain both $c1$ and $c2$ if it contains at least one word from each of the two term sets. This is a heuristics; other approaches are conceivable which we are investigating.

Following this methodology, we constructed an ontolgy service [59] that is accessible via Java RMI or as a SOAP-based Web Service described in WSDL. The service is used in the COMPASS search engine [32], but also in other projects. Figure 2 shows a screenshot from our ontology visualization tool.

One of the difficulties in quantifying ontological relationships is that we aim to measure correlations between *concepts* but merely have statistical information about

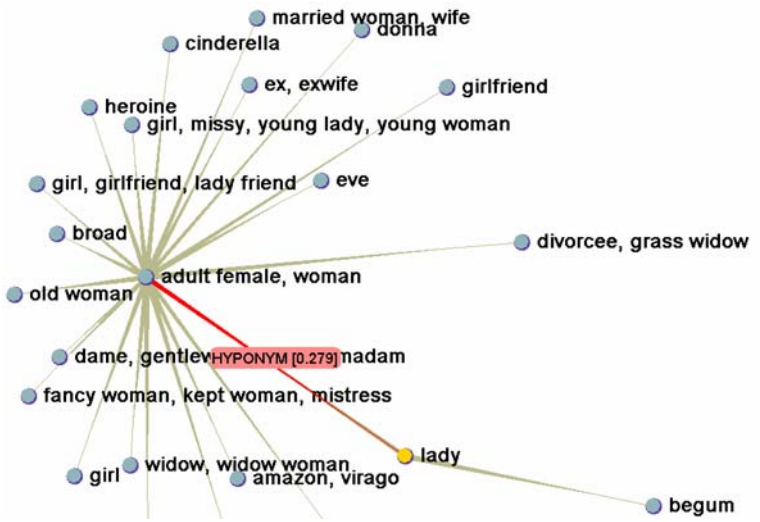

Fig. 2. Ontology Visualization

correlations between *words*. Ideally, we should first map the words in the corpus onto the corresponding concepts, i.e., their correct meanings. This is known as the *word sense disambiguation* problem in natural language processing [45], obviously a very difficult task because of polysemy. If this were solved it would not only help in deriving more accurate statistical measures for "semantic" similarities among concepts but could also potentially boost the quality of search results and automatic classification of documents into topic directories. Our work [59] presents a simple but scalable approach to automatically mapping text terms onto ontological concepts, in the context of XML document classification. Again, statistical reasoning, in combination with some degree of natural language parsing, is key to tackling this difficult problem.

Ontology construction is a highly relevant research issue. Compared to the ample work on knowledge representations for ontological information, the aspects of how to "populate" an ontology and how to enhance it with quantitative similarity measures have been underrated and deserve more intensive research.

5 Efficient Top-k Query Processing with Probabilistic Pruning

For ranked retrieval of semistructured, "semantically" annotated data, we face the problem of reconciling efficiency with result quality. Usually, we are not interested in a complete result but only in the top-k results with the highest relevance scores. The state-of-the-art algorithm for top-k queries on multiple index lists, each sorted in descending order of relevance scores, is the Threshold Algorithm, TA for short [23, 33, 47]. It is applicable to both relational data such as product catalogs and text documents such as Web data. In the latter case, the fact that TA performs random accesses on very long, disk-resident index lists (e.g., all URLs or document ids for a frequently occurring word), with only short prefixes of the lists in memory, makes TA much less attractive, however.

In such a situtation, the TA variant with sorted access only, coined NRA (no random accesses), stream-combine, or TA-sorted in the literature, is the method of choice [23, 34]. TA-sorted works by maintaining lower bounds and upper bounds for the scores of the top-k candidates that are kept in a priority queue in memory while scanning the index lists. The algorithm can safely stop when the lower bound for the score of the rank-k result is at least as high as the highest upper bound for the scores of the candidates that are not among the current top-k. Unfortunately, albeit theoretically instance-optimal for computing a precise top-k result [23], TA-sorted tends to degrade in performance when operating on a large number of index lists. This is exactly the case when we relax query conditions such as `~speaker = ~woman` using semantically related concepts from the ontology[5]. Even if the relaxation uses a threshold for the similarity of related concepts, we may often arrive at query conditions with 20 to 50 search terms.

Statistics about the score distributions in the various index lists and some probabilistic reasoning help to overcome this efficiency problem and re-gain performance. In TA-sorted a top-k candidate d that has already been seen in the index lists in $E(d) \subseteq [1..m]$, achieving score $s_j(d)$ in list j ($0 < s_j(d) \leq 1$), and has unknown scores in the index lists $[1..m] - E(d)$, satisfies:

$$lowerb(d) = \sum_{j \in E(d)} s_j(d) \leq s(d) \leq \sum_{j \in E(d)} s_j(d) + \sum_{j \notin E(d)} high_j = upperb(d)$$

where $s(d)$ denotes the total, but not yet known, score that d achieves by summing up the scores from all index lists in which d occurs, $lowerb(d)$ and $upperb(d)$ are the lower and upper bounds of d's score, and $high_j$ is the score that was last seen in the scan of index list j, upper-bounding the score that any candidate may obtain in list j. A candidate d remains a candidate as long as $upperb(d) > lowerb(rank\text{-}k)$ where $rank\text{-}k$ is the candidate that currently has rank k with regard to the candidates' lower bounds (i.e., the worst one among the current top-k). Assuming that d can achieve a score $high_j$ in all lists in which it has not yet been encountered is conservative and, almost always, overly conservative. Rather we could treat these unknown scores as random variables S_j ($j \notin E(d)$), and estimate the probability that d's total score can exceed $lowerb(rank\text{-}k)$. Then d is discarded from the candidate list if

$$P[lowerb(d) + \sum_{j \notin E(d)} S_j > lowerb(rank\text{-}k)] < \delta$$

with some pruning threshold δ.

This probabilistic interpretation makes some small, but precisely quantifiable, potential error in that it could dismiss some candidates too early. Thus, the top-k result computed this way is only approximate. However, the loss in precision and recall, relative to the exact top-k result using the same index lists, is stochastically bounded and can be set according to the application's needs. A value of $\delta = 0.1$ seems to be acceptable in most situations. Technically, the approach requires computing the convolution

[5] Note that the TA and TA-sorted algorithms can be easily modified to handle both element-name and element-contents conditions (as opposed to mere keyword sets in standard IR and Web search engines).

of the random variables S_j, based on assumed distributions (with parameter fitting) or precomputed histograms for the individual index lists and taking into account the current $high_j$ values, and predicting the (1-δ)-quantile of the sum's distribution. Details of the underlying mathematics and the implementation techniques for this *Prob-sorted* method can be found in [62]. Experiments with the TREC-12 .Gov corpus and the IMDB data collection have shown that such a probabilistic top-k method gains about a factor of ten (and sometimes more) in run-time compared to TA-sorted.

The outlined algorithm for approximate top-k queries with probabilistic guarantees is a versatile building block for XML ranked retrieval. In combination with ontology-based query relaxation, for example, expanding `~woman` into `(woman or wife or witch)`, it can add index lists dynamically and incrementally, rather than having to expand the query upfront based on thresholds. To this end, the algorithm considers the ontological similarity $sim(i, j)$ between concept i from the original query and concept j in the relaxed query, and multiplies it with the $high_j$ value of index list j to obtain an upper bound for the score (and characterize the score distribution) that a candidate can obtain from the relaxation j. This information is dynamically combined with the probabilistic prediction of the other unknown scores and their sum.

The algorithm can also be combined with distance-aware path indexes for XML data (e.g., the HOPI index structure [53]). This is required when queries contain element-name and element-contents conditions as well as path conditions of the form `professor//course` where matches for "course" that are close to matches for "professor" should be ranked higher than matches that are far apart. Thus, the Prob-sorted algorithm covers a large fraction of an XML ranked retrieval engine.

6 Exploiting Collective Human Input

The statistical information considered so far refers to data (e.g., scores in index lists) or metadata (e.g., ontological similarities). Yet another kind of statistics is information about user behavior. This could include relatively static properties like bookmarks or embedded hyperlinks pointing to high-quality Web pages, but also dynamic properties inferred from query logs and click streams. For example, Google's PageRank views a Web page as more important if it has many incoming links and the sources of these links are themselves high authorities [9, 12]. Technically, this amounts to computing stationary probabilities for a Markov-chain model that mimics a "random surfer". What PageRank essentially does is to exploit the intellectual endorsements that many human users (or Web administrators on behalf of organizations) provide by means of hyperlinks.

This rationale can be carried over to analyzing and exploiting entire surf trails and query logs of individual users or an entire user community. These trails, which can be gathered from browser histories, local proxies, or Web servers, capture implicit user judgements. For example, suppose a user clicks on a specific subset of the top 10 results returned by a search engine for a query with several keywords, based on having seen the summaries of these pages. This implicit form of relevance feedback establishes a strong correlation between the query and the clicked-on pages. Further suppose that the user refines a query by adding or replacing keywords, e.g., to eliminate ambiguities in the previous query. Again, this establishes correlations between the new keywords and

the subsequently clicked-on pages, but also, albeit possibly to a lesser extent, between the original query and the eventually relevant pages.

We believe that observing and exploiting such user behavior is a key element in adding more "semantic" or "cognitive" quality to a search engine. The literature contains some very interesting work in this direction (e.g., [19, 65, 67]), but is rather preliminary at this point. Perhaps, the difficulties in obtaining comprehensive query logs and surf trails outside of big service providers is a limiting factor in this line of experimental research. Our own, very recent, work generalizes the notion of a "random surfer" into a "random expert user" by enhancing the underlying Markov chain to incorporate also query nodes and transitions from queries to query refinements as well as clicked-on documents. Transition probabilities are derived from the statistical analysis of query logs and click streams. The resulting Markov chain converges to stationary authority scores that reflect not only the link structure but also the implicit feedback and collective human input of a search engine's users [43].

The de-facto monopoly that large Internet service providers have on being able to observe user behavior and statistically leverage this valuable information may be overcome by building next-generation Web search engines in a truly decentralized and ideally self-organized manner. Consider a peer-to-peer (P2P) system where each peer has a full-fledged Web search engine, including a crawler and an index manager. The crawler may be thematically focused or crawl results may be postprocessed so that the local index contents reflects the corresponding user's interest profile. With such a highly specialized and personalized "power search engine" most queries should be executed locally, but once in a while the user may not be satisfied with the local results and would then want to contact other peers. A "good" peer to which the user's query should be forwarded would have thematically relevant index contents, which could be measured by statistical notions of similarity between peers. These measures may be dependent on the current query or may be query-independent; in the latter case, statistics is used to effectively construct a "semantic overlay network" with neighboring peers sharing thematic interests [8, 42, 48, 18, 7, 66]. Both query routing and "statistically semantic" networks could greatly benefit from collective human inputs in addition to standard IR measures like term and document frequencies or term-wise score distributions: knowing the bookmarks and query logs of thousands of users would be a great resource to build on.

Further exploring these considerations on P2P Web search should become a major research avenue in computer science. Note that our interpretation of Web search includes ranked retrieval and thus is fundamentally more difficult than Gnutella-style file sharing or simple key lookups via distributed hash tables. Further note that, although query routing in P2P Web search resembles earlier work on metasearch engines and distributed IR (see, e.g., [46] and the references given there), it is much more challenging because of the large scale and the high dynamics of the envisioned P2P system with thousands or millions of computers and users.

7 Conclusion

With the ongoing information explosion in all areas of business, science, and society, it will be more and more difficult for humans to keep information organized and

extract valuable knowledge in a timely manner. The intellectual time for schema design, schema integration, data cleaning, data quality assurance, manual classification, directory and search result browsing, clever formulation of sophisticated queries, etc. is already the major bottleneck today, and the situation is likely to become worse. In my opinion, this will render all attempts to master Web-scale information in a perfectly consistent, purely logic-based manner more or less futile. Rather, the ability to cope with uncertainty, diversity, and high dynamics will be mandatory. To this end, statistics and their use in probabilistic inferences will be key assets.

One may envision a rich probabilistic algebra that encompasses relational or even object-relational and XML query languages, but interprets all data and results in a probabilistic manner and always produces ranked result result lists rather than Boolean result sets (or bags). There are certainly some elegant and interesting, but mostly theoretical, approaches along these lines (e.g., [27, 29, 37]). However, there is still a long way to go towards practically viable solutions. Among the key challenges that need to be tackled are customizability, composability, and optimizability.

- *Customizability:* The appropriate notions of ontological relationships, "semantic" similarities, and scoring functions are dependent on the application. Thus, the envisioned framework needs to be highly flexible and adaptable to incorporate application-specific or personalized similarity and scoring models.
- *Composability:* Algebraic building blocks like a top-k operator need to be composable so as to allow the construction of rich queries. The desired property that operators produce ranked list with some underlying probability (or "score mass") distribution poses a major challenge, for we need to be able to infer these probability distributions for the results of complex operator trees. This problem is related to the difficult issues of selectivity estimation and approximate query processing in a relational database, but goes beyond the state of the art as it needs to incorporate text term distributions and has to yield full distributions at all levels of operator trees.
- *Optimizability:* Regardless of how elegant a probabilistic query algebra may be, it would not be acceptable unless one can ensure efficient query processing. Performance optimization requires a deep understanding of rewriting complex operator trees into equivalent execution plans that have significantly lower cost (e.g., pushing selections below joins or choosing efficient join orders). At the same time, the top-k querying paradigm that avoids computing full result sets before applying some ranking is a must for efficiency, too. This combination of desiderata leads to a great research challenge in query optimization for a ranked retrieval algebra.

References

1. Karl Aberer et al.: Emergent Semantics Principles and Issues, International Conference on Database Systems for Advanced Applications (DASFAA) 2004
2. Mohammad Abolhassani, Norbert Fuhr: Applying the Divergence from Randomness Approach for Content-Only Search in XML Documents, ECIR 2004
3. Alexandria Digital Library Project, Gazetteer Development, http://www.alexandria.ucsb.edu/gazetteer/

4. Shurug Al-Khalifa, Cong Yu, H. V. Jagadish: Querying Structured Text in an XML Database, SIGMOD 2003
5. Sihem Amer-Yahia, Laks V. S. Lakshmanan, Shashank Pandit: FleXPath: Flexible Structure and Full-Text Querying for XML, SIGMOD 2004
6. Arvind Arasu, Hector Garcia-Molina: Extracting Structured Data from Web Pages, SIGMOD 2003
7. Mayank Bawa, Gurmeet Singh Manku, Prabhakar Raghavan: SETS: Search Enhanced by Topic Segmentation, SIGIR 2003
8. Matthias Bender, Sebastian Michel, Gerhard Weikum, Christian Zimmer: Bookmark-driven Query Routing in Peer-to-Peer Web Search, SIGIR Workshop on Peer-to-Peer Information Retrieval 2004
9. Sergey Brin, Lawrence Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine, WWW Conference 1998
10. Alexander Budanitsky, Graeme Hirst: Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures, Workshop on WordNet and Other Lexical Resources 2001
11. David Carmel, Yoëlle S. Maarek, Matan Mandelbrod, Yosi Mass, Aya Soffer: Searching XML Documents via XML Fragments, SIGIR 2003
12. Soumen Chakrabarti: Mining the Web: Discovering Knowledge from Hypertext Data, Morgan Kaufmann Publishers, 2002
13. T. Chinenyanga, N. Kushmerick: An Expressive and Efficient Language for XML Information Retrieval, Journal of the American Society for Information Science and Technology (JASIST) 53(6), 2002
14. Sara Cohen, Jonathan Mamou, Yaron Kanza, Yehoshua Sagiv: XSEarch: A Semantic Search Engine for XML, VLDB 2003
15. William W. Cohen, Matthew Hurst, Lee S. Jensen: A Flexible Learning System for Wrapping Tables and Lists in HTML Documents, in: A. Antonacopoulos, J. Hu (Editors), Web Document Analysis: Challenges and Opportunities, Word Scientific Publishing, 2004
16. William W. Cohen, Sunita Sarawagi: Exploiting Dictionaries in Named Entity Extraction: Combining Semi-markov Extraction Processes and Data Integration Methods, KDD 2004
17. Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo: RoadRunner: Towards Automatic Data Extraction from Large Web Sites, VLDB 2001
18. Arturo Crespo, Hector Garcia-Molina: Semantic Overlay Networks, Technical Report, Stanford University, 2003.
19. Hang Cui, Ji-Rong Wen, Jian-Yun Nie, Wei-Ying Ma: Query Expansion by Mining User Logs, IEEE Transactions on Knowledge and Data Engineering 15(4), 2003
20. Hamish Cunningham. GATE, a General Architecture for Text Engineering, Computers and the Humanities 36, 2002
21. Hasan Davulcu, Srinivas Vadrevu, Saravanakumar Nagarajan, I. V. Ramakrishnan: OntoMiner: Bootstrapping and Populating Ontologies from Domain-Specific Web Sites, IEEE Intelligent Systems 18(5), 2003
22. Anhai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, Alon Y. Halevy: Learning to Match Ontologies on the Semantic Web, VLDB Journal 12(4), 2003
23. Ronald Fagin, Amnon Lotem, Moni Naor: Optimal Aggregation Algorithms for Middleware, Journal of Computer and System Sciences 66(4), 2003
24. Christiane Fellbaum (Editor): WordNet: An Electronic Lexical Database, MIT Press, 1998
25. Dieter Fensel, Wolfgang Wahlster, Henry Lieberman, James A. Hendler (Editors): Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, MIT Press, 2002
26. Norbert Fuhr, Kai Großjohann: XIRQL – An Extension of XQL for Information Retrieval, SIGIR Workshop on XML and Information Retrieval 2000

27. Norbert Fuhr: Probabilistic Datalog: Implementing Logical Information Retrieval for Advanced Applications, Journal of the American Society for Information Science (JASIS) 51(2), 2000
28. Norbert Fuhr, Kai Großjohann: XIRQL: A Query Language for Information Retrieval in XML Documents, SIGIR 2001
29. Lise Getoor, Nir Friedman, Daphne Koller, Avi Pfeffer: Learning Probabilistic Relational Models, in: S. Dzeroski, N. Lavrac (Editors), Relational Data Mining, Springer, 2001
30. Georg Gottlob, Christoph Koch, Robert Baumgartner, Marcus Herzog, Sergio Flesca: The Lixto Data Extraction Project – Back and Forth between Theory and Practice, PODS 2004
31. Torsten Grabs, Hans-Jörg Schek: Flexible Information Retrieval on XML Documents. in: H. Blanken et al. (Editors), Intelligent Search on XML Data, Springer, 2003
32. Jens Graupmann, Michael Biwer, Christian Zimmer, Patrick Zimmer, Matthias Bender, Martin Theobald, Gerhard Weikum: COMPASS: A Concept-based Web Search Engine for HTML, XML, and Deep Web Data, Demo Program, VLDB 2004
33. Ulrich Güntzer, Wolf-Tilo Balke, Werner Kießling: Optimizing Multi-Feature Queries for Image Databases, VLDB 2000
34. Ulrich Güntzer, Wolf-Tilo Balke, Werner Kießling: Towards Efficient Multi-Feature Queries in Heterogeneous Environments, International Symposium on Information Technology (ITCC) 2001
35. Lin Guo, Feng Shao, Chavdar Botev, Jayavel Shanmugasundaram: XRANK: Ranked Keyword Search over XML Documents, SIGMOD 2003
36. Maria Halkidi, Benjamin Nguyen, Iraklis Varlamis, Michalis Vazirgiannis: THESUS: Organizing Web Document Collections Based on Link Semantics, VLDB Journal 12(4), 2003
37. Joseph Y. Halpern: Reasoning about Uncertainty, MIT Press, 2003
38. Raghav Kaushik, Rajasekar Krishnamurthy, Jeffrey F. Naughton, Raghu Ramakrishnan: On the Integration of Structure Indexes and Inverted Lists, SIGMOD 2004
39. Nicholas Kushmerick, Bernd Thomas: Adaptive Information Extraction: Core Technologies for Information Agents. in: M. Klusch et al. (Editors), Intelligent Information Agents, Springer, 2003
40. Kristina Lerman, Lise Getoor, Steven Minton, Craig A. Knoblock: Using the Structure of Web Sites for Automatic Segmentation of Tables, SIGMOD 2004
41. Zhenyu Liu, Chang Luo, Junghoo Cho, Wesley W. Chu: A Probabilistic Approach to Metasearching with Adaptive Probing, ICDE 2004
42. Jie Lu, James P. Callan: Content-based Retrieval in Hybrid Peer-to-peer Networks, CIKM 2003
43. Julia Luxenburger, Gerhard Weikum: Query-log Based Authority Analysis for Web Information Search, submitted for publication
44. Alexander Maedche, Steffen Staab: Learning Ontologies for the Semantic Web, International Workshop on the Semantic Web (SemWeb) 2001
45. Christopher D. Manning, Hinrich Schütze: Foundations of Statistical Natural Language Processing, MIT Press, 1999
46. Weiyi Meng, Clement T. Yu, King-Lup Liu: Building Efficient and Effective Metasearch Engines, ACM Computing Surveys 34(1), 2002
47. Surya Nepal, M. V. Ramakrishna: Query Processing Issues in Image (Multimedia) Databases, ICDE 1999
48. Henrik Nottelmann, Norbert Fuhr: Combining CORI and the Decision-Theoretic Approach for Advanced Resource Selection, ECIR 2004
49. Erhard Rahm, Philip A. Bernstein: A Survey of Approaches to Automatic Schema Matching, VLDB Journal 10(4), 2001
50. Stuart J. Russell, Peter Norvig: Artificial Intelligence - A Modern Approach, Prentice Hall, 2002

51. Arnaud Sahuguet, Fabien Azavant: Building Light-weight Wrappers for Legacy Web Data-sources using W4F, VLDB 1999
52. Ralf Schenkel, Anja Theobald, Gerhard Weikum: Ontology-Enabled XML Search. in: H. Blanken et al. (Editors), Intelligent Search on XML Data, Springer, 2003
53. Ralf Schenkel, Anja Theobald, Gerhard Weikum: An Efficient Connection Index for Complex XML Document Collections, EDBT 2004
54. Torsten Schlieder, Holger Meuss: Querying and Ranking XML Documents, Journal of the American Society for Information Science and Technology (JASIST) 53(6), 2002
55. Torsten Schlieder, Holger Meuss: Result Ranking for Structured Queries against XML Documents, DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries, 2000
56. Torsten Schlieder, Felix Naumann: Approximate Tree Embedding for Querying XML Data, SIGIR Workshop on XML and Information Retrieval, 2000
57. Marios Skounakis, Mark Craven, Soumya Ray: Hierarchical Hidden Markov Models for Information Extraction, IJCAI 2003
58. Steffen Staab, Rudi Studer (Editors): Handbook on Ontologies, Springer 2004
59. Martin Theobald, Ralf Schenkel, Gerhard Weikum: Exploiting Structure, Annotation, and Ontological Knowledge for Automatic Classification of XML Data, International Workshop on Web and Databases (WebDB) 2003
60. Anja Theobald, Gerhard Weikum: Adding Relevance to XML. International Workshop on Web and Databases (WebDB) 2000, extended version in: LNCS 1997, Springer, 2001.
61. Anja Theobald, Gerhard Weikum: The Index-based XXL Search Engine for Querying XML Data with Relevance Ranking, EDBT 2002
62. Martin Theobald, Gerhard Weikum, Ralf Schenkel: Top-k Query Evaluation with Probabilistic Guarantees, VLDB 2004
63. Yuri A. Tijerino, David W. Embley, Deryle W. Lonsdale, George Nagy: Ontology Generation from Tables, WISE 2003
64. Ellen M. Voorhees: Query Expansion Using Lexical-Semantic Relations. SIGIR 1994
65. Ji-Rong Wen, Jian-Yun Nie, Hong-Jiang Zhang: Query Clustering Using User Logs, ACM TOIS 20(1), 2002
66. Linhao Xu, Chenyun Dai, Wenyuan Cai, Shuigeng Zhou, Aoying Zhou: T owards Adaptive Probabilistic Search in Unstructured P2P Systems. Asia-Pacific Web Conference (APWeb) 2004
67. Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Wei-Ying Ma, Hong-Jiang Zhang, Chao-Jun Lu: Implicit Link Analysis for Small Web Search, SIGIR 2003
68. Shipeng Yu, Deng Cai, Ji-Rong Wen, Wei-Ying Ma: Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation, WWW Conference 2003
69. Pavel Zezula, Giuseppe Amato, Fausto Rabitti: Processing XML Queries with Tree Signatures. in: H. Blanken et al. (Editors), Intelligent Search on XML Data, Springer, 2003