ELSEVIER
Artificial Intelligence 118 (2000) 1–14
Artificial Intelligence
www.elsevier.com/locate/artint

Editorial

Intelligent Internet systems

Alon Y. Levy *, Daniel S. Weld
Department of Computer Science & Engineering, University of Washington, Box 352350, Seattle, WA 98195-2350, USA

Received 16 February 2000

1. Introduction

The astonishing growth of the Internet is the first sign that every aspect of our economy and society are likely to change. Yet for people to realize the vast promise of networked computing, Internet applications must become dramatically more powerful and easier to use. Artificial Intelligence (AI) technology holds the key to these futuristic applications with the promise of advanced features, adaptive functionality and intuitive interfaces.

We group Internet applications into four categories:
(1) user modeling,
(2) discovery and analysis of remote information sources,
(3) information integration, and
(4) Web-site management.

The seven papers in this special issue represent some of the latest and most exciting research in three of the four categories.[1] This introduction attempts to place the special-issue papers in context, but we caution readers that the field is too young and moving too quickly for a comprehensive survey article.

2. User modeling

Although user modeling has a long history in AI, cognitive science, and computer-aided instruction [21,27,52,111], recent research illuminates the technology's application to intelligent user interfaces and networked recommendation systems.

A popular architecture uses machine learning algorithms to develop a predictive model of a user's behavior as a function of the task attributes or data from other users. These learning systems have been applied to tasks such as meeting scheduling [31], email

* Corresponding author. Email: alon@cs.washington.edu.
[1] Unfortunately, no suitable papers were submitted in the area of user modeling.

PII: S0004-3702(00)00013-8

processing [86], netnews filtering [16,108], Web search [75], book and music recommendations [85,110], intrusion dection [73], and Web browsing recommendations [63,84,97, 98,124]. For example, when run on data from a faculty member's scheduling behavior, the Calendar Apprentice [31,88] might learn rules such as "Meetings with undergraduates have duration 30 minutes, and take place in my office" while "Meetings with the Dean have duration 60 minutes and take place in the Dean's office".

A range of machine learning algorithms have been applied to the user modeling problem: tree learning [105], neural network backpropagation [109], nearest neighbor [28], the naive Bayesian classifier [33], and various statistical techniques such as mean-squared difference and the Pearson-R measure [110].

Early work in the area distinguished between systems that made predictions based on features in the task domain (e.g., who is the meeting with?) and so-called collaborative filtering systems [108] that developed correlations between the behavior of different individuals. Obviously, the latter approach is only possible if a single system has access to the behavior of many people, but Internet-based systems make this a common occurrence, and indeed e-commerce sites such as `Amazon.com` now use collaborative filtering to generate personalized product recommendations. Rather than focusing on a strictly task-feature approach or a strict collaborative filtering approach, recent research has shown that predictive accuracy can be greatly improved by using the methods in the inductive classification framework and learning on explicit social features (e.g., "Jane liked Titanic") as well as content-based features [11,12,54].

Finally, the ReferralWeb [64] offers an interesting twist on Internet search engines—rather than link people to authoritative Web pages, ReferralWeb aims to direct people (or their email questions) to humans who are experts on a given topic. Naturally, this casts a new spin on user modeling and raises some interesting privacy concerns [74].

3. Discovery and analysis of information sources

As anyone who has explored the Internet knows well, a bewildering array of sites come online in ever increasing numbers. As a result, discovering and exploring the range of useful information sources is a Sisyphusian task. Thus it is no surprise that researchers have attempted to apply AI techniques to the problem of automatically discovering and analyzing Internet information sources. Following [99,100], we classify this work as addressing the following four questions:

- **Discovery:** How does an agent find new and unknown information sources? For example, a new stock-quote server has just come on the Web; how should a machine find it?
- **Extraction:** What are the mechanics of accessing an information source and parsing its responses? For example, the stock-quote server is queried by providing a company's ticker symbol to a specific CGI script, and the service responds with an HTML page containing a 4-tuple of data.
- **Translation:** Having parsed the source's response into tokens, how does the agent assign semantics to the resulting tokens? For example, the first element of the tuple is the stock name, the second is current price, etc.

- **Evaluation:** What is the accuracy, reliability and scope of the information source? For example, the source contains only companies listed on the NYSE and quotes are delayed by 20 minutes.

Unfortunately, we know of no significant research that addresses the problem of resource discovery. Most researchers have focused on the problem of extraction; indeed, two of the special issue's papers focus on this subproblem. However, there has been some intriguing work addressing the questions of translation and evaluation.

3.1. Extraction

Shopbot [34,99] was one of the first systems to tackle automated extraction from Web resources, specifically Internet stores. As input, Shopbot took an URL, the relational schema it hoped to populate, and a set of common attribute values for said schema. For example, it might be given the URL for `amazon.com`, be told that books have author names, titles, publishers, and prices, and be given the authors and titles for some common books. Shopbot searched the Web starting from the input URL, looking for HTML forms, probing such forms with common attributes, and classifying responses. Pages that were deemed likely product listings (as opposed to, say, registration or help pages) were converted into an abstraction of HTML and mined for common patterns which lead in turn to a parser. Although Shopbot was entirely heuristic, it was surprisingly successful.

Subsequent work added considerable rigor to the field. Kushmerick et al. [70] defined the problem of wrapper induction, identified a class of information sources for which wrappers could be automatically constructed, and presented algorithms to do precisely that. An extended version of Kushmerick's work [69] is included in this issue. Subsequent work [57,91] present learning algorithms for wrapper classes that are substantially more expressive than Kushmerick's classes, allowing missing attributes, variant attribute orderings, disjunctive delimiters, etc. Freitag [49] describes how grammatical inference can improve the precision of the data extracted from information sources.

Other researchers have addressed semi-automatic extraction. Ashish and Knoblock [9] present a tool which automates the bulk of the wrapper-generation process with a combination of heuristics that exploit the page's HTML parse tree. Bauer [13] uses programming by demonstration (PBD) to construct wrappers.

Other approaches to extraction use Hidden Markov Models (HMMs) [106]. For example, the Cora project [87] defines extraction using HMMs by associating a class (e.g., title or price) with each state. States emit words from a class-specific unigram distribution. By applying the Viterbi algorithm to previously unseen text, the most likely state sequence is produced, and this can be used to label parts of the text with a class.

Several authors have developed machine learning systems that generate pattern-based extraction rules (see [20,49,58,112,113]).

Craven et al. [29] have attempted something even more ambitious than simple information extraction; they seek to autonomously build AI knowledge bases by a combination of extracting data to populate predefined relations and inducing new relations from Web structure. An extended description of their bold endeavor [30] is included in this special issue.

The natural language community has long considered problems similar to information extraction from Web resources. Indeed, the work on message understanding is more ambitious, since the input is unstructured English (requiring anaphora resolution, discourse analysis, etc.) rather than being formatted in some simple tabular or regular form. Soderland, however, has successfully adapted MUC techniques.

An interesting application of information extraction is the automatic identification (and elimination) of advertisements from Web pages; see Kushmerick's work in this area [68].

3.2. Translation

Because formats such as XML are likely to reduce the importance of the extraction problem, we expect semantic translation to attract increasing attention. Perkowitz and Etzioni [100] describe the correspondence heuristic, which allows a learner to use its knowledge of one data source to learn another. Levy and Ordille [80] present a system that learns descriptions of CCSO name servers; while their approach requires that a human instructor provide good examples to the learner, the system is relatively robust. Li [83] learns mappings between semantic categories in different relational databases by examining both the format and content of fields. Several authors have considered the problem of automatically (or semi-automatically) finding mappings between disparate relational schema [15,93]. The problem of merging ontologies has also been considered in the knowledge acquisition and ontology communities [92].

The Cora project [87] has a component which semi-automatically classifies documents into a Yahoo-like hierarchy by bootstrapping. For example, they manually created a 70-leaf hierarchy of Computer Science topics and associated a few keywords with each node. Given this relatively easy to generate input, their system automatically classifies unseen documents into the right node. They start by using keywords as an input to a rule learner that builds a preliminary classifier which is noisy and incomplete. Next, using documents and preliminary labels, they use the naive Bayesian classifier to make an improved classification. Finally, they use expectation maximization and statistical shrinkage to improve their predictions. The results are impressive, almost as good as the labels produced manually.

One component of semantic mapping is the ability to match objects which are named slightly differently at different sites (or even at the same site). For example, how does one determine that "Dan Weld" is the same individual as "Daniel S. Weld"? While this question has been considered at length in the database literature, Cohen's paper in this issue [26] offers a promising new approach.

3.3. Evaluation

Both the Google search engine [18] and Kleinberg's hub and authority model [65] use hypertext link structure to estimate the overall quality of a Web page, but we know of no work that attempts to automatically evaluate the accuracy, reliability or scope of information sources returning relational or semistructured data.

It seems that this topic is ripe for study, however, since a number of researchers have developed representations for encoding such judgments if they could be automatically produced. A logical formulation of the (conditional or local) completeness of information sources is considered in [2,35,39,40,50,77,90], while a probabilistic formalism is developed in [45]. For the most part, these papers focus on algorithms for choosing optimally between sources, leaving the construction of such resource descriptions as an open problem. Motro and Rakov's work [89] is an exception; they suggest a combined manual/statistical approach to rating databases, resulting in quality specifications that are expressive enough to represent variations in quality across different sections of the database.

4. Information integration

The next step after discovery and analysis of information sources is to be able to seamlessly integrate data from multiple sources. This problem has attracted significant attention in the AI community (mostly from knowledge representation and planning) and in the database systems community. The goal of a data integration system is to provide a *uniform* interface to a multitude of data sources. A heavily used example is the task of providing information about movies from data sources on the World-Wide Web (WWW). There are numerous sources on the WWW concerning movies, such as the Internet Movie Database (providing comprehensive listings of movies, their casts, directors, genres, etc.), MovieLink (providing playing times of movies in US cities), and several sites providing reviews of selected movies. Suppose we want to find the names and reviews of all movies starring Matt Damon which are playing tonight in Seattle. None of these data sources *in isolation* can answer this query. However, by combining data from multiple sources, we can answer queries like this one, and even more complex ones. To answer our query, we would first search the Internet Movie Database for the list of movies starring Matt Damon, and then feed the result into the MovieLink database to check which ones are playing in Seattle. Finally, we would find reviews for the relevant movies using any of the movie review sites.

Several systems have been built with the goal of answering queries using a multitude of Web sources [4,7,14,25,38,41,50,51,81,121]. Many of the problems encountered in building these systems are similar to those addressed in building heterogeneous database systems [3,17,48,56,60,114,120]. Web data integration systems have, in addition, to deal with

(1) large and evolving number of Web sources,
(2) little meta-data about the characteristics of the source, and
(3) larger degree of source autonomy.

There are two main differences between data integration systems and traditional database systems. First, as explained in the previous section, instead of obtaining the data from a local store, the system communicates with the data sources through wrappers. The role of the wrappers is to translate the data from the format of the source into a format that can be manipulated by the data integration system. Second, users of data integration systems do not pose queries directly in the schema in which the data is stored. Instead, the user poses queries on a *mediated schema*. The reason for this is that one of the principal goals of a data integration system is to free the user from having to know about the specific data sources and interact with each one. A mediated schema is a set of *virtual* relations, which are designed for a particular data integration application. As a consequence, the data integration system must first *reformulate* a user query into a query that refers directly to the schemas in the sources.

We classify the problems addressed in the area of information integration as follows:

Specification of mediated schema and reformulation. In order for the system to be able to reformulate a user query, it needs to have a set of source descriptions, specifying the semantic mappings between the relations in the sources and the relations in the mediated schema. Broadly speaking, several approaches have been considered for describing data sources:

- *Global as view* (GAV) [3,48,51,56,95,114]: the mediated schema is described as a set of queries (or database views) over the source schemas. In this case, reformulation amounts to unfolding the user's query.
- *Local as view* (LAV) [37,38,50,71,72,81]: the data sources are described as queries over the relations in the mediated schema. Here query reformulation reduces to the problem of answering queries using views [23,36,79,107,116,122].
- *Description Logics:* [22,81]: the mediated schema and the data sources are described as a terminology in some Description Logic. Query reformulation makes use of the subsumption and satisfiability algorithms provided by the Description Logic system.
- *Planning operators:* [7,41,53,71]: data sources are described as a set of planning operators, and query reformulation is posed as a planning problem.

Completeness of data in Web sources. In general, sources that we find on the WWW are not necessarily complete for the domain they are covering. For example, a bibliography source is unlikely to be complete for the field of Computer Science. However, in some cases, we can assert completeness statements about sources. For example, the DB&LP Database [2] has the complete set of papers published in most major database conferences. Knowledge of completeness of a Web source can help a data integration system in several ways. Most importantly, since a *negative* answer from a complete source is meaningful, the data integration system can prune access to other sources. The problem of describing completeness of Web sources and using this information for query processing is addressed in [2,35,39,40,50,77,90]. The work described in [45] describes a probabilistic formalism for describing the contents and overlaps among information sources, and presents algorithms for choosing optimally between sources.

Differing query processing capabilities. From the perspective of the Web data integration system, the Web sources appear to have vastly differing query processing capabilities, and these can result in serious performance effects. The main reasons for the different appearance are

(1) the underlying data may actually be stored in a structured file or legacy system and in this case the interface to this data is naturally limited, and
(2) even if the data is stored in a traditional database system, the Web site may provide only limited access capabilities for reasons of security or performance.

[2] http://sunsite.informatik.rwth-aachen.de/dblp/db/index.html.

To build an effective data integration system, these capabilities need to be explicitly described to the system, adhered to, and exploited as much as possible to improve performance. We distinguish two types of capabilities: Negative capabilities that limit the access patterns to the data, and positive capabilities, where a source is able to perform additional algebraic operations in addition to simple data fetches.

The main form of negative capabilities is limitations on the binding patterns that can be used in queries sent to the source. For example, it is not possible to send a query to the Internet Movie Database asking for *all* the movies in the database and their casts. Instead, it is only possible to ask for the cast of a *given* movie, or to ask for the set of movies in which a particular actor appears. Several works have considered the problem of answering queries in the presence of binding pattern limitations [46,50,71,81,107].

Positive capabilities pose another challenge to a data integration system. If a data source has the ability to perform operations such as selections and joins, we would like to push as much as possible of the processing to the source, thereby hopefully reducing the amount of local processing and the amount of data transmitted over the network. The problem of describing the computing capabilities of data sources and exploiting them to create query execution plans is considered in [56,82,96,114,119].

Query optimization. After the minimal set of data sources has been selected for a given query, a key problem is to find the *optimal* query execution plan for the query. The query execution plan specifies the order and scheduling in which the sources are accessed and the particular algorithms used to combine the data from the sources (e.g., join algorithms). This problem is analogous to the query optimization problem faced in database systems, except that it is complicated here because we have few statistics about the underlying data sources, and because there may be significant delays in data transmission due to network traffic. This problem has been considered in several works [56,61,118,123]. The paper by Ambite and Knoblock in this issue [5] presents an algorithm for query optimization that combines the reformulation and optimization phases using a transformational approach. The paper by Cohen in this issue [26] describes the WHIRL system that considers the problem of quickly obtaining the first few answers to the query. WHIRL focuses on the important case where matching object names between different sources may require fuzzy matches, rather than exact matches. The BIG system, described in this issue's paper by Lesser et al. [76] addresses several additional issues related to information gathering, including the resource tradeoffs of different information gathering plans, extraction of data from unstructured sources and using the extracted data to further refine the search. A followup system to BIG is described by Grass and Zilberstein [55].

We refer the reader to several workshop proceedings [42,66,67] and several surveys [47, 59,78,117] for a more detailed description of work in this area.

5. Web-site management

A final area in which AI techniques have significant potential to contribute to Web-based systems is the flexible construction and intelligent modification of data intensive Web sites. Web sites typically contain and integrate several bodies of data about the enterprise they are describing, and these bodies of data are linked into a rich navigational structure. For example, a company's internal Web site may contain data about its employees, linked to data about the products they produce and/or to the customers they serve. The *data* in a Web site and the *structure* of the links in the site can be viewed as a richly structured knowledge base.

Several projects in the database community have taken a first stab at constructing tools for principled construction of Web sites [6,8,10,24,43,62,94,115]. The key ideas underlying these systems are:

(1) The Web site's structure, content, and graphical layout should be specified independently of one another.
(2) *Declarative* representations are the best way to specify the structural aspects of the site (as well as many forms of the site's content).

Of course, most large Web sites are already driven by content stored in (multiple) relational databases, and the techniques of the previous section can be used to simplify the integration of such data, but what does it mean to specify the structure of a site declaratively? When run on the underlying data, the site specification query defines the *Web-site graph* which is a logical representation of the pages in the site, links between them and the data presented at every page. For example, the query might force a link from the University course nodes to corresponding faculty nodes whenever the `Teaches(Course, Faculty, CurQtr)` relation was true. Finally, the presentation of the pages in the site is specified using a set of HTML templates.

From a representational point of view, a key feature that distinguishes these systems from common database applications is that they consider the data to be *semi-structured* [1,19], and hence represented as possibly irregular graph structure as opposed to rigid relations. The query languages used in these systems take graphs as input and produce a graph as output (as opposed to SQL that is a function from relations to a relation). It is interesting to note that there are recent emerging standards from the W3C for each one of these steps, namely XML for representing data, a query language for XML (e.g., XML-QL [32]) for specifying the site structure, and XSLT for HTML templates.

The main advantage of declarative Web-site management systems is the ability to easily *restructure* a Web site and to construct multiple versions of a Web site from the same underlying content (e.g., consider a company that creates an internal Web site for its employees and several external ones for its customers, suppliers, or other affiliate companies).

From the perspective of AI, these tools provide a platform on which one can start tackling higher-level issues in managing Web sites, such as the following.

Automatically restructuring Web sites. The short experience in building Web sites has already shown that it is a highly iterative process. Even after the Web site is up, designers will frequently want to restructure it after understanding the patterns with which users browse the site. Furthermore, it is rare that one site structure is appropriate for all classes of users. The work by Perkowitz and Etzioni [101–103] pioneered the field of *adaptive* Web sites; an extended description of their work is presented in this issue [104]. Such sites restructure themselves depending on usage patterns. The site can be adapted for classes of users or individual users. The key challenges involved are to infer from the browsing patterns the interesting structures of the site that may be useful for a class of users.

Enforcing integrity constraints on Web sites. As builders of Web sites, we would like to enforce constraints on the structure of our site (e.g., no dangling pointers, an employee's homepage should point to their department's homepage, etc.). Clearly, once we have created the Web site, we can go through it and check whether the constraints are satisfied, but in that case, we would have to repeat the check every time the Web site is updated. A more interesting approach is to reason about that a certain integrity constraint will hold for every site generated by this query, irrespective of the underlying data. Such an approach is described in [44].

6. Conclusions

We are in the midst of very exciting times. We are using the Internet to perform a growing number of everyday tasks, both as individual users and as members of societies. As such, providing tools for aiding in these tasks provides a gold mine of challenges for Artificial Intelligence. The sheer scale of the Internet often necessitates the use of approximate and heuristic techniques that form the core of many AI solutions.

The papers included in this issue provide only the first step in applying AI to research problems related to the Internet. Fortunately, research problems in this area are easy to find; since we are all users of the Internet, we know well the limitations of currently available tools. Validating the solutions we devise is also often easier, because the Internet provides a open, level experimental ground. Finally, deploying our solutions provides a unique opportunity to study how AI techniques can be most effectively embedded within larger systems.

Acknowledgements

We thank our colleagues at the University of Washington (and elsewhere!) for furthering our understanding of research in these areas. We also wish to thank Corin Anderson, Nick Kushmerick, and Tessa Lau for comments on this article. This work was funded by Office of Naval Research Grant N00014-98-1-0147, by National Science Foundation Grants IRI-9303461 and IIS-9978567, by ARPA/Rome Labs grant F30602-95-1-0024, and by a Sloan Fellowship.

References

[1] S. Abiteboul, Querying semi-structured data, in: Proc. International Conference on Database Theory (ICDT), Delphi, Greece, 1997.

[2] S. Abiteboul, O. Duschka, Complexity of answering queries using materialized views, in: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Seattle, WA, 1998.

[3] S. Adali, K. Candan, Y. Papakonstantinou, V.S. Subrahmanian, Query caching and optimization in distributed mediator systems, in: Proc. ACM SIGMOD Conference on Management of Data, Montreal, Quebec, 1996.
[4] J.L. Ambite, N. Ashish, G. Barish, C.A. Knoblock, S. Minton, P.J. Modi, I. Muslea, A. Philpot, S. Tejada, ARIADNE: A system for constructing mediators for Internet sources (system demonstration), in: Proc. ACM SIGMOD Conference on Management of Data, Seattle, WA, 1998.
[5] J.L. Ambite, C.A. Knoblock, Flexible and scalable cost-based query planning in mediators: A transformational approach, Artificial Intelligence 118 (2000) 115–161 (this issue).
[6] C.R. Anderson, A.Y. Levy, D.S. Weld, Declarative Web-site management with Tiramisu, in: Proc. International Workshop on The Web and Databases (WebDB), 1999.
[7] Y. Arens, C.A. Knoblock, W.-M. Shen, Query reformulation for dynamic information integration, International J. Intelligent and Cooperative Information Systems 6 (2/3) (1996) 99–130.
[8] G. Arocena, A. Mendelzon, WebOQL: Restructuring documents, databases and Webs, in: Proc. International Conference on Data Engineering (ICDE), Orlando, FL, 1998.
[9] N. Ashish, C. Knoblock, Semi-automatic wrapper generation for Internet information sources, in: Proc. Cooperative Information Systems, 1997.
[10] P. Atzeni, G. Mecca, P. Merialdo, Design and maintenance of data-intensive Web sites, in: Proc. Conference on Extending Database Technology (EDBT), Valencia, Spain, 1998.
[11] M. Balabanovic, Y. Shoham, Fab: Content-based, collaborative recommendation, 1997.
[12] C. Basu, H. Hirsh, W. Cohen, Recommendation as classification: Using social and content-based information in recommendation, in: Proc. AAAI-98, Madison, WI, AAAI Press, Menlo Park, CA, 1998, pp. 714–720.
[13] M. Bauer, D. Dengler, G. Paul, Instructible information agents for Web mining, in: Proc. 2000 Conference on Intelligent User Interfaces, 2000.
[14] C. Beeri, G. Elber, T. Milo, Y. Sagiv, O. Shmueli, N. Tishby, Y. Kogan, D. Konopnicki, P. Mogilevski, N. Slonim, Websuite-a tool suite for harnessing Web data, in: Proc. International Workshop on the Web and Databases, Valencia, Spain, 1998.
[15] S. Bergamaschi, S. Castano, M. Vincini, Semantic integration of semistructured and structured data sources, SIGMOD Record 28 (1) (1999) 54–59.
[16] D. Billsus, M. Pazzani, A hybrid user model for news story classification, in: Proc. 7th International Conference on User Modelling, 1999, pp. 99–108.
[17] J.A. Blakeley, Data access for the masses through OLE DB, in: Proc. ACM SIGMOD Conference on Management of Data, Montreal, Quebec, 1996, pp. 161–172.
[18] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, in: Proc. 7th World-Wide Web Conference, Melbourne, Australia, 1998.
[19] P. Buneman, Semistructured data, in: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Tucson, AZ, 1997, pp. 117–121.
[20] M. Califf, R. Mooney, Relational learning of pattern-match rules for information extraction, in: Workshop in Natural Language Learning, Conference Assoc. Computational Linguistics, 1997.
[21] S. Carberry, Modeling the user's plans and goals, Computational Linguistics 14 (3) (1988) 23–37.
[22] T. Catarci, M. Lenzerini, Representing and using interschema knowledge in cooperative information systems, J. Intelligent and Cooperative Information Systems (1993).
[23] S. Chaudhuri, R. Krishnamurthy, S. Potamianos, K. Shim, Optimizing queries with materialized views, in: Proc. International Conference on Data Engineering (ICDE), Taipei, Taiwan, 1995.
[24] S. Cluet, C. Delobel, J. Simeon, K. Smaga, Your mediators need data conversion, in: Proc. ACM SIGMOD Conference on Management of Data, Seattle, WA, 1998.
[25] W. Cohen, Integration of heterogeneous databases without common domains using queries based on textual similarity, in: Proc. ACM SIGMOD Conference on Management of Data, Seattle, WA, 1998.
[26] W.W. Cohen, WHIRL: A word-based information representation language, Artificial Intelligence 118 (2000) 163–196 (this issue).
[27] A. Collins, D. Gentner, Constructing runnable mental models, in: Proc. 4th Annual Conference of the Cognitive Science Society, Ann Arbor, MI, 1982.
[28] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inform. Theory 13 (1967) 21–27.

[29] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery, Learning to extract symbolic knowledge from the World Wide Web, in: Proc. AAAI-98, Madison, WI, 1998.
[30] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery, Learning to construct knowledge bases from the World Wide Web, Artificial Intelligence 118 (2000) 69–113 (this issue).
[31] L. Dent, J. Boticario, J. McDermott, T. Mitchell, D. Zabowski, A personal learning apprentice, in: Proc. AAAI-92, San Jose, CA, 1992, pp. 96–103.
[32] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, D. Suciu, A query language for XML, in: Proc. World-Wide Web 8 Conference, Toronto, Ont., 1999.
[33] P. Domingos, M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss, Machine Learning 29 (1997) 103–130.
[34] R. Doorenbos, O. Etzioni, D. Weld, A scalable comparison-shopping agent for the World-Wide Web, in: Proc. First International Conference Autonomous Agents, Marina del Rey, CA, 1997, pp. 39–48.
[35] O. Duschka, Query optimization using local completeness, in: Proc. AAAI-97, Providence, RI, 1997.
[36] O. Duschka, M. Genesereth, A. Levy, Recursive query plans for data integration, J. Logic Programming (Special Issue on Logic Based Heterogeneous Information Systems) (1999).
[37] O.M. Duschka, M.R. Genesereth, Answering recursive queries using views, in: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Tucson, AZ, 1997.
[38] O.M. Duschka, M.R. Genesereth, Query planning in infomaster, in: Proc. ACM Symposium on Applied Computing, San Jose, CA, 1997.
[39] O. Etzioni, K. Golden, D. Weld, Sound and efficient closed-world reasoning for planning, Artificial Intelligence 89 (1–2) (1997) 113–148.
[40] O. Etzioni, K. Golden, D. Weld, Tractable closed-world reasoning with updates, in: Proc. Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR-94), Bonn, Germany, 1994, pp. 178–189.
[41] O. Etzioni, D. Weld, A softbot-based interface to the Internet, Comm. ACM 37 (7) (1994) 72–76.
[42] D. Fensel, C. Knoblock, N. Kushmerick, M.-C. Rousset, in: Proc. IJCAI Workshop on Intelligent Information Integration, Stockholm, Sweden, 1999.
[43] M. Fernandez, D. Florescu, J. Kang, A. Levy, D. Suciu, Catching the boat with Strudel: Experiences with a Web-site management system, in: Proc. ACM SIGMOD Conference on Management of Data, Seattle, WA, 1998.
[44] M. Fernandez, D. Florescu, A. Levy, D. Suciu, Verifying integrity constraints on Web-sites, in: Proc. IJCAI-99, Stockholm, Sweden, 1999.
[45] D. Florescu, D. Koller, A. Levy, Using probabilistic information in data integration, in: Proc. International Conference on Very Large Data Bases (VLDB), Athens, Greece, 1997, pp. 216–225.
[46] D. Florescu, A. Levy, I. Manolesu, D. Suciu, Query optimization in the presence of limited access patterns, in: Proc. ACM SIGMOD Conference on Management of Data, 1999.
[47] D. Florescu, A. Levy, A. Mendelzon, Database techniques for the World-Wide Web: A survey, SIGMOD Record 27 (3) (1998) 59–74.
[48] D. Florescu, L. Raschid, P. Valduriez, A methodology for query reformulation in cis using semantic knowledge, International Journal of Intelligent and Cooperative Information Systems (Special Issue on Formal Methods in Cooperative Information Systems) 5 (4) (1996).
[49] D. Freitag, Information extraction from HTML: Application of a general machine learning approach, in: Proc. AAAI-98, Madison, WI, 1998, pp. 517–523.
[50] M. Friedman, D. Weld, Efficient execution of information gathering plans, in: Proc. IJCAI-97, Nagoya, Japan, 1997.
[51] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, J. Widom, The TSIMMIS project: Integration of heterogeneous information sources, J. Intelligent Information Systems 8 (2) (1997) 117–132.
[52] D. Gentner, A. Stevens (Eds.), Mental Models, Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.
[53] K. Golden, O. Etzioni, D. Weld, Omnipotence without omniscience: Sensor management in planning, in: Proc. AAAI-94, Seattle, WA, 1994, pp. 1048–1054.
[54] N. Good, J. Schafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, J. Riedl, Combining collaborative filtering with personal agents for better recommendations, in: Proc. AAAI-99, Orlando, FL, 1999.

[55] J. Grass, S. Zilberstein, Value-driven information gathering, in: Proc. AAAI Workshop on Building Resource-Bounded Reasoning Systems, Providence, RI, 1997.
[56] L. Haas, D. Kossmann, E. Wimmers, J. Yang, Optimizing queries across diverse data sources, in: Proc. International Conference on Very Large Data Bases (VLDB), Athens, Greece, 1997.
[57] C. Hsu, M. Dung, Generating finite-state transducers for semistructured data extraction from the Web, J. Information Systems 23 (8) (1998).
[58] S. Huffman, Learning information extraction patterns from examples, in: S. Wermter, E. Riloff, G. Scheler (Eds.), Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing, Springer, Berlin, 1996.
[59] R. Hull, Managing semantic heterogeneity in databases: A theoretical perspective, in: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Tucson, AZ, 1997, pp. 51–61.
[60] R. Hull, G. Zhou, A framework for supporting data integration using the materialized and virtual approaches, in: Proc. ACM SIGMOD Conference on Management of Data, Montreal, Quebec, 1996, pp. 481–492.
[61] Z. Ives, D. Florescu, M. Friedman, A. Levy, D. Weld, An adaptive query execution engine for data integration, in: Proc. ACM SIGMOD Conference on Management of Data, 1999.
[62] R. Jakobovits, J.F. Brinkley, Managing medical research data with a Web-interfacing repository manager, in: American Medical Informatics Association Fall Symposium, Nashville, TN, 1997, pp. 454–458.
[63] T. Joachims, D. Freitag, T. Mitchell, Webwatcher: A tour guide for the World Wide Web, in: Proc. IJCAI-97, Nagoya, Japan, 1997, pp. 770–775.
[64] H. Kautz, B. Selman, M. Coen, S. Ketchpel, C. Ramming, An experiment in the design of software agents, in: Proc. AAAI-94, Seattle, WA, 1994.
[65] J. Kleinberg, Authoritative sources in a hyperlinked environment, in: Proc. 9th ACM-SIAM Symposium on Discrete Algorithms, 1998.
[66] C. Knoblock, A. Levy (Eds.), Proc. AAAI Workshop on Intelligent Data Integration, Madison, WI, AAAI Press, Menlo Park, CA, 1998.
[67] C.A. Knoblock, A.Y. Levy (Eds.), Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments, American Association for Artificial Intelligence, 1995.
[68] N. Kushmerick, Learning to remove Internet advertisements, in: Proc. 3rd Annual Conference on Autonomous Agents, 1999, pp. 175–181.
[69] N. Kushmerick, Wrapper induction: Efficiency and expressiveness, Artificial Intelligence 118 (2000) 15–68 (this issue).
[70] N. Kushmerick, D. Weld, R. Doorenbos, Wrapper induction for information extraction, in: Proc. IJCAI-97, Nagoya, Japan, 1997.
[71] C.T. Kwok, D.S. Weld, Planning to gather information, in: Proc. AAAI-96, Portland, OR, 1996.
[72] E. Lambrecht, S. Kambhampati, S. Gnanaprakasam, Optimizing recursive information gathering plans, in: Proc. IJCAI-99, Stockholm, Sweden, 1999, pp. 1204–1210.
[73] T. Lane, C.E. Brodley, An application of machine learning to anomaly detection, in: Proc. 20th Annual National Information Systems Security Conference, Vol. 1, 1997, pp. 366–380.
[74] T. Lau, O. Etzioni, D. Weld, Privacy interfaces for information management, Comm. ACM (1999).
[75] T. Lau, E. Horvitz, Patterns of search: Analyzing and modeling Web query refinement, in: Proc. 7th International Conference on User Modelling, 1999, pp. 119–128.
[76] V. Lesser, B. Horling, F. Klassner, A. Raja, T. Wagner, S.XQ. Zhang, BIG: An agent for resource-bounded information gathering and decision making, Artificial Intelligence 118 (2000) 197–244 (this issue).
[77] A.Y. Levy, Obtaining complete answers from incomplete databases, in: Proc. International Conference on Very Large Data Bases (VLDB), Bombay, India, 1996.
[78] A.Y. Levy, Combining artificial intelligence and databases for data integration, in: M. Wooldridge, M.M. Veloso (Eds.), Artificial Intelligence Today: Recent Trends and Developments, Lecture Notes in Comput. Sci., Vol. 1600, Springer, Berlin, 1999, pp. 249–268.
[79] A.Y. Levy, A.O. Mendelzon, Y. Sagiv, D. Srivastava, Answering queries using views, in: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), San Jose, CA, 1995.
[80] A.Y. Levy, J.J. Ordille, An experiment in integrating internet information sources, in: Working Notes of the AAAI Fall Symposium on AI Applications in Knowledge Navigation, 1995.

[81] A.Y. Levy, A. Rajaraman, J.J. Ordille, Querying heterogeneous information sources using source descriptions, in: Proc. International Conference on Very Large Data Bases (VLDB), Bombay, India, 1996.
[82] A.Y. Levy, A. Rajaraman, J.D. Ullman, Answering queries using limited external processors, in: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Montreal, Quebec, 1996.
[83] W.-S. Li, Knowledge gathering and matching in heterogeneous databases, in: Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogeneous, Distributed Environments, Stanford University, AAAI Press, Menlo Park, CA, 1995.
[84] H. Lieberman, Letizia: An agent that assists Web browsing, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 924–929.
[85] P. Maes, Agents that reduce work and information overload, Comm. ACM 37 (7) (1994) 31–40, 146.
[86] P. Maes, R. Kozierok, Learning interface agents, in: Proc. AAAI-93, Washington, DC, 1993, pp. 459–465.
[87] A. McCallum, K. Nigam, J. Rennie, K. Seymore, A machine learning approach to building domain-specific search engines, in: Proc. IJCAI-99, Stockholm, Sweden, Morgan Kaufmann, San Francisco, CA, 1999, pp. 662–667.
[88] T. Mitchell, R. Caruana, D. Freitag, J. McDermott, D. Zabowski, Experience with a learning personal assistant, Comm. ACM 37 (7) (1994) 81–91.
[89] A. Motro, I. Rakov, Estimating the quality of data in relational databases, in: Proc. 1996 Conference on Information Quality, 1996, pp. 94–106.
[90] A. Motro, Integrity = validity + completeness, ACM Trans. Database Systems 14 (4) (1989) 480–502.
[91] I. Muslea, S. Minton, C. Knoblock, A hierarchical approach to Wrapper induction, in: Proc. 3rd International Conference Autonomous Agents, 1999.
[92] N.F. Noy, M.A. Musen, Smart: Automated support for ontology merging and alignment, in: Proc. Knowledge Acquisition Workshop, Banff, Alberta, 1999.
[93] L. Palopoli, D. Saccà, G. Terracina, D. Ursino, A unified graph-based framework for deriving nominal interscheme properties, type conflicts and object cluster similarities, in: Proc. CoopIS, 1999.
[94] P. Paolini, P. Fraternali, A conceptual model and a tool environment for developing more scalable, dynamic, and customizable Web applications, in: Proc. Conference on Extending Database Technology (EDBT), 1998.
[95] Y. Papakonstantinou, S. Abiteboul, H. Garcia-Molina, Object fusion in mediator systems, in: Proc. International Conference on Very Large Data Bases (VLDB), Bombay, India, 1996.
[96] Y. Papakonstantinou, A. Gupta, H. Garcia-Molina, J. Ullman, A query translation scheme for rapid implementation of wrappers, in: Proc. International Conference on Deductive and Object-Oriented Databases (DOOD), 1995.
[97] M. Pazzani, D. Billsus, Learning and revising user profiles: The identification of interesting Web sites, Machine Learning 27 (1997) 313–331.
[98] M. Pazzani, J. Muramatsu, D. Billsus, Syskill and Webert: Identifying interesting Web sites, in: Proc. AAAI-96, Portland, OR, 1996, pp. 54–61.
[99] M. Perkowitz, R. Doorenbos, O. Etzioni, D. Weld, Learning to understand information on the Internet: An example-based approach, J. Intelligent Information Systems 8 (2) (1997) 133–153.
[100] M. Perkowitz, O. Etzioni, Category translation: Learning to understand information on the Internet, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 930–936.
[101] M. Perkowitz, O. Etzioni, Adaptive Web sites: An AI challenge, in: Proc. IJCAI-97, Nagoya, Japan, 1997.
[102] M. Perkowitz, O. Etzioni, Adaptive Web sites: Automatically synthesizing Web pages, in: Proc. AAAI-98, Madison, WI, 1998.
[103] M. Perkowitz, O. Etzioni, Adaptive Web sites: Conceptual framework and case study, in: Proc. 8th International WWW Conference, Toronto, Ont., 1999.
[104] M. Perkowitz, O. Etzioni, Towards adaptive Web sites: Conceptual framework and case study, Artificial Intelligence 118 (2000) 245–275 (this issue).
[105] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[106] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–285.
[107] A. Rajaraman, Y. Sagiv, J.D. Ullman, Answering queries using templates with binding patterns, in: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), San Jose, CA, 1995.

[108] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, in: Proc. ACM Conference on Computer-Supported Cooperative Work, 1994, pp. 175–186.
[109] D. Rumelhart, G. Hinton, R. Williams, Learning internal representations by error propagation, in: D. Rumelhart, G. McClelland, PDP Research Group (Eds.), Parallel Distributed Processing, MIT Press, Cambridge, MA, 1986.
[110] U. Shardanand, P. Maes, Social information filtering: Algorithms for automating "word of mouth", in: Conference on Human Factors in Computing Systems—CHI-95, 1995.
[111] D. Sleeman, J. Brown, Intelligent Tutoring Systems, Academic Press, London, 1982.
[112] S. Soderland, Learning to extract text-based information from the World Web, in: Proc. 3rd International Conference Knowledge Discovery and Data Mining, 1997.
[113] S. Soderland, Learning information extraction rules for semi-structured and free text, Machine Learning 34 (1–3) (1999) 233–272.
[114] A. Tomasic, L. Raschid, P. Valduriez, Scaling access to distributed heterogeneous data sources with Disco, IEEE Trans. Knowledge and Data Engineering (1998).
[115] M. Toyama, T. Nagafuji, Dynamic and structured presentation of database contents on the Web, in: Proc. Conference on Extending Database Technology (EDBT), Valencia, Spain, 1998.
[116] O.G. Tsatalos, M.H. Solomon, Y.E. Ioannidis, The GMAP: A versatile tool for physical data independence, VLDB J. 5 (2) (1996) 101–118.
[117] J.D. Ullman, Information integration using logical views, in: Proc. International Conference on Database Theory (ICDT), Delphi, Greece, 1997.
[118] T. Urhan, M.J. Franklin, L. Amsaleg, Cost based query scrambling for initial delays, in: Proc. ACM SIGMOD Conference on Management of Data, Seattle, WA, 1998, pp. 130–141.
[119] V. Vassalos, Y. Papakonstantinou, Describing and using the query capabilities of heterogeneous sources, in: Proc. International Conference on Very Large Data Bases (VLDB), Athens, Greece, 1997.
[120] D. Woelk, P. Attie, P. Cannata, G. Meredith, A. Seth, M. Sing, C. Tomlinson, Task scheduling using intertask dependencies in Carnot, in: Proc. ACM SIGMOD Conference on Management of Data, 1993, pp. 491–494.
[121] D. Woelk, B. Bohrer, N. Jacobs, K. Ong, C. Tomlinson, C. Unnikrishnan, Carnot and infosleuth: Database technology and the World Wide Web, in: Proc. ACM SIGMOD Conference on Management of Data, San Jose, CA, 1995, pp. 443–444.
[122] H.Z. Yang, P.A. Larson, Query transformation for PSJ-queries, in: Proc. International Conference on Very Large Data Bases (VLDB), Brighton, England, 1987, pp. 245–254.
[123] R. Yerneni, Y. Papakonstantinou, S. Abiteboul, H. Garcia-Molina, Fusion queries over internet databases, in: Proc. Conference on Extending Database Technology (EDBT), Valencia, Spain, 1998, pp. 57–71.
[124] I. Zukerman, D.W. Albrecht, A.E. Nicholson, Predicting users' requests on the WWW, in: Proc. 7th International Conference on User Modelling, 1999, pp. 275–284.