Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

ELSEVIER International Journal of Approximate Reasoning 34 (2003) 97–104

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Information retrieval in the Web: beyond current search engines ☆

Ricardo Baeza-Yates

Center for Web Research, Department of Computer Science, University of Chile, Blanco Encalada, 2120 Santiago, Chile

Received 1 January 2003; accepted 1 July 2003

Abstract

In this paper we briefly explore the challenges to expand information retrieval (IR) on the Web, in particular other types of data, Web mining and issues related to crawling. We also mention the main relations of IR and soft computing and how these techniques address these challenges.

1. Introduction

The Web has become the largest easy available repository of data. Hence, it is natural to extract information from it and Web search engines have become one of the most used tools in Internet. However, the exponential growth and the fast pace of change of the Web, makes really hard to retrieve all relevant information. In fact, crawling the Web is perhaps the main bottleneck for Web search engines. In addition, there is the unwritten assumption that a physical file is a logical document, which is not always true.

Recent work on the challenges of searching the Web include the following problems [10,20,25]:

☆ Funded by Millennium Nucleus Center for Web Research, Mideplan, Chile.
E-mail address: rbaeza@dcc.uchile.cl (R. Baeza-Yates).

0888-613X/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2003.07.002

- Keeping the index fresh and complete, including hidden content.
- Identifying and removing malicious content and linking, called search engine spam. Some authors call this problem *adversarial IR*.
- Identifying content of good quality. The Web is full of low quality (syntactic and semantically) content, including noisy, unreliable and contradictory data. Hence, we have the problem of how much a Web site can be trusted. This includes HTML structure, that in most cases is vague and heterogeneous.
- Exploiting user feedback, either from explicit user evaluation or implicitly from Web logs. We can include here implicit information given by the authors of Web pages in the form of several conventions used in HTML design.
- Detecting duplicate hosts and content, to avoid unnecessary crawling.
- Distinguishing the information need: informational, navigational, or transactional. It is estimated that less than 50% of the queries are of the first kind, which was the classic case.
- Improving the query language, adding the context of the information needed, such as genre or time.
- Improving ranking, in particular to make it dependent on the person posing the query. Relevance is based in personal judgments, so ranking based in user profiles or other user based context information can help. Here we can add quality, trust, and user feedback issues.

All these problems are difficult to understand without real data. More experimental results are then needed. Additional material can be found in [1,2,11].

The Web is more than plain HTML and other text dominant formats and we would like to search well other data types. Among them we have dynamic pages, multimedia objects, XML data and associated semantic information. If the Semantic Web becomes a reality in spite of all the social issues that need to be solved, we may have an XML-based Web, with standard semantic metadata and schema. In that possible world, information retrieval (IR) becomes easier, and even multimedia search is simplified. Spam should disappear in this setting and it is easier to recognize good content. On the other hand, new retrieval problems appear, such as XML processing and retrieval, and Web mining on structured data.

The concept of soft computing (SC) was introduced by Zadeh [28] as a synergy of methodologies which collectively provide a foundation for the conception, design, construction and utilization of information/intelligent systems. Some of main methodologies of SC are fuzzy logic, genetic algorithms, neural networks, rough sets, Bayesian networks, and other probabilistic techniques. The main characteristic of SC is that it is tolerant to imprecision, vagueness, partial truth, and approximation. The subjectivity,

vagueness, and imprecision are typical properties of any IR process. SC techniques have been used satisfactorily to improve IR processes. In particular, we think that its application to solve the different IR problems recently appeared in the Web can be of help.

We start covering data challenges, followed by a short introduction to Web mining, continuing with some ideas to partially solve the crawling problem. We end with a short description on the use of SC in IR.

2. Data challenges

There are several data issues that need to be addressed. Among them we have to mention hidden or dynamic pages, multimedia data, structured data, and semantic data. Next we describe each one of them, except hidden data, that is a particular case of generic data with the problem of restricted access.

2.1. Dynamic data

The static Web has become small compared to content generated on demand, in particular by querying in e-business or information services sites. Current crawling software can follow dynamic links, but that has to be done with care, as there might be no limits or even the same page can be generated again and again. Accessing pages behind query forms is even more difficult as the crawler does not have knowledge of the database. On the other hand, even if the database is known, asking all possible queries might be too time consuming (exponential on the size of the database) and even if we stick to simple queries, some of them might be never posed by real persons. Web services might be a partial solution to this problem if they allow to learn from the database and how people query in it. For example, obtaining the most frequent one thousand queries could be enough. Another possibility is analyzing the page as in [23].

2.2. Multimedia data

Multimedia data includes images, animations, audio in several forms, and video. All of them have no standard formats. Dominant ones are JPG, GIF and PNG for images, MP3 for music, Real Video or Quicktime for video, etc. The ideal solution is to search on any kind of data, including text, using the same model and with a single query language. This ambitious goal is probably not feasible.

For a particular data type we can develop a similarity model, and depending on the type the query language will change. For instance, query by example for images or query by humming for audio. All this area belongs more to image and signal processing rather than classical IR.

2.3. Structured data

Most data has some structure, leading to what is called *semi-structured data*. Examples are e-mail, news postings, etc. If XML becomes prevalent, the structure level is even higher. The first challenge is to design data models and associated query languages that allow to mix content and structure. Structured text was considered before XML and several efficiency/expressiveness trade-offs were designed [3]. After XML, the WWW Consortium has proposed XQuery as standard [27]. [1]

There are several challenges when retrieving XML data:

- The answer can be a fragment of XML and not necessarily a complete object. Nevertheless, the answers should also be XML based data.
- Many answers can appear in a single XML object, and they can overlap.
- How we can rank an answer and how ranking is inherited if we need to project the answer to certain structure types? Sometimes the combination of subtrees should have a better ranking if they are close, but in other cases is good if they are far apart.

Recent research on these topics is given in [6,7,17,18].

An additional problem is processing XML streams. That is, filtering a stream of XML objects with a large set of queries. Here the queries can be indexed, but not the data. See [24] for an introduction to this problem.

2.4. Semantic data

The two main problems with semantic information are standards for metadata that describe the semantic, and the quality or degree of trust of an information source. The first is being carried out by the WWW Consortium while the second needs certification schemes that must be developed in the future.

Other problems are common issues such as scaling, rate of change, lack of referential integrity (links are physical, not logical), distributed authority, heterogeneous content and quality, multiple sources, etc. An introduction to these and other challenges for the Semantic Web are presented in [8,21,26].

3. Web mining

In typical IR we know the query. Data mining is when we do not know the query. Hence, we try to find relations in the data that look like an interesting

[1] Although XPath and XSLT can also be considered as query languages, they were designed for different goals.

answer and then we study it to find the corresponding query. In the Web this leads to Web mining, a further challenge beyond IR on the Web. Some authors include IR as Web mining. However, we believe that this is incorrect. Web mining includes information extraction, followed by its generalization and analysis.

There are three types of Web data that can be mined: content, usage, and structure. Content includes text and multimedia mining. Usage includes Web log mining including search logs and other usage data. Structure implies analyzing the link structure of the Web (however this is ambiguous, considering the possibility of XML structure mining). In addition, for all the three cases we have a temporal dimension related to the dynamics of how the Web growths and changes. This implies temporal data. The first two types are covered in [14], while the third is the main topic of [12]. The later type is less studied and some results are presented in [4].

Web mining can be used for several purposes in addition to find new information or knowledge. It can be used for adaptive Web design (for example, user-driven Web design), Web site reorganization, Web site personalization, and several performance improvements.

4. Toward the perfect web search engine

A perfect search engine would solve the problems mentioned before, retrieving any type of data and collecting information to do better web mining. However, the bottleneck would be the same as today: gathering the data. The problem of crawling is related to volume and growth, together with duplicated and volatile data, and a very inefficient technique: pulling.

Current search engines perform their work without cooperation from the Web servers, they must transfer the pages using the standard HTTP protocol through TCP ASCII connections, and poll them to see if a page has been modified, to update their indexes after pulling updated or new pages.

It is more efficient to send an agent to the server, where it can locally search for new pages, links, and modified pages. It can also pack updated pages all together in a compressed file to be transferred to the search engine. The main search server could interact with the remote agent to decide if it is worth to transfer the existing batch based on several parameters such as the number of files, importance of them, etc. The intelligence of the crawler can then be distributed between the main search engine and the existent agents. Brandman et al. [9] study the impact on the bandwidth when Web servers publish metadata of their Web pages, such as actualization dates, size, etc. They show that there are savings and also the freshness of the pages increases. A similar paper focuses on freshness [19]. However, we can go one step further and instead of only pulling information, we can push it.

The interaction then drifts from pulling pages to pushing changes. As usual, the other extreme is also not efficient, as pushing too much will overload the centralized server. Hence, the best solution is that the server negotiates in advance with the agent when and how to send a message warning that a batch of changes is ready (or even better, that the changes have been already indexed and a partial index is available). Then, the main server will pull at due time those changes. This implies a long-term scheduling, which may find more changes when it really visits a Web server that pushed a warning. Nevertheless, this scheduling is simpler than current ones as we have more information, and we do not need to worry about politeness as we are sure that all accesses are not frequent and they are always successful.

In general, Web servers will want to cooperate in this architecture, because today it is an accepted value to be indexed on a popular search engine. On the other hand, even if they are spending CPU cycles on the search engine behalf, they are not being polled by the crawler, thus they are effectively diminishing their Web server access load. Also these cycles can be spent in periods of lower load.

As a first testing stage, while a global available agent platform is not available, a simple module, associated to the Web server, could be developed to provide a similar functionality and measure the performance improvement. As we already mentioned, small changes to the Web server have been suggested to enable cooperation with search engines [9,19], but they lack flexibility and they interfere with the crawler policies. Agents could improve a lot this behavior, enabling their algorithms to priorize pages to be embedded in the agents code. In this sense, the agent is an important component of the crawler's algorithm, and its logic follows a particular search engine's policies [5].

5. Soft computing and information retrieval

As we mentioned in the introduction, the term *soft computing* was introduced by Zadeh [28]. It refers to a synergy of methodologies useful for solving problems requiring some form of intelligence that diverts from traditional computing. SC provides a set of techniques appropriate for handling vagueness, subjectivity, and imprecision existing in several problems.

IR aims at modeling, designing, and implementing systems able to provide fast and effective content-based access to large amounts of information. The aim of an IR system is to estimate the relevance of information items to a user information need expressed in a query. This is a very hard and complex task, since it is pervaded with subjectivity, vagueness and imprecision.

SC includes different methodologies as fuzzy logic, genetic algorithms, neural networks, rough sets, and Bayesian networks. The IR problem is a

typical application field of SC. Some of the main SC approaches in IR are the following:

- Fuzzy sets and logic: information fusion, text extraction, query language models, and document clustering.
- Neural networks: document and term classification and clustering, and multimedia retrieval.
- Genetic algorithms: document classification, image retrieval, relevance feedback, and query learning.
- Probabilistic techniques: ranking, web mining.
- Rough sets and multivalued logics: document clustering.
- Bayesian networks: retrieval models, ranking, thesaurus construction, and relevance feedback.

There are at least one hundred papers devoted to the problems just enumerated, and listing all of them is matter of a full survey. Nevertheless, we refer the reader to Miyamoto's book [22] as well as the excellent volume edited by Crestani and Pasi [16], a special issue of IP&M [15], a survey paper by Chen [13] and this journal issue.

At least half of the problems mentioned in the introduction and in the subsequent sections can be addressed with the techniques above. Hence, further research lies ahead. The main drawbacks could be performance issues (for example, can be used in practical settings with bounded answer time?) and answer explanation (that is, for example, why a document is classified in a given category?). There have been recent applications of SC to IR on the Web that includes adaptive agents, user profiles, Web page categorization, quality assessment, etc. Hence, this shows that it is possible to progress in Web IR by using SC techniques.

Acknowledgements

We are grateful to the editors of this issue for the invitation to write this paper and their helpful comments to improve it.

References

[1] A. Arasu, J. Cho, H. Garcia-Molina, S. Raghavan, Searching the web, ACM Transactions on Internet Technologies 1 (1) (2001).
[2] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, England, 1999, 513p.
[3] R. Baeza-Yates, G. Navarro, Integrating contents and structure in text retrieval, SIGMOD Record 25 (1) (1996) 67–79.

[4] R. Baeza-Yates, F. Saint-Jean, C. Castillo, Web Dynamics, Structure and Page Ranking, SPIRE 2002, Springer LNCS, Lisbon, Portugal, 2002, pp. 117–130.
[5] R. Baeza-Yates, J. Piquer, Agents, Crawlers, and Web Retrieval, CIA 2002, Springer LNIA, Madrid, Spain, 2002, pp. 1–9.
[6] R. Baeza-Yates, D. Carmel, Y. Maarek, A. Sofer (Eds.), IR and XML, JASIST 53 (6) (2002) (Special issue).
[7] R. Baeza-Yates, N. Fuhr, Y. Maarek, Organizers. SIGIR Workshop on XML and IR, Tampere, Finland, 2002.
[8] R. Benjamins, J. Contreras, O. Corcho, A. Gomez-Perez, Six Challenges for the Semantic Web, KR2002 Workshop on Formal Ontology, Knowledge Representation and Intelligent Systems for the Web, Toulouse, France, 2002.
[9] O. Brandman, J. Cho, H. Garcia-Molina, N. Shivakumar, Crawler-friendly web servers, in: Workshop on Performance and Architecture of Web Servers (PAWS), June 2000.
[10] A. Broder, A taxonomy of web search, SIGIR Forum 36 (2) (2002).
[11] S. Chakrabarti, Recent results in automatic web resource discovery, ACM Computing Surveys (1999).
[12] S. Chakrabarti, Mining the Web: Discovering Knowledge from Hypertext Data, Morgan Kaufmann, 2003.
[13] H. Chen, Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms, Journal of the American Society for Information Science 46 (3) (1995) 194–216.
[14] R. Cooley, B. Mobasher, J. Srivastava, Web mining: information and pattern discovery on the world wide web, ICTAI, 1997, pp. 558–567.
[15] F. Crestani, G. Pasi (Eds.), Handling vagueness, subjectivity and imprecision in information access, Information Processing and Management, 39(2) (2003).
[16] F. Crestani, G. Pasi (Eds.), Soft Computing in Information Retrieval: Techniques and Applications, Physica-Verlag, Heidelberg, 2000.
[17] P. Fankhauser, A. Halevy, XML data management, The VLDB Journal 12 (1) (2003) (Special issue).
[18] L. Fegaras, R. Elmasri, Query engines for web-accessible XML data, The VLDB Journal (2001) 251–260.
[19] V. Gupta, R. Campbell, Internet search engine freshness by web server help, Technical Report UIUCDCS-R-2000-2153, Digital Computer Laboratory, University of Illinois at Urbana-Champaign, January 2000.
[20] M. Henzinger, R. Motwani, C. Silverstein, Challenges in web search engines, SIGIR Forum 36 (2) (2002).
[21] S. Lu, M. Dong, Ming, F. Fotouhi, The semantic web: opportunities and challenges for next-generation web applications, Information Research 7 (4) (2002).
[22] S. Miyamoto, Fuzzy Sets in Information Retrieval and Cluster Analysis, Kluwer Academic Publishers, 1990.
[23] S. Raghavan, H. Garcia-Molina, Crawling the hidden web, in: 27th International Conference on Very Large Data Bases, September 2001.
[24] D. Suciu, From searching text to querying XML streams, SPIRE 2002, Lisbon, Portugal, pp. 11–26.
[25] H. Tirri, Search in vain: challenges for internet search, IEEE Computer (January) (2003) 115–116.
[26] F. van Harmelen, How the Semantic Web will change KR: challenges and opportunities for a new research agenda, The Knowledge Engineering Review 17 (1) (2002).
[27] WWW Consortium. XML Query Language Proposal. Available from <http://www.w3.org/XML/Query/>.
[28] L.A. Zadeh, Fuzzy logic, neural networks and soft computing, Communication of ACM 37 (3) (1994) 77–84.