SIGMOD Officers, Committees, and Awardees

Chair
Raghu Ramakrishnan
Yahoo! Research
2821 Mission College
Santa Clara, CA 95054
USA
<First8CharsOfLastName AT yahoo-inc.com>

Vice-Chair
Yannis Ioannidis
University of Athens
Department of Informatics & Telecom
Panepistimioupolis, Informatics Buildings
157 84 Ilissia, Athens
HELLAS
<yannis AT di.uoa.gr>

Secretary/Treasurer
Mary Fernández
ATT Labs - Research
180 Park Ave., Bldg 103, E277
Florham Park, NJ 07932-0971
USA
<mff AT research.att.com>

SIGMOD Officers, Committees, and Awardees (continued)

SIGMOD Edgar F. Codd Innovations Award

For innovative and highly significant contributions of enduring value to the development, understanding, or use of database systems and databases. Until 2003, this award was known as the "SIGMOD Innovations Award." In 2004, SIGMOD, with the unanimous approval of ACM Council, decided to rename the award to honor Dr. E.F. (Ted) Codd (1923 - 2003) who invented the relational data model and was responsible for the significant development of the database field as a scientific discipline. Recipients of the award are the following:

Michael Stonebraker (1992)	Jim Gray (1993)	Philip Bernstein (1994)
David DeWitt (1995)	C. Mohan (1996)	David Maier (1997)
Serge Abiteboul (1998)	Hector Garcia-Molina (1999)	Rakesh Agrawal (2000)
Rudolf Bayer (2001)	Patricia Selinger (2002)	Don Chamberlin (2003)
Ronald Fagin (2004)	Michael Carey (2005)	Jeffrey D. Ullman (2006)
Jennifer Widom (2007)	Moshe Y. Vardi (2008)	

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services. Recipients of the award are the following:

Maria Zemankova (1992)	Gio Wiederhold (1995)	Yahiko Kambayashi (1995)
Jeffrey Ullman (1996)	Avi Silberschatz (1997)	Won Kim (1998)
Raghu Ramakrishnan (1999)	Michael Carey (2000)	Laura Haas (2000)
Daniel Rosenkrantz (2001)	Richard Snodgrass (2002)	Michael Ley (2003)
Surajit Chaudhuri (2004)	Hongjun Lu (2005)	Tamer Özsu (2006)
Hans-Jörg Schek (2007)	Klaus R. Dittrich (2008)	

SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to *recognize excellent research by doctoral candidates in the database field.* This award, which was previously known as the SIGMOD Doctoral Dissertation Award, was renamed in 2008 with the unanimous approval of ACM Council in honor of Dr. Jim Gray. Recipients of the award are the following:

- **2008** *Winner*: Ariel Fuxman (advisor: Renee J. Miller), University of Toronto
  Honorable Mentions: Cong Yu (advisor: H. V. Jagadish), University of Michigan;
  Nilesh Dalvi (advisor: Dan Suciu), University of Washington.
- **2006** *Winner*: Gerome Miklau, University of Washington
  Runners-up: Marcelo Arenas, Univ. of Toronto; Yanlei Diao, Univ. of California at Berkeley.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley
  Honorable Mentions: Xifeng Yan, UIUC; Martin Theobald, Saarland University

A complete listing of all SIGMOD Awards is available at: **http://www.sigmod.org/awards/**

[Last updated on January 31, 2009]

Editor's Notes

Welcome to the December 2008 issue of SIGMOD Record. We begin the issue with a welcome message from Yannis Ioannidis to SIGMOD members in general and to student members in particular.

In lieu of regular articles, this issue has a **Special Section on Managing Information Extraction**, edited by AnHai Doan, Luis Gravano, Raghu Ramakrishnan, and Shivakumar Vaithyanathan. The field, fueled by the explosion of unstructured data on the Web, has become increasingly important in recent years, receiving attention from multiple communities (AI, DB, WWW, KDD, Semantic Web, and IR). The nine papers included in this special section highlight the wide range of Information Extraction problems; they are properly presented in the introduction to the special issue. Many thanks to AnHai, Luis, Raghu, and Shivakumar for their hard work in putting together this special section, and especially to AnHai who took care of all the details with the manuscript submission system.

The **Database Principles Column** (edited by Leonid Libkin) features two articles. The first article, by Kimelfeld and Sagiv, surveys recent results on modeling and querying probabilistic XML data. The second article, by Koch, proposes a query algebra for probabilistic databases.

The **Distinguished Profiles in Data Management Column** (edited by Marianne Winslett) features an interview of Surajit Chaudhuri, who is a research area manager at Microsoft Research, an ACM Fellow, and has received the SIGMOD Contributions Award in 2004. Read Surajit's interview to find out (among many other things) how data mining led him to self-tuning databases and about life as a research manager.

We continue with an article in the **Research Centers Column** (edited by Ugur Cetintemel) about the ETH Zurich Systems Group and Enterprise Computing Center and, in particular, their research projects across three technology trends: multicore, virtualization, and cloud computing.

Next is the **Open Forum Column**, which is meant to provide a forum for members of the broader data management community to present (meta-)ideas about non-technical issues and challenges of interest to the entire community. In this issue, we have a fantastic paper by Graham Cormode, about *how NOT to review a paper*, i.e., how to avoid being an adversarial reviewer. I really enjoyed reading this article, I am sure you will too.

We continue with five articles in the **Event Reports Column** (edited by Brian Cooper). First is the *Report on the First European Conference on Software Architecture (ECSA 2007)*, written by Cuesta and Marcos. Second is the *Report on International Workshop on Privacy and Anonymity in the Information Society (PAIS 2008)*, written by Xiong, Truta, and Fotouhi. Third is the *Report on the IFIP WG5.8 International Workshop on Enterprise Interoperability (IWEI 2008)*, written by van Sinderen, Johnson, and Kutvonen. Fourth is the *Report on the First Workshop on Very Large Digital Libraries (VLDL 2008)*, written by Manghi, Pagano, and Zezula. Finally, fifth is the *Report on the First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008)*, written by Silva, Pan, Assmann, and Herinksson.

We close the issue with multiple **Calls for Papers/Participation**:

- SIGMOD 2009 Workshops: DaMoN, DBTest, IDAR, MobiDE, and WebDB,
- ER 2009: 28th International Conference on Conceptual Modeling, and
- WSDM 2009: 2nd ACM International Conference on Web Search and Data Mining

Alexandros Labrinidis
January 2009

Message to SIGMOD Members (especially Students)

In the past few months there have been several important developments at ACM SIGMOD that I would like you to know about. First of all, preparations for the 2009 SIGMOD/PODS Conference are well under way. Providence, RI, is getting ready to welcome all of us from June 29th to July 2nd and promises some memorable times. The officers responsible for the various parts of the program and the relevant committees are working hard to ensure that an exciting and diverse program is produced. Note that all six workshops of the conference have open calls for papers (http://www.sigmod09.org/delegates_workshops.shtml), so there are still several opportunities for your research results to be presented next June to the SIGMOD/PODS audience.

As always, special efforts are made to increase participation of students to the conference. In addition to a significantly reduced registration fee for all students, particular attention is given to undergraduates with an interest in databases. Following the tradition of recent years, several undergraduates will receive scholarships to defray their conference attendance costs; they will, thus, have the opportunity to personally experience the SIGMOD/PODS Conference atmosphere and come close to a wide variety of cutting-edge database research. Selected students will present their research accomplishments in a special poster session, while one of them will be chosen to receive a "best poster" award. The deadline for applying for undergraduate scholarships is April 3rd; additional details on eligibility and submission are available at http://www.sigmod09.org/calls_papers_sigmod_undergrad_poster.shtml.

In general, ACM SIGMOD tries to pay special attention to student matters independent of the main conference (http://sigmod.org/sigmod/sigmod-student/). There are significant benefits to being a student member of ACM SIGMOD, as for the same fee as online professional members, student members receive the benefits of online plus membership, which includes the annual DiSC dvd as well as any new additions to the SIGMOD Anthology dvds. In addition, the SIGMOD Dissertation Award has been established, recognizing scientific excellence in database doctoral work. Also, the *dbgrads* and *dbjobs* database systems have been established and are searchable through the SIGMOD website, facilitating matching graduating database students with related job opportunities.

Finally, an important activity that SIGMOD members can be involved in and benefits students (whether members or not) is that of mentoring. ACM has partnered with MentorNet, the leading organization promoting e-mentoring relationships between students and professionals in the areas of engineering and science. By joining, students gain invaluable career advice, encouragement and support, while professionals lend their expertise by helping to educate and inspire young professionals. Visit www.acm.org/mentornet, consider joining the program, and promote it to other students in your university that may benefit from it.

The database field has much to gain by increased participation of students to ACM SIGMOD activities. Encourage students in your environment to join and become members of the largest professional organization of database researchers!

Sincerely,
Yannis Ioannidis
ACM SIGMOD Vice-Chair
(responsible for membership issues)

Introduction to the Special Issue on Managing Information Extraction

AnHai Doan[1], Luis Gravano[2], Raghu Ramakrishnan[3], Shivakumar Vaithyanathan[4]

[1]University of Wisconsin, [2]Columbia University, [3]Yahoo! Research, [4]IBM Almaden Research

The field of information extraction (IE) focuses on extracting structured data, such as person names and organizations, from unstructured text. This field has had a long history. It attracted steady attention in the 80s and 90s, largely in the AI community.

In the past decade, however, spurred on by the explosion of unstructured data on the World-Wide Web, this attention has turned into a torrent, gathering the efforts of researchers in the AI, DB, WWW, KDD, Semantic Web and IR communities. New IE problems have been identified, new IE techniques developed, many workshops organized, tutorials presented, companies founded, academic and industrial products deployed, and open-source prototypes developed (e.g., [5, 4, 3, 1, 2]; see [5] for the latest survey). The next few years are poised to witness even more accelerated activities in these areas.

It is against this vibrant backdrop that we assemble this special issue. Our objective is threefold. First, we want to provide a glimpse into the current state of the field, highlighting in particular the wide range of IE problems. Second, we want to show that many IE problems can significantly benefit from the wealth of work on managing structured data in the database community. We believe therefore that our community can make a substantial contribution to the IE field. Finally, we hope that examining IE problems can in turn help us gain valuable insights into managing data in this Internet-centric world, a long-term goal of our community.

Keeping in mind the above goals, we end this introduction by briefly describing the nine papers assembled for the issue. These papers fall into four broad categories.

IE Management Systems

IE has typically been viewed as executing a program *once* to extract structured data from unstructured text. Over the past few years, however, there is a growing realization that in many real-world applications, instead of being a "one-shot execution," IE is often a *long-running process* that must be *managed*, ideally by an *IE management system*.

The first three papers of the issue – *"SystemT: A System for Declarative Information Extraction"* by Krishnamurthy et al., *"Information Extraction Challenges in Managing Unstructured Data"* by Doan et al., and *"Purple SOX Extraction Management System"* by Bohannon et al. – explain why this is the case, then describe three IE management systems currently under development at IBM Almaden, Wisconsin, and Yahoo! Research, respectively. Taken together, these systems provide four major capabilities. First, they provide *declarative IE languages* for developers to write IE programs. Compared to today's IE programs, which are often multiple IE "blackbox" modules stitched together using procedural code, these declarative IE programs are easier to develop, understand, debug, and maintain. They facilitate "plug and play" with IE modules, a critical need in real-world IE applications.

Second, the systems can efficiently *optimize* the above declarative IE programs, and then execute them over large data sets. Scaling up IE programs to large data sets is critical (e.g., as demonstrated clearly in the IBM Almaden paper). The optimization is cost-based, in the same spirit of cost-based optimization in RDBMSs. Third, the systems can *explain IE results* to the user: why a particular result is or is *not* produced. Such explanation capabilities are important for users to gain confidence in the system, and for debugging purposes. Finally, the systems provide a set of techniques to *solicit and incorporate user feedback* into the extraction process. Given that IE is inherently imprecise, such feedback is important for improving the quality of IE applications.

Novel IE Technologies

As IE applications proliferate, the need for new IE technologies constantly arises. The next two papers of the special issue provide examples of such needs. The paper *"Building a Query Optimizer for Information Extraction: The SQoUT Project"*, by Jain, Ipeirotis, and Gravano from Columbia University and New York University, demonstrates that different execution strategies of the same IE program often produce output with significantly varying *extraction accuracy*. Consequently, IE optimization must take into account not just runtime (as considered in IE management systems such as those described earlier), but also extraction accuracy. The paper then discusses the challenges in doing so,

and proposes a set of solutions.

The paper *"Domain Adaptation of Information Extraction Models"*, by Gupta and Sarawagi, considers the problem of adapting an IE model trained in one domain to another related domain (e.g., from extracting person names in news articles to the related domain of extracting person names in emails). Such adaptation can significantly reduce the human effort involved in constructing and training IE models, thereby facilitating the rapid spread of IE applications. The paper briefly surveys existing adaptation methods, then describes a new method currently under development at IIT Bombay.

Building Knowledge Bases with IE

In the second half of the special issue, we turn our attention from IE management systems and technologies to IE applications. An important and popular IE application is to build large knowledge bases. In this direction, the paper *"The YAGO-NAGA Approach to Knowledge Discovery"*, by Kasneci et al., describes a project to build a conveniently searchable, large-scale, highly accurate knowledge base of common facts (e.g., Sarkozy is a politician, Sarkozy is the President of France, etc.) at the Max Planck Institute for Informatics. The approach extracts such facts from Wikipedia using a combination of rule-based and learning-based extractors. A distinguishing aspect of this approach is its emphasis on achieving high extraction precision while carefully increasing the recall. Toward this end, the approach employs a variety of powerful consistency checking methods, including exploiting the concept hierarcy of WordNet.

The paper *"Webpage Understanding: Beyond Page-Level Search"*, by Nie, Wen, and Ma, describes a powerful set of learning-based techniques that can be used to extract structured data from Web pages. The paper describes how this set of techniques can be used to build a variety of knowledge-base applications at Microsoft Research Asia, such as block-based search, object-level search, and entity-relationship search.

Web-Scale, Open IE

Perhaps the "ultimate" IE application is to extract information from the entire World-Wide Web. The last two papers of the issue address this problem. The paper *"Web-Scale Extraction of Structured Data"*, by Cafarella, Madhavan, and Halevy, describes three IE systems that can be operated on the entire Web. The first system, TextRunner, does not attempt to populate a given target relation, like most current IE approaches do. Rather, it aims to discover such relations during processing. TextRunner calls this *open information extraction*. This kind of extraction is necessary if we want to extract data at the Web scale, and to automatically obtain brand-new relations as they appear over time. The second system, WebTables, extracts HTML-embedded tables, and the third system extracts Deep-Web data pages. Both of these systems, developed at Google, also operate in an open-IE fashion, without a pre-specified target schema.

The last paper, *"Using Wikipedia to Bootstrap Open Information Extraction"* by Weld, Hoffmann, and Wu, shows that all current open-IE systems adopt a *structural targeting* approach. Specifically, these systems build a general extraction engine that looks for some form of relation-independent structure on Web pages and uses this to extract tuples. A postprocessing step is then often used to normalize the extractions, determining the precise relation and entities that have been extracted.

The paper then describes Kylin, an open-IE system under development at the University of Washington, which adopts the traditional approach of *relational targeting*. Specifically, Kylin learns relation-specific extractors, then applies them at the Web scale. A key distinguishing aspect of Kylin is that it employs a set of self-supervising techniques to automatically acquire training data from Wikipedia for its large number of extractors. Another key aspect of Kylin is the use of mass collaboration to correct the extracted data. The paper compares Kylin with current open-IE systems, and discusses the topic of open IE in depth.

1. REFERENCES

[1] Eugene Agichtein and Sunita Sarawagi. Scalable information extraction and data integration, 2006. Tutorial, KDD-06, www.it.iitb.ac.in/~sunita/KDD06Tutorial.pdf.

[2] William Cohen. Information extraction. Tutorial, www.cs.cmu.edu/~wcohen/ie-survey.ppt.

[3] AnHai Doan, Raghu Ramakrishnan, and Shivakumar Vaithyanathan. Managing information extraction, 2006. Tutorial, SIGMOD-06, www.cs.wisc.edu/~anhai/papers/ie-tutorial06-final.ppt.

[4] Andrew McCallum. Information extraction: distilling structured data from unstructured text. *ACM Queue*, 3(9):48–57, 2005.

[5] Sunita Sarawagi. Information extraction. *FnT Databases*, 1(3), 2008.

SystemT: A System for Declarative Information Extraction

Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan,
Frederick Reiss, Shivakumar Vaithyanathan, and Huaiyu Zhu
IBM Almaden Research Center,
http://www.almaden.ibm.com/cs/projects/avatar/

ABSTRACT

As applications within and outside the enterprise encounter increasing volumes of unstructured data, there has been renewed interest in the area of information extraction (IE) – the discipline concerned with extracting structured information from unstructured text. Classical IE techniques developed by the NLP community were based on cascading grammars and regular expressions. However, due to the inherent limitations of grammar-based extraction, these techniques are unable to: (i) scale to large data sets, and (ii) support the expressivity requirements of complex information tasks. At the IBM Almaden Research Center, we are developing SystemT, an IE system that addresses these limitations by adopting an algebraic approach. By leveraging well-understood database concepts such as declarative queries and cost-based optimization, SystemT enables scalable execution of complex information extraction tasks. In this paper, we motivate the SystemT approach to information extraction. We describe our extraction algebra and demonstrate the effectiveness of our optimization techniques in providing orders of magnitude reduction in the running time of complex extraction tasks.

1. INTRODUCTION

Enterprise applications for compliance, business intelligence, and search are encountering increasing volumes of unstructured text in the form of emails, customer call records, and intranet/extranet Web pages. Since unstructured text, in its raw form, has limited value, there is significant interest in extracting structured information from these documents – for example, extracting entities like persons and organizations, extracting relationships amongst such entities, detecting types of customer problems, etc.

Historically, the area of information extraction (IE) was studied by the Natural Language Processing community [1, 3, 5]. Both *knowledge engineering* approaches [2] and *machine learning* based approaches [6, 8, 9] have been proposed for building *annotators* that extract structured information from text. In the knowledge engineering approach, annotators consist of carefully crafted sets of *rules* for each task. In early IE systems, text was viewed as an input sequence of symbols and rules were specified as regular expressions over the lexical features of the symbols. The Common Pattern Specification Language (CPSL) developed in the context of the TIPSTER project [2] emerged as a popular language for expressing such extraction rules.

Numerous rule-based extraction systems were built by the NLP community in the 80's and early 90's, based on the formalism of cascading grammars and the theory of finite-state automata. These systems primarily targeted two classes of extraction tasks: *entity extraction* (identifying instances of persons, organizations, locations, etc.) and *relationship/link extraction* (identifying relationships between pairs of such entities). However, emerging applications, both within the enterprise (e.g., corporate governance and Business Intelligence), and on the Web (e.g., extracting reviews and opinions from blogs and discussion forums), are creating challenges of complex information extraction from large document collections. Due to inherent fundamental limitations, traditional grammar-based approaches are unable to support these new demands.

At the IBM Almaden Research Center, we are developing SystemT, an information extraction system that applies classical database ideas to overcome the limitations of grammar-based extraction. SystemT is used in several IBM products, such as Lotus Notes and eDiscovery Analyzer, to extract complex entities from enterprise documents and emails. SystemT is currently available for download [13]. In this paper, we describe the architecture, operators, query language, and optimization techniques that form the core of SystemT.

1.1 Limitations of grammar-based approaches

To motivate SystemT, we use the following example to highlight the limitations of grammar-based extraction with regard to performance and expressive power.

Example 1 (Informal reviews from Blogs). *Consider the task of extracting, from blogs, informal reviews of live performances by music bands. Figure 1 shows the high-level organization of an annotator that captures the domain knowledge needed to accomplish this task. The two individual modules* ReviewInstance *and* ConcertInstance *identify specific snippets of text in a blog. The* ReviewInstance *module identifies snippets that indicate portions of a concert review – e.g., "show was great",*

SIGMOD Record Special Issue

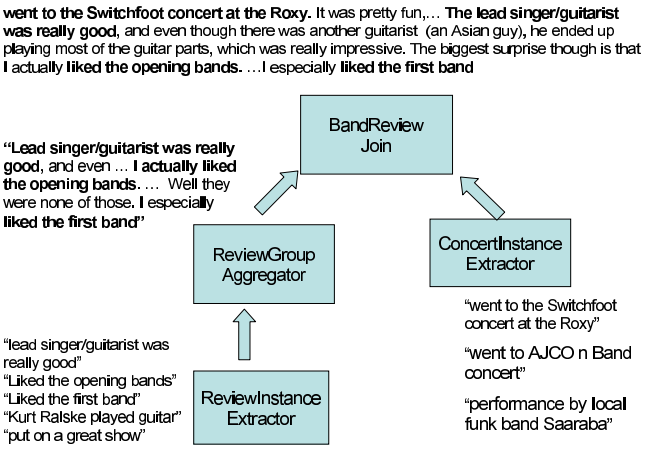


Figure 1: Extraction of informal band reviews

"liked the opening bands" and "Kurt Ralske played guitar". Similarly, the ConcertInstance *module identifies occurrences of bands or performers – e.g., "performance by the local funk band Saaraba" and "went to the Switchfoot concert at the Roxy". The output from the* ReviewInstance *module is fed into the* ReviewGroup *module to identify contiguous blocks of text containing* ReviewInstance *snippets. Finally, a* ConcertInstance *snippet is associated with one or more* ReviewGroups *to obtain individual* BandReviews.

In a traditional grammar-based IE system, the annotator described in Example 1 would be specified as a complex series of cascading grammars. For example, consider a rule in the ReviewInstance module described informally as: BandMember *followed within 30 characters by* Instrument. A translation of this specification into a cascading grammar yields:

```
ReviewInstance  ←  BandMember .{0,30} Instrument                    (R1)
BandMember      ←  RegularExpression ( [A-Z]\w+(\s+[A-Z]\w+)* )     (R2)
Instrument      ←  RegularExpression ( d1|d2|...|dn )               (R3)
```

Figure 2: Cascading grammar rules

The top-level grammar rule R_1 expresses the requirement that the pattern BandMember and Instrument appear within 30 characters of each other. Executing R_1 invokes rules R_2 and R_3, which in turn identify BandMember and Instrument instances[1]. For identifying Instrument instances, an exhaustive *dictionary* of instrument names is used. However, the actual implementation of a dictionary in a grammar-based system is via a regular expression expressed as a union of all the entries in the dictionary as shown in rule R_3.

Using a custom implementation of a CPSL-based cascading grammar system (similar to the implementation in JAPE [4]), we implemented the annotator shown in Figure 1. However, despite extensive performance tuning, the total running time *over 4.5 million blog entries was approximately seven hours* (see Figure 7) on an IBM xSeries server with two 3.6GHz Intel Xeon CPUs. A careful analysis of our annotator revealed that the primary reason for such high execution times is the cost associated with the actual evaluation of each grammar rule. As an anecdotal data point, when executing only the 3 grammar rules listed in Figure 2 over 480K blog entries, the CPU cost of regular expression evaluation dominated all other costs (IO cost of reading in documents, generating output matches, etc.), accounting for more than 90% of the overall running time. Such high CPU cost is a consequence of the fact that for a grammar rule to be evaluated over a document, potentially every character in that document must be examined. As a consequence, for complex extraction tasks, the total execution time over large document collections becomes enormous.

[1]The particular task that necessitated the extraction of such band reviews concerned the identification of new bands. This precluded the usage of any existing dictionary containing the names of current bands and required the use of a fairly complex regular expression. In the interest of readability, we have include a simpler version of the actual expression that we used for BandMember.

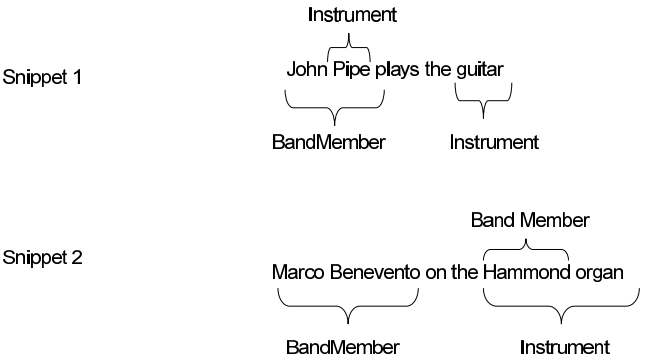


Figure 3: Overlapping Annotations

To address this scalability problem, in SystemT, we draw inspiration from the approach pioneered by relational databases. We develop an algebraic view of extraction in which rules are composed of individual extraction operators (Sec. 3) and employ cost-based optimization techniques to choose faster execution plans (Sec. 5).

Besides scalability, our algebraic approach addresses another fundamental issue in the way grammar-based systems handle *overlapping annotations*, a common phenomenon in complex extraction tasks. To illustrate, consider the following example:

EXAMPLE 2 (OVERLAPPING ANNOTATIONS).
Figure 3 shows two snippets of text drawn from real world blog entries. Snippet 1 has one instance of BandMember *and two instances of* Instrument *while Snippet 2 has one instance of Instrument and two instances of* BandMember. *Notice that both snippets have overlapping annotations. The text fragments "Pipe" in Snippet 1 and "Hammond" in Snippet 2 have both been identified as part of* BandMember *as well as* Instrument.

Annotations overlap because individual rules are run independently and these rules may make mistakes (in the sense that the author of that rule did not intend to capture a particular text snippet even though the

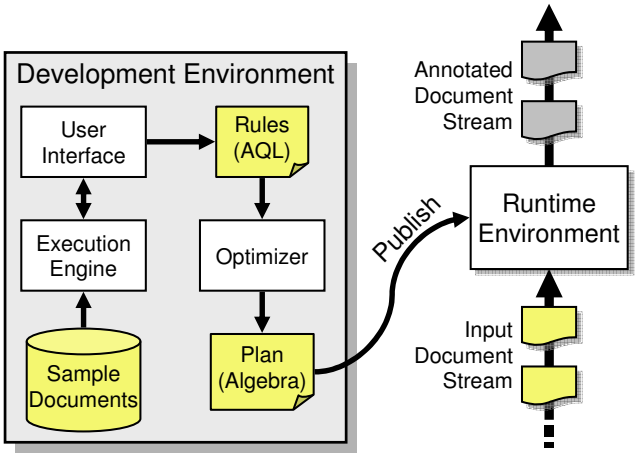


Figure 4: System Architecture

snippet turned out to be a match). Since the input to a grammar must be a sequence of tokens, overlapping annotations must necessarily be disambiguated before being fed as input to the next higher level of a cascading grammar. For example, "Pipe" must either be an Instrument or a part of BandMember, and a similar choice must be made for "Hammond". Typically, one of several ad hoc disambiguation strategies are employed. Two such popular strategies are: (a) retain the annotation that starts earlier (e.g., BandMember for *John Pipe*), and (b) a priori, impose global tie-breaking rules (e.g., BandMember dominates Instrument). Using (a), the choice in the Snippet 2 is unclear since both annotations start at the beginning of *Hammond*. Using (b) and assuming BandMember dominates, Snippet 2 will not be identified by the cascading grammar in Figure 2. On the other hand, with the choice of Instrument dominating, Snippet 1 will not be identified.

To appreciate the true effects of such ad hoc disambiguation, we ran two experiments using the rules from Figure 2 on 4.5 million blogs. When Instrument was chosen to be the dominant annotation, 6931 instances of ReviewInstance were identified. On the other hand, reversing the dominance resulted in only 5483 instances. Thus, with only three rules arranged into a 2-level cascading grammar, the number of resulting annotations varies dramatically depending on the choice of disambiguation. For extraction tasks with more rules, the situation can only become progressively worse.

On the other hand, by imposing no such requirements for input sequencing, SystemT eliminates the need for such forced disambiguation.

2. THE SYSTEMT ARCHITECTURE

Figure 4 illustrates the major components of SystemT. The *Development Environment* supports the iterative process that is involved in constructing and refining the rules for an extraction task. The rules are specified in a language called AQL (Annotation Query Language) as described in Section 4. The development environment provides facilities for compiling the rules into an algebra expression and for visualizing the results of executing the rules over a corpus of representative documents.

Once a developer is satisfied with the results that her rules produce on these documents, she can *publish* her annotator. Publishing an annotator is a two-step process. First, the AQL rules are fed into the *Optimizer*, which compiles them into an optimized algebraic expression. Then, the *Runtime Environment* instantiates the corresponding physical operators.

The Runtime Environment ("Runtime" for short) is typically embedded inside the processing pipeline of a text analytics application. The Runtime receives a continuous stream of documents, annotates each document, and outputs the annotations for further application-specific processing. The source of this document stream depends on the overall application. In Lotus Notes, for example, email messages are fed to the Runtime Environment as the user opens them in her mail client. In other applications, the document stream could come from a web crawler, an incoming message stream, or an offline document archive.

Operator class	Operators
Relational operators	σ, π, $\times$, $\cup$, $\cap$, ...
Span extraction operators	$\mathcal{E}_{re}$, $\mathcal{E}_{d}$
Span aggregation operators	Ω_o, Ω_c, β

Table 1: Operators in the SystemT algebra

3. OPERATORS AND ALGEBRA

The Optimizer and Runtime components of SystemT are based on an operator algebra that we have specifically developed for information extraction. In this section, we briefly summarize the salient aspects of our algebra and refer the reader to [10] for a more in-depth description.

3.1 Data and Execution Model

Since our algebra is designed to extract annotations from a single document at a time, we define its semantics in terms of the current document being analyzed. The current document is modeled as a string called doctext.

Our algebra operates over a simple relational data model with three data types: *span*, *tuple*, and *relation*. A span is an ordered pair $\langle begin, end \rangle$ that denotes the region of doctext from position *begin* to position *end*. A *tuple* is a finite sequence of w spans $\langle s_1, ..., s_w \rangle$; we call w the *width* of the tuple. A *relation* is a multiset of tuples with the constraint that every tuple must be of the same width. Each operator in our algebra takes zero or more relations as input and produces a single relation as output.

3.2 Algebra Operators

Based on their functionality, the set of operators in our algebra fall into the following three categories (Table 3.2):

Relational Operators

Since our data model is a minimal extension to the relational model, all of the standard relational operators (select, project, join, etc.) apply without any change. The main addition is that we use a few new selection predicates applicable only to spans [10].

Span Extraction Operators:

A span extraction operator identifies segments of text that match a particular input pattern and produces spans corresponding to each such text segment. Our algebra incorporates two kinds of span extraction operators:

- *Regular expression matcher ($\mathcal{E}_{re}$).* Given a regular expression r, $\mathcal{E}_{re}(r)$ identifies all non-overlapping matches when r is evaluated from left to right over the text represented by s. The output of $\mathcal{E}_{re}(r)$ is the set of spans corresponding to these matches. The operator takes several optional parameters such as the maximum number of tokens that a match may span over.
- *Dictionary matcher ($\mathcal{E}_d$).* Given a dictionary *dict* consisting of a set of words/phrases, the dictionary matcher $\mathcal{E}_d(dict)$ produces an output span for each occurrence of some entry in *dict* within doctext.

Span Aggregation Operators:

Span aggregation operators take in a set of input spans and produce a set of output spans by performing certain aggregate operations over their entire input. The precise details of how the aggregation is computed is different for the individual operators. Below, we describe two example aggregation operators:

- *Block.* The block operator is used to identify regions of text where input spans occur with enough regularity. For instance, in Example 1 (Figure 1), ReviewGroup is constructed by using the block operator to identify regions of text containing regular occurrences of ReviewInstance. The block operator takes in two user-defined parameters – a *separation constraint* and a *length constraint*. The separation constraint controls the regularity with which input spans must occur within the block and the length constraint specifies minimum and maximum number of such input spans that must be contained within the block.
- *Consolidate.* The consolidate operator is motivated by our observation that when multiple extraction patterns are used to identify the same concept, two different patterns often produce matches over the same or overlapping pieces of text. For instance, "liked the opening bands" and "liked the opening" are overlapping ReviewInstance occurrences identified by two different extraction patterns. To resolve such "duplicate" matches, we define several consolidation functions that define how overlapping spans need to be handled. Example consolidation functions in our algebra include

```
-- Define a dictionary of instrument names
create dictionary Instrument as ('flute', 'guitar', ... );

-- Use a regular expression to find names of band members
create view BandMember as
extract regex /[A-Z]\w+(\s+[A-Z]\w+)*/
    on 1 to 3 tokens of D.text
    as name
from Document D;

-- A single ReviewInstance rule. Finds instances of
-- BandMember followed within 30 characters by an
-- instrument name.
create view ReviewInstance as
select CombineSpans(B.name, I.inst) as instance
from
    BandMember B,
    (extract dictionary 'Instrument' on D.text as inst
     from Document D) I
where
    Follows(B.name, I.inst, 0, 30)
consolidate on CombineSpans(B.name, I.inst);
```

Figure 5: The ReviewInstance rules from Figure 2, expressed in AQL.

- *Containment consolidation*: discard annotation spans that are wholly contained within other annotation spans.
- *Overlap consolidation*: produce new spans by merging overlapping spans.

4. DECLARATIVE RULE LANGUAGE

The annotator developer's interactions with SystemT occur through the annotation rule language called AQL. AQL is a declarative language that combines the familiar syntax of SQL with the full expressive power of our text-specific operator algebra. AQL is specifically geared towards SystemT's document-at-a-time execution model. A built-in view called Document models the current document. The rule developer uses extraction primitives and text-specific predicates to build up a collection of higher-level views, eventually producing structured output annotations.

Figure 5 shows an example AQL rule from the ReviewInstance annotator. The first two statements define low-level *features* — dictionary and regular expression matches — that serve as inputs to the rule. The third statement defines the rule itself as a join with text-specific join predicates.

In addition to expressing complex low-level patterns in a declarative way, AQL also provides a compact and declarative way to define complex high-level entities. Consider the top-level "BandReview Join" rule of our annotator for informal concert reviews (see Figure 1). In English, this rule translates to:

> *Find all instances of ConcertInstance, followed within 0-30 characters by a block of 3 to 10 ReviewInstance annotations. Successive ReviewInstance annotations must be within 100 characters of each other. For each such ConcertIn-*

stance annotation, create a new output annotation that starts at the beginning of the ConcertInstance annotation and runs to the end of the last ReviewInstance annotation. Handle overlapping matches by removing any overall match that is completely contained within another match.

Such a complex pattern is nearly impossible to express in previous-generation languages, but AQL can express it quite succinctly:

```
create view BandReview as
select
    CI.instance as concert,
    CombineSpans(CI.instance, RI.instblock) as review
from
    ConcertInstance CI,
    (
        extract blocks
            with count between 3 and 10
            and separation between 0 and 100 characters
            on I.instance as instblock
        from ReviewInstance I
    ) RI
where
    Follows(CI.instance, RI.instblock, 0, 30)
consolidate on CombineSpans(CI.instance, RI.instblock)
    using 'ContainedWithin';
```

5. OPTIMIZER

Compared to traditional relational query optimization [11], the optimization problem in SystemT is distinct in several important ways. The cost of a relational database query, invariably, consists mostly of I/O and join costs. In contrast, the running time of extraction rules is dominated by the CPU cost of operations like regular expression matching and dictionary evaluation (in our experience, a typical naive execution plan will spend 90 percent or more of its time on these operators). Indeed, since SystemT's Runtime processes documents one-at-a-time, executing complex operations on each document, the time spent on I/O is generally insignificant compared to the time spent processing a document. Also, as a result of this document-at-a-time execution model, one can view the SystemT Runtime as evaluating the same operator graph over multiple separate "database instances", one per document. Thus, the goal of the SystemT optimizer is to find a plan that minimizes the *expected* cost over all these instances.

We have developed an Optimizer for SystemT that is designed specifically for the unique challenges of document-at-a-time execution and expensive extraction primitives. Note that the optimization approach adopted in SystemT, of reducing the cost of the CPU-intensive extraction operators, is complementary to the related work on optimizing extraction workflows [12]. The latter treats the individual extraction operations as black boxes and the focus is on the optimization of higher level workflows involving multiple extractors.

In the following sections, we describe the two phases in the SystemT optimization process: rule rewriting and cost-based optimization. The rewrite component of the Optimizer applies text-specific query rewrites to reduce the costs of extraction primitives. Then the cost-based component chooses join orders and methods that minimize the costs of primitive operations by taking advantage of document-at-a-time execution.

5.1 Rule Rewriting

Rule rewrites in SystemT directly improve the performance of the expensive regular expression and dictionary operators by applying transformations that do not affect semantics but almost always lead to a more efficient execution plan.

To reduce the cost of regular expressions, we have developed a technique, called *regular expression strength reduction*, that was motivated by the following observation: a major reason that regular expression evaluation is so expensive is that most regular expression engines support the full expressive power of the POSIX regular expression standard. However, for certain restricted classes of regular expressions, it is possible to build specialized engines that offer a significant performance improvement. For example, a regular expression that reduces to finding a finite set of strings can be executed far more efficiently by using a string matching engine. SystemT implements several such specialized engines. Strength reduction works by analyzing each regular expression in an AQL statement and choosing the fastest engine that can execute the expression. This technique cuts the running times of certain rules by an order of magnitude.

Another form of rule rewriting that is used in SystemT is called *shared dictionary matching*. This rewrite uses a version of the Dictionary operator that can evaluate many dictionaries at once in a single pass. Dictionary evaluation consists of three steps: identifying token boundaries, looking up each token in a hash table, and generating information about dictionary matches. By sharing the first two of these steps among many dictionaries, shared dictionary matching improves dictionary performance significantly, especially for rule sets involving a large number of small dictionaries.

5.2 Cost-Based Optimization

The rule rewrites described in the previous section work by directly reducing the overhead of expensive extraction operations. While these techniques are beneficial, SystemT achieves far more impressive performance improvements when the system can avoid executing these expensive operations at all. SystemT implements special text-specific join operators that take advantage of document boundaries and the sequential nature of text to avoid evaluating expensive extraction operations on some or all of the input text.

Figure 6 shows an example of one such operator, the *conditional evaluation join* operator, CEJoin. CEJoin, a physical implementation of the logical join operator $\bowtie$, takes advantage of SystemT's document-at-a-time execution to avoid executing extraction operators located below it in the operator graph. If the outer (left-hand) argument of the join produces no output tuples for a

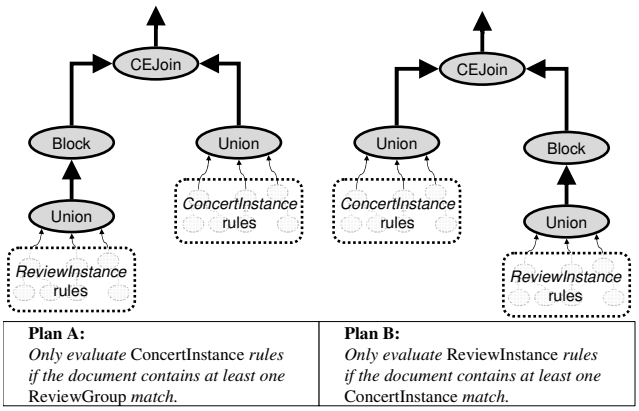


Figure 6: Two alternative plans that use Conditional Evaluation to evaluate the top-level join from the annotator in Figure 1.

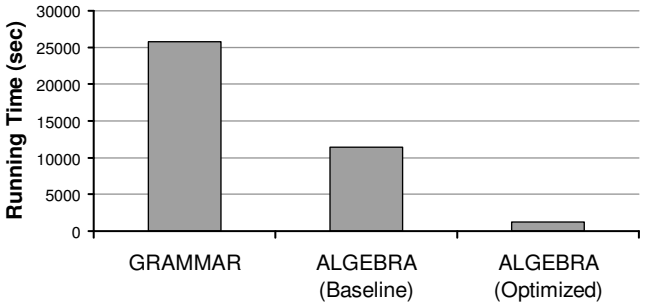


Figure 7: Results of an experiment that compares the running times of three different annotator implementations over a 4.5 million document corpus of blogs. SystemT's text-specific optimizations led to a speedup of 10x versus a naive algebraic implementation and 20x versus a hand-tuned grammar implementation.

given document, then the operator does not evaluate its inner (right-hand) argument. Depending on the expected cost of the two subtrees, as well as the probability that each subtree will produce zero output tuples on a given document, swapping the outer and inner operands of CEJoin can change overall execution times by orders of magnitude.

5.3 Experimental Results

To test the effectiveness of our optimization framework, we have performed numerous experiments over multiple data sets. In this section, we present sample results from experiments involving the annotator described in Example 1. We constructed three different implementations of this annotator:

- $GRAMMAR$: A hand-tuned grammar-based implementation.
- $ALGEBRA_{Baseline}$: A baseline algebraic plan constructed by directly translating each rule to the algebra.
- $ALGEBRA_{Optimized}$: A plan obtained by applying the text-specific optimizations described in this section.

To compare performance, we used a document corpus consisting of a collection of 4.5 million blogs (5.1GB of data) crawled from `http://www.blogspot.com`. All the experiments were run single-threaded on an IBM xSeries server with two 3.6GHz Intel Xeon CPUs.

Figure 7 shows the execution times for the three different implementations of the annotator. Simply moving from a grammar-based implementation to a naive algebraic plan led to a two-fold improvement in performance, primarily due to the fact that an algebraic implementation makes fewer passes over the text of the document. However, the application of text-specific optimizations (such as conditional evaluation) led to an additional *order of magnitude* increase in performance, for an overall improvement of close to 20 times. The net result of this experiment is that the exact same information extraction task (BandReview) which took about seven hours in an optimized grammar-based implementation now runs in under 30 minutes using SystemT.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented SystemT, a declarative information extraction system that represents a paradigm shift in the way rule-based information extraction systems are built. Using declarative queries built upon a powerful operator algebra and employing effective cost-based optimization techniques, SystemT is able to support complex extraction tasks and scale to large document collections – well beyond the capabilities of traditional grammar-based extraction systems.

We are continuing to actively develop and enhance SystemT. Some of the areas that we are actively working on include:

- *Enhanced cost-based optimization:* Modern relational engines use statistics about the distribution of values in a database table to better estimate execution cost. In a similar fashion, we are looking to enhance SystemT's optimizer with more accurate cost estimates. However, while most database query optimization work focuses on join and I/O costs, much of the cost in SystemT is highly concentrated in the extraction primitives. Thus, accurate costing of plans in SystemT requires extending the state of the art in cost modeling to deal with the unique characteristics of text. For example, changing a single character in a regular expression can change the expression's running time or number of matches by an order of magnitude. Similarly, to cost plans involving conditional evaluation, the system needs to estimate the probability that an AQL sub-expression will produce zero results on a given document.
- *Indexing techniques:* While using the SystemT Development environment, a fixed corpus of docu-

ments is used to iteratively refine and craft a rule set for an extraction task. During this development process, rule writers need to repeatedly execute their rule sets over the same document collection and execution speed becomes critical to enable effective development. With this in mind, we are working on novel index structures and algorithms to reduce the time spent in computing regular expression and dictionary matches. For instance, we are looking at indexing techniques that will allow SystemT to completely avoid executing a regular expression on a document if it can be deduced that no matches will result.

- *Language extensions:* We are continuing to work on several extension and improvements to the AQL rule language. Based on feedback from the numerous deployments of SystemT within IBM product and research groups, we are extending the language with support for features like user-defined functions and recursion. In addition, we are working towards enabling support within SystemT for part-of-speech tagging and shallow parsing.

- *Distributed computing platforms:* There is increasing interest within enterprises to use distributed computing platforms to process and analyze large volumes of unstructured and semi-structured data such as email archives, Web server logs, query logs, etc. Since information extraction is an essential component of these analyses, we are actively working to embed SystemT into the popular Hadoop platform [7]. Our goal is to enable applications to easily exploit the full power of a large cluster when performing expensive extraction tasks.

7. REFERENCES

[1] E. Agichtein and S. Sarawagi. Scalable information extraction and integration. KDD, 2006.

[2] D. E. Appelt and B. Onyshkevych. The common pattern specification language. In *TIPSTER workshop*, 1998.

[3] W. Cohen and A. McCallum. Information extraction from the World Wide Web. KDD, 2003.

[4] H. Cunningham, D. Maynard, and V. Tablan. JAPE: a java annotation patterns engine. Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, 2000.

[5] A. Doan, R. Ramakrishnan, and S. Vaithyanathan. Managing information extraction: State of the art and research directions. SIGMOD, 2006.

[6] D. Freitag. Multistrategy learning for information extraction. In *ICML*, 1998.

[7] Hadoop. http://hadoop.apache.org/.

[8] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[9] F. Peng and A. McCallum. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL*, 2004.

[10] F. Reiss, S. Raghavan, R. Krishnamurthy, H. Zhu, and S. Vaithyanathan. An algebraic approach to rule-based information extraction. In *ICDE*, 2008.

[11] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *SIGMOD*, pages 23–34, 1979.

[12] W. Shen, A. Doan, J. Naughton, and R. Ramakrishnan. Declarative information extraction using datalog with embedded extraction predicates. In *VLDB*, 2007.

[13] System Text for Information Extraction. `http://www.alphaworks.ibm.com/tech/systemt`.

Information Extraction Challenges in Managing Unstructured Data

AnHai Doan, Jeffrey F. Naughton, Raghu Ramakrishnan, Akanksha Baid, Xiaoyong Chai, Fei Chen, Ting Chen, Eric Chu, Pedro DeRose, Byron Gao, Chaitanya Gokhale, Jiansheng Huang, Warren Shen, Ba-Quy Vuong

University of Wisconsin-Madison

ABSTRACT

Over the past few years, we have been trying to build an end-to-end system at Wisconsin to manage unstructured data, using extraction, integration, and user interaction. This paper describes the key information extraction (IE) challenges that we have run into, and sketches our solutions. We discuss in particular developing a declarative IE language, optimizing for this language, generating IE provenance, incorporating user feedback into the IE process, developing a novel wiki-based user interface for feedback, best-effort IE, pushing IE into RDBMSs, and more. Our work suggests that *IE in managing unstructured data* can open up many interesting research challenges, and that these challenges can greatly benefit from the wealth of work on *managing structured data* that has been carried out by the database community.

1. INTRODUCTION

Unstructured data, such as text, Web pages, emails, blogs, and memos, is becoming increasingly pervasive. Hence, it is important that we develop solutions to manage such data. In a recent CIDR-09 paper [12] we have outlined an approach to such a solution. Specifically, we propose building *unstructured data management systems (UDMSs)*. Such systems extract structures (e.g., person names, locations) from the raw text data, integrate the structures (e.g., matching "David Smith" with "D. Smith") to build a structured database, then leverage the database to provide a host of user services (e.g., keyword search and structured querying). Such systems can also solicit user interaction to improve the extraction and integration methods, the quality of the resulting database, and the user services.

Over the past few years at Wisconsin we have been attempting to build exactly such a UDMS. Building it has raised many difficult challenges in information extraction, information integration, and user interaction. In this paper we briefly describe the key challenges in information extraction (IE) that we have faced, sketch our solutions, and discuss future directions (see [11, 10] for a discussion of non-IE challenges). Our work suggests that managing unstructured data can open up many interesting IE directions for database researchers. It further suggests that these directions can greatly benefit from the vast body of work on managing structured data that has been carried out in our community, such as work on data storage, query optimization, and concurrency control.

The work described here has been carried out in the context of the Cimple project. Cimple started out trying to build community information management systems: those that manage data for online communities, using extraction, integration, and user interaction [13]. Over time, however, it became clear that such systems can be used to manage unstructured data in many contexts beyond just online communities. Hence, Cimple now seeks to build such a general-purpose unstructured data management system, then apply it to a broad variety of applications, including community information management [13], personal information management [3], best-effort/on-the-fly data integration [17], and dataspaces [14] (see `www.cs.wisc.edu/~anhai/projects/cimple` for more detail on the Cimple project).

The rest of this paper is organized as follows. In Sections 2-4 we describe key IE challenges in developing IE programs, interacting with users during the IE process, and leveraging RDBMS technology for IE. Then in Section 5 we discuss how the above individual IE technologies can be integrated and combined with non-IE technologies to build an end-to-end UDMS. We conclude in Section 6.

2. DEVELOPING IE PROGRAMS

To extract structures from the raw data, developers often must create and then execute one or more *IE programs*. Today, developers typically create such IE programs by "stitching together" smaller IE modules (obtained externally or written by the developers themselves), using, for example, C++, Perl, or Java. While powerful, this *procedural* approach generates large IE programs that are difficult to develop, understand, debug, modify, and optimize. To address this problem, we have developed xlog, a declarative language in which to write IE programs. We now briefly describe xlog and then techniques to optimize xlog programs for both static and dynamic data.

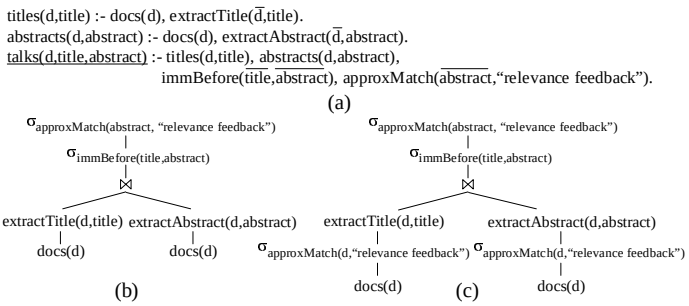


Figure 1: (a) An IE program in xlog, and (b)-(c) two possible execution plans for the program.

The xlog Declarative Language: xlog is a Datalog extension. Each xlog program consists of multiple Datalog-like *rules*, except that these rules can also contain user-defined *procedural predicates* that are pieces of procedural code (e.g., in Perl, Java).

Figure 1.a shows a tiny such xlog program with three rules, which extracts titles and abstracts of those talks whose abstracts contain "relevance feedback." Consider the first rule. Here $docs(d)$ is an extensional predicate (in the usual Datalog sense) that represents a set of text documents, whereas the term $extractTitle(\overline{d}, title)$ is a procedural predicate, i.e., a piece of code that takes as input a document d, and produces as output a set of tuples $(d, title)$, where $title$ is a talk title in document d. The first rule thus extracts all talk titles from the documents in $docs(d)$. Similarly, the second rule extracts all talk abstracts from the same documents. Finally, the third rule pairs the titles and abstracts, then retains only those where the title is immediately before the abstract and the abstract contains "relevance feedback" (allowing for misspellings and synonym matching).

The language xlog therefore allows developers to write IE programs by stitching together multiple IE "black-boxes" (e.g., $extractTitle$, $extractAbstract$, etc.) using declarative rules instead of procedural code. Such an IE program can then be converted into an execution plan and evaluated by the UDMS. For example, Figure 1.b shows a straightforward execution plan for the IE program in Figure 1.a. This plan extracts titles and abstracts, selects only those (title,abstract) pairs where the title is immediately before the abstract, then selects further only those pairs where the abstract contains "relevance feedback." In general, such a plan can contain both relational operators (e.g., $\bowtie$) and user-defined operators (e.g., $extractTitle$).

Optimizing xlog Programs: A key advantage of IE programs in xlog, compared to those in procedural languages, is that they are highly amenable to query optimization techniques. For example, consider again the execution plan in Figure 1.b. Recall that this plan retains only those (title,abstract) pairs where the abstract contains "relevance feedback." Intuitively, an abstract in a document d cannot possibly contain "relevance feedback" unless d itself also contains "relevance feedback." This suggests that we can "optimize" the above plan by discarding a document d as soon as we find out that d does not contain "relevance feedback" (a technique reminiscent of pushing down selection in relational contexts). Figure 1.c shows the resulting plan.

Of course, whether this plan is more efficient than the first plan depends on the selectivity of the selection operator $\sigma_{approxMatch(d,''relevance\ feedback'')}$ and the runtime cost of $approxMatch$. If a data set mentions "relevance feedback" frequently (as would be the case, for example, in SIGIR proceedings), then the selection selectivity will be low. Since $approxMatch$ is expensive, the second plan can end up being significantly worse than the first one. On the other hand, if a data set rarely mentions "relevance feedback" (as would likely be the case, for example, in SIGMOD proceedings), then the second plan can significantly outperform the first one. One way to address this choice of plans is to perform cost-based optimization, like in relational query optimization.

In [18] we have developed such a cost-based optimizer. Given an xlog program P, the optimizer conceptually generates an execution plan for P, employs a set of rewriting rules (such as pushing down a selection, as described above) to generate promising plan candidates, then selects the candidate with the lowest estimated cost, where the costs are estimated using a cost model (in the same spirit as relational query optimization). The work [18] describes the optimizer in detail, including techniques to efficiently search for the best candidate in the often huge candidate space.

Optimizing for Evolving Data: So far we have considered only *static* text corpora, over which we typically have to apply an xlog program only once. In practice, however, text corpora often are *dynamic*, in that documents are added, deleted, and modified. They evolve over time, and to keep extracted information up to date, we often must apply an xlog program *repeatedly*, to consecutive corpus snapshots. Consider, for example, DBLife, a structured portal for the database community that we have been developing [8, 9]. DBLife operates over a text corpus of 10,000+ URLs. Each day it recrawls these URLs to generate a 120+ MB corpus snapshot, and then applies an IE program to this snapshot to find the latest community information.

In such contexts, applying IE to each corpus snapshot *in isolation, from the scratch*, as typically done today, is very time consuming. To address this problem, in [5] we have developed a set of techniques to efficiently execute an xlog program over an evolving text corpus. The key idea underlying our solution is to recycle previous IE results, given that consecutive snapshots of a text corpus often contain much overlapping content. For example, suppose that a corpus snapshot contains the text fragment "the Cimple project will meet in room CS 105 at 3pm", from which we have extracted "CS 105" as a room number. Then when we see the above text fragment again in a new snapshot, under certain conditions (see [5]) we can immediately conclude that "CS 105" is a room number, without re-applying the IE program to the text fragment.

Overall, our work has suggested that xlog is highly

promising as an IE language. It can seamlessly combine procedural IE code fragments with declarative ones. In contrast to some other recent efforts in declarative IE languages (e.g., UIMA at `research.ibm.com/UIMA`), xlog builds on the well-founded semantics of Datalog. As such, it can naturally and rigorously handle recursion (which occurs quite commonly in IE [1, 2]). Finally, it can also leverage the wealth of execution and optimization techniques already developed for Datalog. Much work remains, however, as our current xlog version is still rudimentary. We are currently examining how to extend it to handle negation and recursion, and to incorporate information integration procedures (see Section 5), among others.

3. INTERACTING WITH USERS

Given that IE is an inherently imprecise process, user interaction is important for improving the quality of IE applications. Such interaction often can be solicited. Many IE applications (e.g., DBLife) have a sizable development team (e.g., 5-10 persons at any time). Just this team of developers *alone* can already provide a considerable amount of feedback. Even more feedback can often be solicited from the multitude of application users, in a Web 2.0 style.

The goal then is to develop techniques to enable efficient user interaction (where by "user" we mean both developers and application users). Toward this goal, we have been pursuing four research directions: explain query result provenance, incorporating user feedback, developing novel user interfaces, and developing novel interaction modes. We now briefly explain these directions.

Generating the Provenance of Query Result: Much work has addressed the problem of generating the provenance of query results [20]. But this work has focused only on *positive provenance*: it seeks to explain why an answer is produced.

In many cases, however, a user may be interested in *negative provenance*, i.e., why a certain answer is *not* produced. For example, suppose we have extracted two tables TALKS(talk-title, talk-time, room) and LOCATIONS(room,building) from text documents. Suppose the user now asks for the titles of all talks that appear at 3pm in Dayton Hall. This requires joining the above two tables on "room", then selecting those where "talk-time" is 3pm and "building" is Dayton Hall. Suppose the user expects a particular talk with title "Declarative IE" to show up in the query result, and is surprised that it does not. Then the user may want to ask the system why this talk does not show up. We call such requests "asking for the provenance of a non-answer". Such non-answer provenance is important because it can provide more confidence in the answer for the user, and can help developers debug the system.

In [15] we have developed an initial approach to providing the provenance of non-answers. In the above example, for instance, our solution can explain that no tuple with talk-title = "Declarative IE" and talk-time = 3pm has been extracted into the table TALKS, and that if such a tuple were to be extracted, then the non-answer will become an answer. Alternatively, our approach can explain that such a tuple indeed has been extracted into table TALKS, but that the tuple does not join with any tuple in table LOCATIONS, and so forth.

Incorporate User Feedback: Consider again the IE program P in Figure 1.b, which extracts titles and abstracts, pairs them, then retains only those satisfying certain conditions. Conceptually, this program can be viewed as an execution tree (in the spirit of an RDBMS execution tree), where the leaves specify input data (the table $docs(d)$ of text documents in this case), the internal nodes specify relational operations (e.g., join, select), IE operations (e.g., $extractTitle$), or procedures (e.g., $immBefore$), and the root node specifies the output (which is the table $talks(d, title, abstract)$ in this case).

Executing the above program then amounts to a bottom-up execution of the above execution tree. After the execution, a user may inspect and correct mistakes in the output table $talks(d, title, abstract)$. For example, he or she can modify a title, remove a tuple that does not correspond to a correct pair of title and abstract, or add a tuple that the IE modules fail to extract.

But the user may go even further. If during the execution we have materialized the intermediate tables (that are produced at internal nodes of the above execution tree), then the user can also correct those. For example, the user may try to correct the intermediate table $titles(d, title)$ (the output of the node associated with the IE module $extractTitle$), then propagate these corrections "up the tree". Clearly, correcting a mistake "early" can be highly beneficial as it can drastically reduce the number of incorrect tuples "further up the execution tree".

Consequently, in recent work [4] we have developed an initial solution that allows users to correct mistakes *anywhere* during the IE execution, and then propagate such corrections up the execution tree. This raises many interesting and difficult challenges, including (a) developing a way to quickly specify which parts of the data are to be corrected and in what manner, (b) redefining the semantics of the declarative program, in the presence of user corrections, (c) propagating corrections up the tree, but figuring out how to reconcile them with prior corrections, and (d) developing an efficient concurrency control solution for the common case where multiple users concurrently correct the data.

[4] addresses the above challenges in detail. Here, we briefly focus on just the first challenge: how to quickly specify which parts of the data are to be corrected and in what manner. To address this challenge, our solution allows developers to write declarative "human interaction" (HI) rules. For example, after writing the IE program in Figure 1.a, a developer may write the following HI rule:

```
extracted-titles(d,title)#spreadsheet
                :- titles(d,title), d > 200.
```

This rule states that during the program execution, a

view *extracted-titles*$(d, title)$ over table *titles*$(d, title)$ (defined by the above rule to be those tuples in the *titles*$(d, title)$ table with the doc id d exceeding 200) should be materialized, then exposed to users to edit via a spreadsheet user interface (UI). Note that the system comes pre-equipped already with a set of UIs. The developer merely needs to specify in the HI rule that which UI is to be used. The system will take care of the rest: materialize the target data part, expose it in the specified UI, incorporate user corrections, and propagate such corrections "up the execution tree."

Develop Novel User Interfaces: To correct the extracted data, today users can only use a rather limited set of UIs, such as spreadsheet interface, form interface, and GUI. To maximize user interaction with the UDMS, we believe it is important to develop a richer set of UIs, because then a user is more likely to find an UI that he or she is comfortable with, and thus is more likely to participate in the interaction.

Toward this goal, we have recently developed a wiki-based UI [7] (based on the observation that many users increasingly use wikis to collect and correct data). This UI exposes the data to be corrected in a set of wiki pages. Users examine and correct these pages, then propagate the correction to the underlying data. For example, suppose the data to be corrected is the table *extracted-titles*$(d, title)$ mentioned earlier (which is a view over table *titles*$(d, title)$). Then we can display the tuples of this table in a wiki page. Once a user has corrected, say, the first tuple of the table, we can propagate the correction to the underlying table *titles*$(d, title)$.

A distinguishing aspect of the wiki UI is that in addition to correcting *structured data* (e.g., relational tuples), users can also easily add comments, questions, explanations, etc. in *text* format. For example, after correcting the first tuple of table *extracted-titles*$(d, title)$, a user can leave a comment (right next to this tuple in the wiki page) stating why. Or another user may leave a comment questioning the correctness of the second and third tuples, such as "these two tuples seem contradictory, so at least one of them is likely to be wrong". Such text comments are then stored in the system together with the relational tables. The comments clearly can also be accommodated in traditional UIs, but not as easily or naturally as in a wiki-based UI.

Developing such a wiki-based UI turned out to raise many interesting challenges. A major challenge is how to display the structured data (e.g., relational tuples) in a wiki page. The popular current solution of using a natural-text or wiki-table format makes it easy for users to edit the data, but very hard for the system to figure out afterwards which pieces of structured data have been edited. Another major challenge is that after a user U has revised a wiki page P into a page P' and has submitted P' to the system, how does the system know which sequence of edit actions U actually intended (as it is often the case that many different edit sequences can transform P into P')?. Yet another challenge is that once the system has found the intended edit sequence, how can it efficiently propagate this sequence to the underlying data? Our recent work [7] discusses these challenges in detail and proposes initial solutions.

Develop Novel Modes of User Interaction: So far we have discussed the following mode of user interaction for UDMSs: a developer U writes an IE program P, the UDMS executes P, then U (and possibly other users) interacts with the system to improve P's execution. We believe that this mode of user interaction is not always appropriate, and hence we have been interested in exploring novel modes of user interaction.

In particular, we observe that in the above traditional mode, developer U must produce a *precise* IE program P (one that is fully "fleshed out"), before P can be executed and then exposed for user interaction. As such this mode suffers from three limitations. First, it is often difficult to execute partially specified IE programs and obtain meaningful results, thereby producing a long "debug loop". Second, it often takes a long time before we can obtain the first meaningful result (by finishing and running a precise IE program), thereby rendering this mode impractical for time-sensitive IE applications. Finally, by writing precise IE programs U may also waste a significant amount of effort, because an approximate result – one that can be produced quickly – may already be satisfactory.

To address these limitations, in [17] we have developed a novel IE mode called *best-effort IE* that interleaves IE execution with user interaction from the start. In this mode, U uses an xlog extension called alog to quickly write an initial approximate IE program P (with a possible-worlds semantics). Then U evaluates P using an approximate query processor to quickly extract an approximate result. Next, U examines the result, and further refines P if necessary, to obtain increasingly more precise results. To refine P, U can enlist a *next-effort assistant*, which suggests refinements based on the data and the current version of P.

To illustrate, suppose that given 500 Web pages, each listing a house for sale, developer U wants to find all houses whose price exceeds \$500000. Then to start, U can quickly write an initial approximate IE program P, by specifying what he or she knows about the target attributes (i.e., price in this case). Suppose U specifies only that price is numeric, and suppose further that there are only nine house pages where each page contains at least one number exceeding 500000. Then the UDMS can immediately execute P to return these nine pages as an "approximate superset" result for the initial extraction program. Since this result set is small, U may already be able to sift through and find the desired houses. Hence, U can already stop with the IE program.

Now suppose that instead of nine, there are actually 120 house pages that contain at least one number exceeding 500000. Then the system will return these 120 pages. U realizes that the IE program P is "underspecified", and hence will try to refine it further (to "narrow" the result set). To do so, U can ask the next-effort assistant to suggest what to focus on next. Suppose that this module suggests to check if price is in bold font,

and that after checking, U adds to the IE program that price is in bold font. Then the system can leverage this "refinement" to reduce the result set to only 35 houses. U now can stop, and sift through the 35 houses to find the desired ones. Alternatively, U can try to refine the IE program further, enlisting the next-effort assistant whenever appropriate.

In [17] we describe in detail the challenges of best-effort IE and proposes a set of possible solutions.

4. LEVERAGING RDBMS TECHNOLOGIES

So far we have discussed how to develop declarative IE programs and effective user interaction tools. We now turn our attention to efficiently implementing such programs.

We begin by observing that most of today's implementations perform their IE without the use of an RDBMS. A very common method, for example, is to store text data in files, write the IE program as a script, or in a recently developed declarative language (e.g., xlog [18], AQL of System-T [16], UIMA at `research.ibm.com/UIMA`), then execute this program over these text files, using the file system for all storage.

This method indeed offers a good start. But given that IE programs fundamentally extract and manipulate *structured data*, and that RDBMSs have had a 30-year history of managing structured data, a natural question arises: *Do RDBMSs offer any advantage over file systems for IE applications?* In recent work [6, 19], we have explored this question, provided an affirmative answer, and further explored the natural follow-on questions of *How can we best exploit current RDBMS technology to support IE?* and *How can current RDBMS technology be improved to better support IE?*. For space reasons, in what follows we will briefly describe only the work in [19], our latest work on the topic.

We begin in [19] by showing that executing and managing IE programs (such as those discussed so far in this paper) indeed require many capabilities offered by current RDBMSs. First, such programs often execute many relational operations (e.g., joining two large tables of extracted tuples). Second, the programs are often so complex or run over so much data that they can significantly benefit from indexing and optimization. Third, many such programs are long running, and hence crash recovery can significantly assist in making program execution more robust. Finally, many such programs and their data (i.e., input, output, intermediate results) are often edited concurrently by multiple users (as discussed earlier), raising difficult concurrency control issues.

Given the above observations, in the file-based approach the developers of IE programs can certainly develop all of the above capabilities. But such development would be highly non-trivial, and could duplicate substantial portions of the 30-year effort the DBMS community has spent developing RDBMS capabilities.

Consequently, leveraging RDBMS for IE seems like an idea that is worth exploring, and in [19] we outline a way to do so. First, we identify a set of core operations on text data that IE programs often perform. Examples of core operations include retrieving the content of a text span given its start and end positions in a document, verifying a certain property of a text span (e.g., whether it is in bold font, to support for instance best-effort IE as discussed in Section 3), and locating all substrings (of a given text span) that satisfy certain properties.

We then explore the issue of how to store text data in an RDBMS in a way that is suitable for IE, and how to build indexes over such data to speed up the core IE operations. We show that if we divide text documents into "chunks", and making this "chunking" visible to the IE operation implementations, we can exploit certain properties of these core operations to optimize data access. Furthermore, if we have sufficiently general indexing facilities, we can use indexes both to speed the retrieval of relevant text and to cache the results of function invocations, thereby avoiding repeatedly inferring useful properties of that text.

We then turn our attention to the issue of executing and optimizing IE programs within RDBMS. We show that IE programs can significantly benefit from traditional relational query optimization and show how to leverage the RDBMS query optimizer to help optimize IE programs. Finally, we show how to apply text-centric optimization (as discussed in Section 2) in conjunction with leveraging the RDBMS query optimizer. Overall, our work suggests that exploiting RDBMSs for IE is a highly promising direction in terms of possible practical impacts as well as interesting research challenges for the database community.

5. BUILDING AN END-TO-END UDMS

So far we have discussed the technologies to solve individual IE challenges. We now discuss how these technologies are being integrated to build an end-to-end prototype UDMS, an ongoing effort at Wisconsin. In what follows, our discussion will also involve information integration (II), as the UDMS often must perform both extraction and integration over the raw text data.

Figure 2 shows the architecture of our planned UDMS prototype. This architecture consists of four layers: the physical layer, the data storage layer, the processing layer, and the user layer. We now briefly discuss each layer, highlighting in particular our ongoing IE efforts and opportunities for further IE research.

The Physical Layer: This layer contains hardware that runs all the steps of the system. Given that IE and II are often very computation intensive and that many applications involve a large amount of data, the ultimate system will probably need parallel processing in the physical layer. A popular way to achieve this is to use a computer cluster (as shown in the figure) running Map-Reduce-like processes.

For now, for simplicity we plan to build the UDMS to run on a single machine. In the long run, however, it would be an important and interesting research direction to study how to run all steps of the system on a cluster of machines, perhaps using a Map-Reduce-like framework. This will require, among other tasks, de-

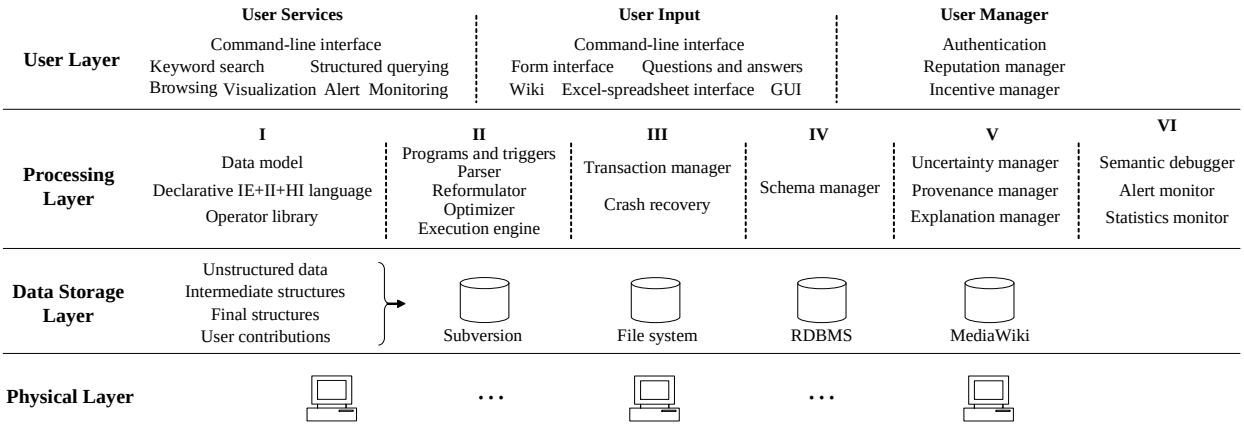


Figure 2: The architecture of our planned UDMS prototype.

composing a declarative IE/II program so that it can run efficiently and correctly over a machine cluster.

The Data Storage Layer: This layer stores all forms of data: the original data, intermediate structured data (kept around, for example, for debugging, user feedback, or optimization purposes), the final structured data, and user feedback. These different forms of data have very different characteristics, and may best be kept in different storage systems, as depicted in the figure (of course, other choices are possible, such as developing a single unifying storage system).

For example, if the original data is retrieved daily from a collection of Web sites, then the daily snapshots will overlap a lot, and hence may be best stored in a system such as Subversion, which only stores the "diff" across the snapshots, to save space. As another example, the system often executes only sequential reads and writes over intermediate structured data, in which case such data can best be kept in a file system.

For the prototype system, we will utilize a variety of storage systems, taking into account our work on storing certain parts of the IE process in RDBMSs (Section 4). Future research can then study what should be the best storage solution under which condition.

The Processing Layer: This layer is responsible for specifying and executing IE/II processes. At the heart of this layer is a data model (which is the relational data model in our current work), a declarative IE+II+HI language (over this data model), and a library of basic IE/II operators (see Part I of this layer in the figure). We envision that the above IE+II+HI declarative language will be a variant of xlog, extended with certain II features, then with HI (i.e., human interaction) rules such as those discussed in Section 3.

Developers can then use the language and operators to write declarative IE/II programs that specify how to extract and integrate the data and how users should interact with the extraction/integration process. These programs can be parsed, reformulated (to subprograms that are executable over the storage systems in the data storage layer), optimized, then executed (see Part II in the figure). Note that developers may have to write domain-specific operators, but the framework makes it easy to use such operators in the programs.

The remaining four parts, Parts III-VI in the figure, contain modules that provide support for the IE/II process. Part III handles transaction management and crash recovery. Part IV manages the schema of the derived structure. Part V handles the uncertainty that arise during the IE/II processes. It also provides the provenance for the derived structured data.

Part VI contains an interesting module called the "semantic debugger." This module learns as much as possible about the application semantics. It then monitors the data generation process, and alerts the developer if the semantics of the resulting structure are not "in sync" with the application semantics. For example, if this module has learned that the monthly temperature of a city cannot exceed 130 degrees, then it can flag an extracted temperature of 135 as suspicious. This part also contains modules to monitor the status of the entire system and alert the system manager if something appears to be wrong.

We are currently developing technical innovations for Parts I-II of the processing layer, as discussed throughout the paper. We are not working on the remaining parts of this layer, opting instead to adapt current state-of-the-art solutions.

The User Layer: This layer allows users (i.e., both lay users and developers) to exploit the data as well as provide feedback to the system. The part "User Services" contains all common data exploitation modes, such as command-line interface (for sophisticated users), keyword search, structured querying, etc. The part "User Input" contains a variety of UIs that can be used to solicit user feedback, such as command-line interface, form interface, question/answering, and wiki-based UI, as discussed in Section 3 (see the figure).

We note that modules from both parts will often be combined, so that the user can also conveniently provide feedback while querying the data, and vice versa. Finally, this layer also contains modules that authenti-

cate users, manage incentive schemes for soliciting user feedback, and manage user reputation data (e.g., for mass collaboration).

For this part, we are developing several user services based on keyword search and structured querying, as well as several UIs, as discussed in Section 3. When building the prototype system, we plan to develop other modules for this layer only on an as-needed basis.

6. CONCLUDING REMARKS

Unstructured data has now permeated numerous real-world applications, in all domains. Consequently, managing such data is now an increasingly critical task, not just to our community, but also to many others, such as the Web, AI, KDD, and SIGIR communities.

Toward solving this task, in this paper we have briefly discussed our ongoing effort at Wisconsin to develop an end-to-end solution that manages unstructured data. The discussion demonstrates that handling such data can raise many information extraction challenges, and that addressing these challenges requires building on the wealth of data management principles and solutions that have been developed in the database community. Consequently, we believe that our community is well positioned to play a major role in developing IE technologies in particular, and in managing unstructured data in general.

Acknowledgment: This work is supported by NSF grants SCI-0515491, Career IIS-0347943, an Alfred Sloan fellowship, an IBM Faculty Award, a DARPA seedling grant, and grants from Yahoo, Microsoft, and Google.

7. REFERENCES

[1] E. Agichtein, L. Gravano, J. Pavel, V. Sokolova, and A. Voskoboynik. Snowball: A prototype system for extracting relations from large text collections. In *SIGMOD*, 2001.

[2] S. Brin. Extracting patterns and relations from the world wide web. In *WebDB*, 1998.

[3] Y. Cai, X. Dong, A. Y. Halevy, J. Liu, and J. Madhavan. Personal information management with semex. In *SIGMOD*, 2005.

[4] X. Chai, B. Vuong, A. Doan, and J. F. Naughton. Efficiently incorporating user interaction into extraction and integration programs. Technical Report UW-CSE-2008, University of Wisconsin-Madison, 2008.

[5] F. Chen, A. Doan, J. Yang, and R. Ramakrishnan. Efficient information extraction over evolving text data. In *ICDE*, 2008.

[6] E. Chu, A. Baid, T. Chen, A. Doan, and J. F. Naughton. A relational approach to incrementally extracting and querying structure in unstructured data. In *VLDB*, 2007.

[7] P. DeRose, X. Chai, B. Gao, W. Shen, A. Doan, P. Bohannon, and X. Zhu. Building community wikipedias: A machine-human partnership approach. In *ICDE*, 2008.

[8] P. DeRose, W. Shen, F. Chen, A. Doan, and R. Ramakrishnan. Building structured web community portals: A top-down, compositional, and incremental approach. In *VLDB*, 2007.

[9] P. DeRose, W. Shen, F. Chen, Y. Lee, D. Burdick, A. Doan, and R. Ramakrishnan. Dblife: A community information management platform for the database research community (demo). In *CIDR*, 2007.

[10] A. Doan. Data integration research challenges in community information management systems, 2008. Keynote talk, Workshop on Information Integration Methods, Architectures, and Systems (IIMAS) at ICDE-08.

[11] A. Doan, P. Bohannon, R. Ramakrishnan, X. Chai, P. DeRose, B. Gao, and W. Shen. User-centric research challenges in community information management systems. *IEEE Data Engineering Bulletin*, 30(2):32–40, 2007.

[12] A. Doan, J. F. Naughton, A. Baid, X. Chai, F. Chen, T. Chen, E. Chu, P. DeRose, B. Gao, C. Gokhale, J. Huang, W. Shen, and B. Vuong. The case for a structured approach to managing unstructured data. In *CIDR*, 2009.

[13] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen. Community information management. *IEEE Data Engineering Bulletin*, 29(1):64–72, 2006.

[14] A. Y. Halevy, M. J. Franklin, and D. Maier. Principles of dataspace systems. In *PODS*, 2006.

[15] J. Huang, T. Chen, A. Doan, and J. F. Naughton. On the provenance of non-answers to queries over extracted data. *PVLDB*, 1(1):736–747, 2008.

[16] R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan, and H. Zhu. Systemt: A system for declarative information extraction, 2008. SIGMOD Record, Special Issue on Managing Information Extraction.

[17] W. Shen, P. DeRose, R. McCann, A. Doan, and R. Ramakrishnan. Toward best-effort information extraction. In *SIGMOD*, 2008.

[18] W. Shen, A. Doan, J. F. Naughton, and R. Ramakrishnan. Declarative information extraction using datalog with embedded extraction predicates. In *VLDB*, 2007.

[19] W. Shen, C. Gokhale, J. Patel, A. Doan, and J. F. Naughton. Relational databases for information extraction: Limitations and opportunities. Technical Report UW-CSE-2008, University of Wisconsin-Madison, 2008.

[20] W. C. Tan. Provenance in databases: Past, current, and future. *IEEE Data Eng. Bull.*, 30(4):3–12, 2007.

Purple SOX Extraction Management System

Philip Bohannon[♭], Srujana Merugu[♭], Cong Yu[♭], Vipul Agarwal[♭],
Pedro DeRose[♮], Arun Iyer[♭], Ankur Jain[♭], Vinay Kakade[♭],
Mridul Muralidharan[♭], Raghu Ramakrishnan[♭], Warren Shen[♮]
[♭]Yahoo! Research [♮]University of Wisconsin Madison
plb@yahoo-inc.com

ABSTRACT

We describe the Purple SOX (PSOX) EMS, a prototype Extraction Management System currently being built at Yahoo!. The goal of the PSOX EMS is to manage a large number of sophisticated extraction pipelines across different application domains, at the web scale and with minimum human involvement. Three key value propositions are described: *extensibility*, the ability to swap in and out extraction operators; *explainability*, the ability to track the provenance of extraction results; and *social feedback support*, the facility for gathering and reconciling multiple, potentially conflicting sources.

1. INTRODUCTION

Most documents, including electronic documents such as web pages or emails, are created for *human* consumption. Nevertheless, significant value may be added by processing these documents computationally to extract entities and/or structured data. For example, it may be useful to a financial services firm to analyze news stories for rumors of corporate takeovers, or to a consumer to determine the price at which an item of interest is being offered for sale on a particular vendor's web page. These tasks (and many others) are examples of Information Extraction (IE) (see, e.g. [11]). While techniques for IE have steadily improved, the dramatic growth of the Web strongly motivates continued innovation in this area. For example, a variety of popular Internet portals are based on structured information, some or all of which is automatically extracted from other web pages. Examples include ZoomInfo, OpenClinical, and Citeseer, which deal with professional, medical and bibliographic information respectively.

Purple SOX (Socially Oriented eXtraction) is an information extraction project at Yahoo! Research with two key goals. First, PSOX should develop extraction operators capable of working well on a semantic domain even across sites that format that information differently. Second, PSOX should develop an Extraction Management System, "PSOX EMS", that supports rapid development and large-scale deployment of on information extraction operators and pipelines, and the ability to accept and effectively manage intermittent and noisy "social" feedback on the quality of extracted data. The current prototype of the PSOX EMS is described in this paper.

The need to effectively manage information extraction has been recently discussed [7, 5]. The job of any information Extraction Management System (EMS) is to support the execution of the many component extraction *operators*—classifiers, sectioners, language analyzers, wrapper generators, etc.—as they operate on a variety of documents. In the examples mentioned above, these documents are snapshots of web pages crawled from the web.

PSOX EMS provides three key benefits: extensibility, explainability and support for social feedback. *Extensibility* means that it should be easy to add a new operator by adapting the input and output signature of an existing operator, and to quickly prototype the use of this operator in an information extraction pipeline with easy perusal of the results. The key enabler of extensibility is a declarative infrastructure for information extraction "operators." As new operators are added by humans, or trained from minimum-supervision transfer-learning techniques, a new facility becomes important: the ability to *explain* extraction results. Explanation allows the user to ask questions about extracted results, much as the questions asked by users of scientific data management systems (e.g. [4]). For example, a user looking at an extracted bibliographic record might ask how the information was produced and what other information was produced in the same way. A key application of such questions is operator debugging: by tracing back the chain of inferences that led to an extracted datum, the extraction designer can more easily and quickly localize problems to the operators or data sources at fault. Finally, PSOX EMS supports the light-weight gathering of *social feedback* on the quality of extracted results, and combining this feedback with the explanation capabilities to develop quality profiles of different opera-

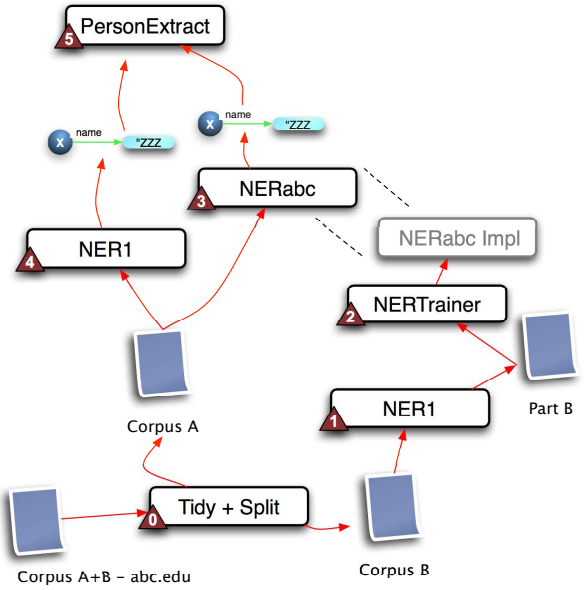


Figure 1: Academic Homepage Extraction

tors and of the feedback itself. This final capability is critical for *large scale* information extraction, as only low-supervision techniques can scale, and feedback *after* extraction will naturally be required as supervision *before* is decreased.

The rest of the paper is organized as follows. In Section 2, we introduce our running example. Section 3 describes the operator model, which supports the extensibility. The provenance model for explainability is discussed in Section 4, while the scoring model for social feedback is described in Section 5. Finally, we discuss related work and conclude in Sections 6 and 7.

2. MOTIVATING EXAMPLE

In this section, we introduce a motivating example of an information extraction pipeline.

EXAMPLE 1. *In Figure 1, an example execution of a hypothetical pipeline for extraction from academic home pages on a site* abc.edu *is shown. First, a set of downloaded web pages, "Corpus A+B" from a fictional academic website,* abc.edu*, is crawled and tidied (to create well-formed* HTML*). The corpus is then split into two sets: a small set "A" for training and set "B" for prediction. Next a "named entity recognition" operator, NER1 is used to label parts of the corpus as entities like "Person Name" or "School Name". Designed to work across different sites, NER1 is tuned for high precision but perhaps low recall. NERtrainer uses the output of NER1 "Mark A" to create the operator NERabc. The idea of NERtrainer is to adapt to features of site* abc.edu *and thus potentially allow NERabc to do better than NER1. However, since it is trained by NER1's output rather than human annotated examples, it may end up amplifying some errors in NER1 and in fact do worse - a situation typical of low-supervision techniques for information extraction. Both operators, NER1 and NERabc are then applied to the rest of the corpus (set B). Finally, a PersonExtractor operator accumulates evidence from the predictions of NER1 and NERabc to identify person entities, educational institutions, and the relevant relationships.*

Figure 2 illustrates part of a data instance in the PSOX data model that might result from the plan execution shown in Figure 1. This data instance is a graph with *score* annotations on the edges. The model is similar to RDF [9], but we use a slightly different terminology - "entity" rather than "resource" and "relationship" instead of "statement". Other features of our system that further distinguish it from most triple stores, such as support for provenance and uncertain information, are discussed below.

EXAMPLE 2. *There are four major entities (E1-E4, circles) in our example. E1 and E2 represent* system entities*: E1 represents a snapshot of a web page, as indicated by its* type *relationship, which targets the atomic entity "WebPageSnapshot". (We use round rectangles to represent atomic entities.) E2 represents the web page itself, and has a* url *that may change over time. Intuitively, E1 and E2 participate in the* snapshot-of *relationship. These entities and relationships would result from the* Tidy *operator in Figure 1.*

On the right of the figure, E3 and E4 represent semantic entities*: E3 is a school while E4 is a person with position "Professor" at that school. While all relationships (arrows) can be mapped into entities and thus be the source or target of other relationships, only two such situations are shown in Figure 2, the* name *relationship (E5) between E4 and the atomic "Lisa S. Martasyou" and the* position *relationship (E6) between E4 and the atomic "Professor". The* mentions *relationships, which connect E1 and E5/E6, illustrate the cases where a relationship itself is involved in another relationship (similar to a reified statement in RDF). Finally, every relationship has a score associated with it, but only two scores are shown:* 0.8 *on the name of the person and* 0.95 *on the name of the school). The topic of how scores are arrived at and what they mean is discussed in detail in Section 5.*

One important point is that the "type" relationship in Figure 2 is treated as a normal relationship in terms of having a score, but also represents the type of the object. Having such a "soft type" is critical in information extraction as types in practice are associated with objects by some form of classifier, and thus may be subject to error just like any other attribute.

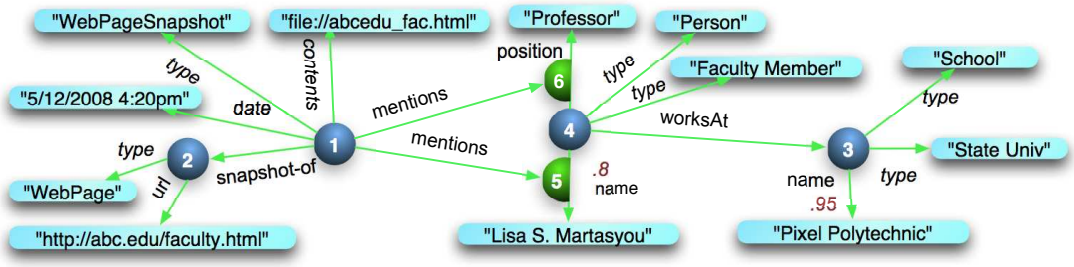


Figure 2: PSOX Data Model Example

3. OPERATORS AND EXTENSIBILITY

The extensibility of PSOX derives from a policy of *declaratively* specifying operators. In this section, we introduce the modeling of extraction operators at logical and physical levels. It is worth noting that operators are themselves represented within the PSOX data model.

Operators correspond to basic units of functionalities (e.g., page classification) and are defined at both a *logical* and *physical* level. At the logical level, an operator specifies the information it needs from the data instance and the information it produces for populating the data instance. As such, it is defined by an *operator specification*, consisting of the input it consumes and the output it produces, where the input is a collection of retrieval queries and the output is a collection of assertion queries. At the physical level, an operator is further defined by the executables, the associated arguments, the actual input and output files locations, and the necessary environment variables comprising the *operator implementation*.

A key assumption of PSOX, the *black box assumption*, is that the system should not need much visibility into *how* an operator works. At the logical level, this implies that scores from operators may not be comparable. At the physical level, this implies that we offer a language-neutral model of operators as independent executables. Compared to other extraction pipelines, this emphasizes flexibility, similar to Cimple [6] more than performance on single documents, as in UIMA [8].

Given the separation between the logical and physical levels, specific operator implementations can be easily swapped into and out of the extraction pipeline without affecting the rest of the system, as long as the logical specification is fixed. Consequently, operators written by third parties can be seamlessly leveraged inside the PSOX system.

3.1 Operator Specification

EXAMPLE 3. *We illustrate in Figure 3 a simple operator specification. Intuitively, it takes input from PSOX as specified by the input relation (i.e. retrieval query) WebPages, generates output in the format as specified by the output relation Faculty, and asserts information back into PSOX according to the output assertion (i.e., assertion query) FacultyAssertion.*

Input Specifications : The first component of an operator specification is the input specification, comprised of a set of named *retrieval queries*. While the results of a retrieval query are the *output* from the PSOX data instance, they serve as the *input* to the operator. Semantically, before the operator executes, a set of files are materialized, each containing the results of one retrieval query in the operator's input spec. Note that for performance, the input query may not always be executed: for example, if an output file of a previous operator is known to be semantically equivalent to the input relation of the next operator. We now describe retrieval queries.

A PSOX retrieval query rq is a relatively straightforward language for querying data graphs. More formally, each query a 4-tuple $(\mathsf{name}, \mathsf{V}, \mathsf{ICols}, \mathsf{CE})$, where name is the name of the query, V is the set of entity or score variables, ICols is the set of variables ($\mathsf{ICols} \in \mathsf{V}$) whose values are to be retrieved from the PSOX data instance, and CE is a constraint expression, which is recursively defined as $\mathsf{CE} = \mathsf{c}|(\mathsf{CE}' \text{ and } \mathsf{CE}'')|(\mathsf{CE}' \text{ or } \mathsf{CE}'')|$ (not CE'), where $\mathsf{c} \in \mathsf{C}$ and CE' and CE'' are themselves constraint expressions. The satisfaction of a constraint expression follows the typical logic rules. For example, a constraint expression ($ce1$ and $ce2$) is satisfied if both $ce1$ and $ce2$ are satisfied. The answer to the query is the set of tuples T. Each $t \in T$ is a set of value assignments to ICol, and there exists a set of value assignments o to variables ($\mathsf{V}' = \mathsf{V} - \mathsf{ICol}$) such that CE is satisfied given t and o.

In our example operator spec, we have the retrieval query, "WebPages", whose SELECT clause contains the three variables in ICols, and whose WHERE clause describes the CE. Intuitively, this operator asks for snapshots of web pages with URLs matching the pattern "faculty".

Output Specifications : The goal of the output specification is similar to the goal of an "ETL" script—it specifies the relationship between an operator's output (and it's input) and new data that should be added to the PSOX data instance as a set of assertions. Note that the "new data" can include new assertions on existing rela-

OPERATOR ExtractFaculty

INPUT RELATION WebPages **AS**
SELECT X, X.url, Y.contentPointer
WHERE X.type = "WebPage" and Y.type = "WebPageSnapshot"
and IsSnapShotOf(X, Y, s2) and X.url *like* "faculty"

OUTPUT RELATION Faculty
(conf, name, nameConf, pos, posConf, page)

OUTPUT ASSERTION FacultyAssertion **AS**
FROM Faculty(c, n, nc, p, pc, g)
WHERE p = "professor"
ON ENTITIES X = f(n,g)
ASSERT type(X, "Faculty Member", c)
and name(X, n, nc) and position(X, p, pc)

Figure 3: Example Operator Spec

tionships, so it may be that no new entities or attribute values are added.

An output specification contains two parts, a set of *output relation specifications* and a set of *assertion queries*. The output relation specifications simply describes the schema of a particular output file produced by the operator. Our example operator produces the relation "Faculty", which contains a list of flat tuples for extracted faculty members with attributes corresponding to: *overall confidence* about the tuple (conf), *name* of the faculty and *confidence about the name* (n, nc), *position* of the faculty and *confidence about the position* (p, pc), and where the information is extracted from (page).

Assertion queries describe how to assert the extracted information from the operator back into the PSOX data instance. They are defined in a similar way to retrieval queries, with the addition of *assertion constraints*, which are 4-tuples corresponding to new relationships being added to the data instance. In our example, the assertion query "FacultyAssertion" asserts $type$, $name$, $position$ relationships for each extracted faculty member with a position "professor".

Basic de-duplication: The variables in the ON ENTITIES clause (e.g., X) guide the creation of new entities, and the optional function following allows "single-operator" deduping. The problem is that pages may include many mentions of the same entity (e.g. bibliography pages), and it may be prohibitively expensive to create dozens or hundreds of entities only to subsequently combine them in a deduping step. In this example, we use "f(n,g)" to indicate that only one new entity X should be created for each unique (name,webpage) pair. A second mechanism allows "key functions" associated with each type. Unlike relational keys that prevent inserts, these functions ensure dedupping across extraction events. (Note that this does not replace entity resolution, and is only used when equality of entities is certain.)

3.2 Operator Implementations

IMPLEMENTATION SVMExtractor
IMPLEMENTS ExtractFaculty

RUNNING Python PROGRAM
EXTERNAL AT "/usr/bin//extractors/svm-faculty.py"
WORKING IN "/data/faculty"

INPUT FILE "pages.txt" AS WebPages
OUTPUT FILE "faculty.txt" AS Faculties

Figure 4: Example Operator Spec

Figure 4 illustrates an example operator implementation of our ExtractFaculty operator. As mentioned before, operator implementation describes the details of how the operator should be invoked. Here, it is a python program that should be executed within the directory "/data/faculty", and that it takes input file "pages.txt" (which corresponds to the input relation WebPages) and produces output file "faculty.txt" (which corresponds to the output relation Faculties).

Operator training: As mentioned before, PSOX maintains operators (both specification and implementations) as part of the data instance. As a result, an operator can assert a new operator into the data instance just like it can assert a new regular relationship. This feature allows the modeling of *training operators*, which can be considered as higher order operators that produce other operators (e.g., a classifier trainer operator can take training data and produce a classifier operator, which can then classify web pages). Often, this simply involves inserting a new operator implementation satisfying an existing operator specification.

4. PROVENANCE MODEL AND EXPLAINABILITY

In this section, we describe the way execution traces are represented in the data model.

4.1 Execution Model

Plan: Composing multiple operators together gives us a PSOX plan. Similar to the operators, plans are defined at both the logical level - as a DAG of operator specifications - and at the physical level - as a DAG of operator implementations. Currently the choice of physical operators for each spec is up to the user, but the architecture is designed to support both *plan generation* (i.e., how to produce a plan with a simple task specification like "find persons and institutions on abc.edu") and *plan optimization* (i.e., how to select the right operator for each step to maximize extraction quality and minimize extraction cost – see, e.g., [10]).

Execution : Extraction results are produced through the execution of an extraction plan, which in turn consists of executions of each individual operators in the plan. Formally, a unique *execution* x of an operator implementation, o, represents a (successful) run of physical plan step s_P where $\mathsf{oplmpl}\ (s_P) = o$. For each

execution, PSOX keeps track of the *operator* responsible for the execution, *time* of the execution, *environment* of the execution. For each execution result (i.e., an entity or relationship being asserted into the data instance), PSOX maintains the entities and relationships that lead to its generation. There are also many open research challenges. For example, *optimizing plan re-execution* (i.e., re-executing a plan that has been executed before) and *asynchronous plan execution* (i.e., separating the assertion of execution results from the plan execution to maximize throughput).

Assertion : In a data model instance I that conforms to an extraction schema, every relationship is either an axiomatic relation generated by the system or is the result of an assertion from at least one *execution event*. We define an *assertion* as a special type of relationship that has as its source an execution event and as target a relationship that is itself not an assertion. In Figure 5, the three blue arrows each indicate an assertion relationship from the operators NER1, NERabc and user Joe to the target relationship 5 (name).

The supporting evidence used by the operator in making the assertion is captured via *basis relationships*, which are defined as relationships with the evidence assertion (generated by the system or other operators) as the source and the inferred assertion as the target. For example, in Figure 5, each of the three red arrows from the system assertion associated with relationship 7 (contents) to the assertions (blue arrows) made by the different operators are examples of a basis relationship, i.e., the page contents support the operators' assertions. The notion of *basis* relationships captures the dependence between the output and input of an operator execution. However, we do not explicitly model the dependence between the output assertions of a single operator execution. The underlying assumption is that *the output assertions are conditionally independent of each other given their respective basis assertions*. This assumption is critical for tractable storage and computation, but could result in loss of information when the assumption is violated, for example in the case of collective prediction tasks.

Mention : The *mentions* relationship is another special type of relationship for supporting provenance and captures the fact that the contents of a text artifact support a relationship. Specifically, a virtual mentions relationship m from e to r_2 is supported by the PSOX query executor whenever there is an assertion a and the basis b of a, such that $\mathsf{src}(b) = e$, $label(b) =' contents''$ and $\mathsf{tgt}(a) = r_2$. Figure 2 shows example of a mentions relationship between 1 and relationship 5 (name).

4.2 Lineage

The existence of *basis* relationships enables us to readily obtain forward or backward lineage for any assertion in the data instance. Let $Asrt$ denote the set of all assertions in the data instance I and $\mathcal{P}(Asrt)$ its power set. We define the functions $Basis : Asrt \mapsto \mathcal{P}(Rel)$ and $Cons : Asrt \mapsto \mathcal{P}(Rel)$ (Consequence) as mappings from an assertion to all the supporting assertions and all the assertions that depend on it respectively, i.e., $\forall r \in Asrt$,

$$Basis(r) = \{r' | \exists (r', r, \text{"}basis\text{"}) \in Rel\}$$

$$Cons(r) = \{r' | \exists (r, r', \text{"}basis\text{"}) \in Rel\} \ \forall r \in Asrt.$$

Given the causal nature of the assertions, the data instance restricted to only "basis" relationships turns out be an directed acyclic graph. Let $Basis^*(r)$ and $Cons^*(r)$ be the transitive closure of $Basis$ and $Cons$ respectively.

In Figure 5, the subgraph restricted to red edges clearly shows the backward lineage of the operator relOp's assertion on the $worksAt$ relationship between 4 and 3, i.e., it depends on the fact that the names of person 4 and organization 3, which in turn were extracted by other NER operators are mentioned in close proximity in web page 1. Similarly, one can also identify all the assertions inferred from the contents of web page 1 by considering the forward lineage of relationship 7 (contents). Maintaining this lineage is essential for credit assignment along the pipeline, identifying erroneous operators and data, and timely reruns of operators to ensure high quality extraction. For example, if a new assertion substantially lowers the score of a relationship r as discussed in the next section, then relationships in $Cons^*(r)$ might need their scores re-evaluated, and operators that input these relationships might need to be re-run.

5. SCORING AND SOCIAL FEEDBACK

In this section, we discuss how scores on assertions and relationships are computed and updated, and outline the process of belief revision in the face of changing scores.

5.1 Assertion and Relationship Scores

Each assertion in the data instance is associated with a score, which can be interpreted as a function of the operator's estimated probability that the target relationship is true given the basis assertions. In case of axiomatic target relationships, there is no uncertainty and the assertion either supports or does not support the relationship. We choose to interpret the score as a function of probability rather than the probability value itself in order to accommodate a wide variety of execution scenarios that are common in a realistic extraction pipeline. A prime example is one where the operator implicitly assumes extra conditions (e.g., training and test data have identical distributions or restriction to a subset of all possible outcomes) so that the scores do not exactly correspond to conditional probability given the basis assertions. Another important scenario involves operators that output

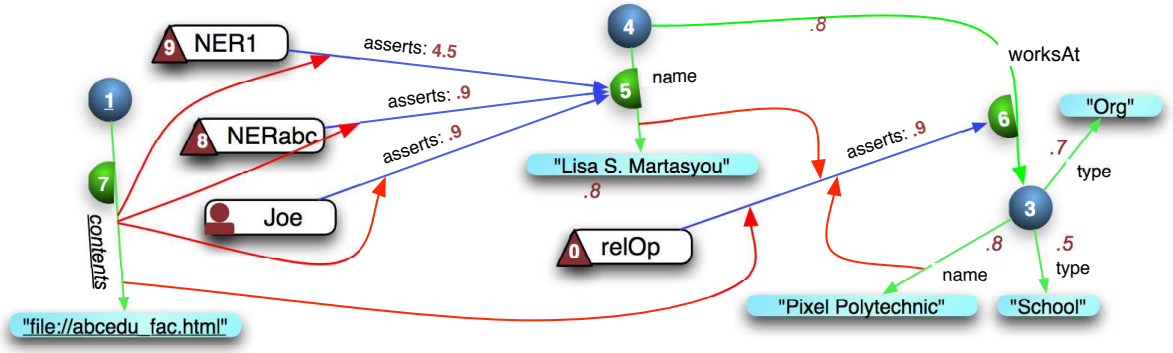


Figure 5: Provenance and Chained Inference Example

scores that cannot be readily interpreted as conditional probabilities over outcomes, e.g., SVM classifiers and margin-based predictors. Thus, the interpretation of the assertion score could vary depending on the operators as well as the nature of the target relation and the associated tasks (e.g., collection of text artifacts, classification and segmentation of text, record assembly, deduping, etc.)

Assertions by the system (or system reconciler to be exact) constitute an important subset of all assertions. In fact, each non-assertion relationship is the target of at least one system generated assertion. Furthermore, the score of a non-assertion relationship r is defined as the score of the most *recent* system assertion associated with r. For this special class of assertions, the scores can be interpreted as the probability that a relationship is true given the schema constraints and various system-specific assumptions.

Figure 5 shows this distinction between the three assertion scores on the blue arrows and the relationship (system) assertion score (0.8) on the green arrow corresponding to the relationship (names,4,"Lisa S. Martasyou") . Operator NER1 might return margins and the assertion score (4.5) cannot be viewed as a probability value, but the system adjusts for abthese variations to assign a probability of 0.8 to the target relationship.

5.2 Scoring and Social Feedback

Scoring each relationship in the data instance is a critical component of an extraction management system and requires taking into account the following key issues:

Varied Operator Behavior. In the real-world, extraction pipelines frequently involve operators with varying bias, scale and confidence levels, and often provide conflicting assertions. For instance, in figure 5, we have two automated operators (NER1, NERabc) and a user Joe providing different scores for the same target relationship (names,4,"Lisa S. Martasyou"). Hence, it is imperative to adjust for these variations in operator assertion scores by monitoring how these correlate with the "true" probability scores.

Social Feedback. Incorporating feedback from non-editorial human users enables one to rapidly obtain large amounts of training data as well as naturally scale up extraction process across various application domains. In PSOX, these human users (e.g. Joe in Figure 5) are modeled as operators with fairly general I/O spec based on their data access and editorial privileges. Compared to automated operators, human users have expertise in a large number of heterogeneous domains (e.g., text classification, segmentation, entity deduping, etc.). Further, the feedback is often incomplete and corresponds to a biased sample. Anonymity on the Internet also creates additional challenges by allowing malicious behavior and collusion among users.

Schema Constraints. Since the relationship scores are conditioned on the specified extraction schema, it is also critical to ensure that there are no violations of the schema constraints pertaining to typing, inheritance, relationship cardinality, mutual exclusion, etc. These constraints in general translate to linear equalities and inequalities over the relationship scores that determine a feasible region. For instance, in figure 5, the probability of entity 3 being a school is less than that of it being an organization, i.e., $Score(type, 3,'' school'') < Score(type, 3, ''organization'')$ since *school* is a subtype of *organization*.

Oracular Assumptions. Calibration of operator and human assertion scores requires making certain "oracular" assumptions about how they correlate to the "true" probabilities of the relationships. Such assumptions could take the form of knowledge of "true" probabilities on limited number of relationships or a functional mapping from the assertion scores to the "true" probabilities for a small set of operators.

Bayesian Solution. To address this problem, we adopt a Bayesian approach that relies on modeling the process of generating the assertion scores as a stochastic transformation of the unknown "true" probabilities of the relationships. The key idea is to use all the available operator assertions, oracular information as well as the schema constraints to estimate the most likely para-

metric model for the operator (user) behavior. The interpretation of the operator specific parameters depends heavily on the nature of assertion scores and the allowed class of transformations. For example, in Figure 5 the parameters could correspond to a linear scaling of the relationship probabilities, for example, $(9,\ 1,\ 0.9)$ for the operators NER1, NERabc and Joe respectively and the final score 0.8 assigned to (names,4,"Lisa S.") is obtained by appropriate adjustment of the assertion scores of these operators, i.e., $0.8 = (1/3) \times (4.5/9 + 0.9/1 + 0.9/0.9)$. In general, the parameters need not be specific to individual operators, but relate to observed characteristics of the operators, such as training dataset, and of the target relationships, for example, gender/profession of a person.

6. RELATED WORK

Our effort is closely related to several extraction management system projects. The first notable one is the Cimple project [6] at Wisconsin. Cimple aims to build an EMS as part of a larger community building platform, with a focus on developing declarative information extraction language and optimization techniques [10] and handling evolving data [3]. PSOX seeks to complement the capabilities of Cimple, adding further support for low-supervision extraction on a wide variety of domains, and with an emphasis on explainability and social feedback. Another closely related project is GATE [5], which provides both an architecture and a framework for natural language engineering. GATE employs a document-centric model where extracted entities are inherently associated with the documents they come from. In contrast, PSOX employs the entity-centric model, where documents and entities are both first class citizens in the model and the entities can be used and reasoned independent of the documents they are coming from. Similar to GATE, the UIMA project [8] at IBM provides an even richer framework for information extraction and its integration into various product platforms (e.g., WebSphere). While UIMA moves one step away from the document-centric model by allowing entities to be maintained independently from documents, it does not emphasize much on the after-extraction processing of those entities. In contrast, support for after-extraction entity processing like entity reconciliation is an integral part of the PSOX platform.

Scientific Data Management [1] is an active research area that is closely related to the PSOX effort regarding the plan execution model, which consists of individual operators. Efficiently and effectively tracking provenance for complex data manipulation systems (e.g., scientific data management systems) has also been receiving steady attention for quite some time [4, 2]. PSOX seeks to apply progress in this area to information extraction.

7. CONCLUSION

We have presented the fundamental architectural and modeling decisions of the Purple SOX (PSOX) Extraction Management System. We emphasized three desired characteristics of PSOX: *extensibility*, which is supported by the flexible operator model; *explainability*, which is accomplished through an extensive provenance model, and *support for social feedback*, which is achieved through an effective scoring model.

Acknowledgments We thank Michael Benedikt for several discussions on the data and scoring model. We thank AnHai Doan, Mani Abrol, Krishna Chitrapura, Keerthi Selvaraj, Minos Garorfalakis, Nilesh Dalvi, Ashwin Machanavajjhala, Arup Choudhury, Prakash Ramanan and Alok Kirpal for helpful discussions on various aspects of the system architecture.

8. REFERENCES

[1] Ilkay Altintas et al. Introduction to scientific workflow management and the Kepler system. In *SC*, 2006.

[2] A. Chapman, H. V. Jagadish, and P. Ramanan. Efficient provenance storage. In *SIGMOD*, 2008.

[3] F. Chen, A. Doan, J. Yang, and R. Ramakrishnan. Efficient information extraction over evolving text data. In *ICDE*, 2008.

[4] S. Cohen, S. Boulakia, and S. Davidson. Towards a model of provenance and user views in scientific workflows. In *DILS*, 2006.

[5] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *ACL*, 2002.

[6] P. DeRose, W. Shen, F. Chen, A. Doan, and R. Ramakrishnan. Building structured web community portals: A top-down, compositional, and incremental approach. In *VLDB*, 2007.

[7] A. Doan, R. Ramakrishnan, and S. Vaithyanathan. Managing information extraction: state of the art and research directions. In *SIGMOD*, 2006.

[8] D. Ferrucci and A. Lally. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.

[9] F. Manola and E. Miller. RDF Primer W3C Recommendation, 2004.

[10] W. Shen, A. Doan, J. Naughton, and R. Ramakrishnan. Declarative information extraction using datalog with embedded extraction predicates. In *VLDB*, 2007.

[11] J. Turmo, A. Ageno, and N. Català. Adaptive information extraction. *ACM Comput. Surv.*, 38(2):4, 2006.

Building Query Optimizers for Information Extraction: The SQoUT Project

Alpa Jain[1], Panagiotis Ipeirotis[2], Luis Gravano[1]
[1]Columbia University, [2]New York University

ABSTRACT

Text documents often embed data that is structured in nature. This structured data is increasingly exposed using *information extraction systems*, which generate structured relations from documents, introducing an opportunity to process expressive, structured queries over text databases. This paper discusses our SQoUT[1] project, which focuses on *processing structured queries over relations extracted from text databases*. We show how, in our extraction-based scenario, query processing can be decomposed into a sequence of basic steps: retrieving relevant text documents, extracting relations from the documents, and joining extracted relations for queries involving multiple relations. Each of these steps presents different alternatives and together they form a rich space of possible query execution strategies. We identify *execution efficiency* and *output quality* as the two critical properties of a query execution, and argue that an optimization approach needs to consider both properties. To this end, we take into account the user-specified requirements for execution efficiency and output quality, and choose an execution strategy for each query based on a principled, cost-based comparison of the alternative execution strategies.

1. INTRODUCTION

Real-world applications frequently rely on the information in large collections of text documents. A financial analyst is interested in tracking business transactions regarding a specific sector from news articles; a company wants to trace the general sentiment towards a recently launched product from blog articles; a biomedical research group needs to identify disease outbreaks from recent health-related reports; an intelligence agency needs to study alliances between groups of people and their past professions by analyzing web pages or email messages. In general, users in the above scenarios are interested in accessing intrinsically structured information embedded in unstructured text databases. To uncover this structured data in text documents, we can use *information extraction systems*. Information extraction systems automatically extract structured relations from text documents, enabling the effective querying of the extracted data in more powerful and expressive ways than possible over the unstructured text. In this paper, we will discuss fundamental issues in defining and efficiently *processing structured queries over relations extracted from text databases*, in the context of our SQoUT project [10, 11, 12, 13, 14]. To understand the family of structured queries on which we focus in this paper, consider the following example.

EXAMPLE 1. *Consider an archive of newspaper articles (such as the New York Times (NYT), as shown in Figure 1) along with an appropriately trained extraction system to extract a* Communications(Person, MetWith) *relation, where a tuple* $\langle \alpha, \beta \rangle$ *indicates that the person* α *communicated (e.g., via a meeting or a phone call) with person* β*. Consider now an intelligence agent who is interested in recent communications reported in the NYT news articles involving a specific person, named Nabil Shaath. So an appropriate query could be expressed using SQL as* `SELECT C.Person FROM Communications C WHERE C.MetWith = 'Nabil Shaath'`*. In principle, such a query can be answered using the data embedded in the news articles. Specifically, to address this query we can extract information on communications using the extraction system over appropriate documents retrieved from the news archive, to generate tuples such as* $\langle$*Qaftan Majali, Nabil Shaath*$\rangle$ *and then project only the necessary columns, as shown in Figure 1.*

Query processing in an extraction-based scenario can be decomposed into a sequence of basic steps: retrieving relevant text documents, extracting relations from the documents, and joining extracted relations for queries involving multiple relations. During query processing, there are generally various alternatives for each of these steps, for multiple reasons. First, information extraction systems are often far from perfect, and might output erroneous information or miss information that they should capture. As extraction systems may vary in their output quality, we may consider more than one extraction system to extract a relation. Second, information extraction is a time-consuming task, so query processing strategies may focus on minimizing the number of documents that they process. For instance, to process the query in Example 1, we can follow an "exhaustive" execution that sequentially processes and extracts in-

[1]SQoUT stands for "*Structured Queries over Unstructured Text Databases*."

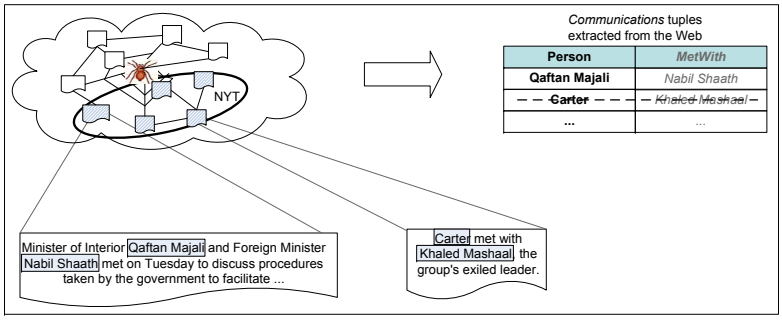


Figure 1: Processing a selection query on the relation *Communications(Person, MetWith)* from *The New York Times* news articles on the Web.

formation from every document in the New York Times archive. Alternatively, we can also follow a query-based execution that retrieves documents likely to contain the target information, or we can use a classifier-based approach that determines which New York Times articles are useful for the task at hand. Additionally, we may have multiple join algorithms to join relations extracted from text documents. By composing the various options for the query processing steps we can form a rich space of possible *query execution strategies*.

The candidate execution strategies for a query may differ substantially in their efficiency and output. The choice of extraction systems, as well as of the documents to be processed, affects not only the execution efficiency, but also the output *quality*. In the above example, the exhaustive execution can generate results that are more complete than those from a query- or classifier-based execution; however, the exhaustive execution is likely to be substantially slower than the (potentially less complete) alternatives. Thus, efficiency-related decisions meant to avoid processing useless documents may also compromise the output completeness.

As a natural consequence of this efficiency-quality trade-off, and depending on the nature of the information need, users may have varying preferences regarding the execution efficiency and output quality expected from the querying process: sometimes users may be after exhaustive, quality-oriented query answers, for which users may be willing to wait a relatively long time. Some other times, users may tolerate "quick and dirty" results, which should be returned fast. Therefore, a query execution strategy must be selected following a principled, cost-based comparison of the available candidate strategies, and taking user preferences into consideration.

We pose this problem of selecting an appropriate execution strategy for a query as a *query optimization* problem. To guide the query optimization task, we characterize query execution strategies by their execution efficiency and output quality. Selecting a desirable query execution strategy among all the candidates requires that we effectively estimate execution time and output quality for the candidate query execution strategies at query optimization time. Just as in the relational world, we need to decide which execution plan is best for a given query. Unlike in the relational world, though, we often lack direct access to detailed statistics (e.g., histograms) about the (not-yet-extracted) contents of the underlying database. So, the task of an optimizer is non-trivial, because a principled, cost-based plan selection requires addressing multiple challenges:

- *How can we account for the imprecise and incomplete nature of the information extraction output?*
- *How do we accurately predict the efficiency and output quality for an execution strategy? What database-specific information should we collect?*

The rest of the paper is structured as follows: Section 2 further motivates the need for a query optimizer by reviewing a broad family of information extraction systems, various document retrieval strategies, and join processing algorithms, which together serve as alternatives for critical operations in our query processing approach. To guide the task of query optimization, Section 3 shows how we can estimate important execution characteristics that allow us to compare query processing strategies and pick a strategy that closely meets user-specified preferences. Finally, Section 4 discusses some interesting future directions for research, and we conclude the discussion in Section 5.

2. THE NEED FOR A QUERY OPTIMIZER

For each of the important query processing steps in our extraction-based scenario, we generally have several options available. To understand this space of candidate execution strategies for a query, we discuss the choice of information extraction systems (Section 2.1), of methods to retrieve database documents to be processed during the extraction process (Section 2.2), and of algorithms to join the output from multiple extraction systems (Section 2.3). Given these alternatives, we show how we can pick a query execution strategy and discuss the underlying query optimization problem that needs to be addressed (Section 2.4).

2.1 Extracting Structured Data from Text

Information extraction automatically identifies structured data from inherently unstructured natural-language

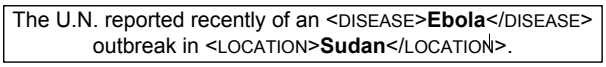


Figure 2: Extracting *DiseaseOutbreaks* from text.

text documents. In general, extraction systems are trained for a specific task. An extraction system typically begins by preprocessing a given document using lexical analysis tools (e.g., to identify nouns, verbs, and adjectives), and named-entity taggers (e.g., to identify instances of organizations, locations, and dates). An extraction system then applies *extraction patterns*, which are extraction task-specific rules, to the tagged document to identify the structured information of interest.

EXAMPLE 2. *Consider the task of extracting a* DiseaseOutbreaks(Disease, Location) *relation from a text database, where a tuple* $\langle d, \ell \rangle$ *indicates that an outbreak of disease* d *occurred in location* ℓ*. Figure 2 illustrates how an instance of the* DiseaseOutbreaks *relation can be identified, after annotating the input, using the pattern "*$\langle$*DISEASE*$\rangle$ *outbreak in* $\langle$*LOCATION*$\rangle$*."*

The extraction patterns used by an extraction system often consist of "connector" phrases or words that capture the textual context generally associated with the target information in natural language, but other models have been proposed [5]. These extraction patterns may be constructed manually, as in KnowitAll [8], or automatically, notably by using bootstrapping as in DIPRE [3], Rapier [4], Snowball [1], or in the work of Pasca et al. [17].

Information extraction is generally a time-consuming process, as it can involve a series of expensive text processing operations (e.g., part-of-speech or named-entity tagging). As a result, several approaches have been proposed recently to improve the efficiency of an extraction task [6, 9, 19, 16, 20]. We revisit one important aspect of this issue in Section 2.2.

Information extraction is also a noisy process and the extracted relations are neither perfect nor complete [7, 12, 13, 14]. An extraction system may generate erroneous tuples due to various problems (e.g., erroneous named-entity recognition or imprecise extraction patterns). Additionally, the extraction system may not extract all the valid tuples from a document (e.g., because the language in the document does not match any of the extraction patterns). To examine the quality of the output generated by an extraction system, we can measure the number of *good* and *bad* tuples in the output and, in turn, the *precision* and *recall* of the extraction output. Intuitively, precision measures the fraction of tuples extracted by the system from a text database that are *good*, while recall measures the fraction of *good* tuples that the system manages to extract from the database. There is a natural trade-off between precision and recall, and extraction systems may be trained to favor one or the other. For instance, a *precision-oriented* extraction system might be preferable for critical data in the medical domain. In contrast, a *recall-oriented* extraction system might be appropriate for an analyst interested in tracking all company mergers as reported in newspaper articles. In some scenarios, we might have more than one extraction system for a relation and the choice of extraction system for the final execution depends on the characteristics (e.g., precision, recall, or efficiency) of these systems. Furthermore, in some cases, information extraction systems may export a tuning "knob" that affects the proportion of *good* and *bad* tuples observed in the extracted relation, and the choice of the knob setting used for the final execution depends on the characteristics associated with each of these settings [13].

2.2 Retrieving Documents for Extraction

We now discuss various document retrieval methods and how they impact the execution efficiency and output quality [10, 12].

Scan: *Scan* sequentially retrieves and processes each document in the database. While this strategy guarantees that we process all the database documents, it may be unnecessarily wasteful in that many useless documents will be processed by the information extraction systems. For instance, to extract the *DiseaseOutbreaks* relation (see Example 2) from a newspaper archive, *Scan* processes all articles in the archive, including those that are likely not to produce any tuples, such as the articles in the Sports section of the newspaper.

Filtered Scan: *Filtered Scan* uses a document classifier to decide whether a database document retrieved is relevant to an extraction task. Thus, *Filtered Scan* avoids processing useless documents not relevant to the task and tends to be more efficient than *Scan*. However, this gain in efficiency might result in a loss of recall, because classifiers are not perfect and might discard documents that are indeed useful for an extraction task.

PromD: *PromD* is a query-based document retrieval technique that also focuses on *promising* documents relevant to an extraction task, while avoiding *useless* documents not relevant to the task. Specifically, *PromD* exploits the search interface of the text database and, for a given extraction task, sends appropriate queries derived using machine learning techniques [2]. For example, we may derive the query [*outbreaks AND fatality*] to retrieve documents for the *DiseaseOutbreaks* relation. Just as *Filtered Scan*, *PromD* might be substantially more efficient than *Scan*, but at the expense of a potential loss of recall.

Const: *Const* is a query-based technique based on "pushing down selections" from the user query. *Const* finds constants (if any) in a user query to retrieve only documents that contain those constants. For the query `SELECT * FROM DiseaseOutbreaks WHERE Location = Sudan`, *Const* uses [*Sudan*] as a keyword query to focus only on documents that contain this word, because documents without it could not contribute useful tuples. *Const* thus avoids processing all documents, and

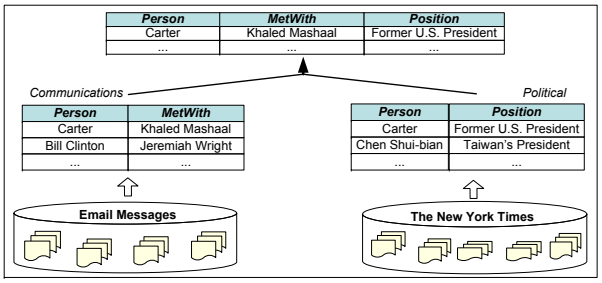


Figure 3: Joining information derived using two extraction systems.

its efficiency is determined by the selectivity of the constants in the query. Finally, we can naturally combine the *Const* queries with the *PromD* queries to generate keyword queries such as [*outbreaks AND fatality AND Sudan*] to generate results for the above query.

2.3 Joining the Extracted Data

Earlier, in Section 1, we discussed a single-relation query. By composing the output from multiple extraction systems, perhaps deployed over multiple text databases, we can also answer more complex, multiple-relation structured queries, as illustrated by the following example.

EXAMPLE 3. *Consider two text databases, a collection of email messages (EM), and the archive of The New York Times (NYT) newspaper (see Figure 3). These databases embed information that can be used to answer an intelligence analyst's query asking for all recent communications between two people, including information regarding their political role. To answer such a query, we can use information extraction systems to extract the* Communications *relation of Example 1 from EM and a* Political(Person, Position) *relation from NYT. So the analyst's query can be expressed in SQL as* `SELECT * FROM Communications C, Political P WHERE C.Person = P.Person`. *For Communications, we extract tuples such as ⟨Carter, Khaled Mashaal⟩, indicating that Carter met with Khaled Mashaal; for Political, we extract tuples such as ⟨Carter, Former U.S. President⟩, indicating that Carter has been a U.S. President. After joining all the extracted tuples, we can construct the information sought by the analyst.*

To process multiple-relation queries, we identify a variety of *join execution algorithms* that are adaptations of their relational world counterparts. Specifically, we explored three join algorithms that are naturally available in the context of text databases [14].

Independent Join: Our first algorithm joins two relations by first independently extracting their tuples and then joining them to produce the final join result. This algorithm does not exploit any knowledge from the extraction of one relation to influence the extraction of the other relation. For example, in Figure 3 we extract and join the *Communications* and *Political* relations following an independent join algorithm.

Outer/Inner Join: An alternate join execution algorithm resembles an *index nested-loops* join [18] and uses extracted tuples from the "outer" relation (e.g., tuple ⟨Carter, Khaled Mashaal⟩ for *Communications*) to build keyword queries to retrieve documents and guide the extraction of the "inner" relation (e.g., this algorithm may build query [*Carter*] to retrieve documents from which to extract *Political* tuples that will join with the *Communications* tuple).

Zig-Zag Join: Yet another alternate join execution algorithm that we consider follows a "zig-zag" execution. Specifically, this algorithm fully interleaves the extraction of the two relations in a binary join: starting with one tuple extracted for one relation (e.g., ⟨Carter, Khaled Mashaal⟩ for *Communications*), this algorithm retrieves documents—via keyword querying on the join attribute values—for extracting the second relation. In turn, the tuples from the second relation are used to build keyword queries to retrieve documents for the first relation, and the process iterates, effectively alternating the role of the outer relation of a nested loops execution over the two relations.

2.4 Selecting a Query Execution Strategy

Query processing, as discussed above, involves analyzing a rich space of candidate execution plans by seamlessly combining extraction systems along with appropriately chosen document retrieval strategies and join algorithms. A critical observation is that the choices of information extraction systems, document retrieval methods, and join algorithms influence the two main properties of a query execution discussed above, namely, execution efficiency and output quality. Thus, candidate execution strategies may differ in their efficiency and output quality, and no single query execution strategy may strictly dominate; in fact, which execution strategy among all candidates is the best option depends on user-specific requirements.

Following relational query optimization, we built a query optimizer that explores a space of candidate execution strategies for a query and selects a final execution strategy in a principled, cost-based manner. To compare candidate execution strategies, we characterize each execution strategy in terms of its execution efficiency and output quality. The execution efficiency of an execution strategy is naturally a function of the total time to run the strategy. The output quality of an execution strategy can be characterized using different metrics, such as precision and recall (see Section 2.1); alternatively, we can measure quality simply in terms of the output composition, namely, in terms of the number of *good* and *bad* tuples among the extracted tuples, or based on any other metric that uses these values. We explored such quality metrics in [12, 13, 14] and presented techniques to predict them, which we briefly review next in Section 3. Upon estimating the characteristics of the candidate execution strategies, which execution strategy is appropriate for a query depends

on user-specific needs. As an important feature of our query processing approach, we explore a variety of query paradigms to capture user-specific requirements, which we discuss later in Section 4.

3. ESTIMATING QUERY EXECUTION CHARACTERISTICS: AN OVERVIEW

The candidate execution strategies for a query may differ substantially in their execution efficiency and output quality. The choice of execution strategy for a query heavily depends on the nature of the underlying text database. For instance, text databases that are *dense* in the information to be extracted may be good candidates for using *Scan* for document retrieval; on the other hand, text databases with sparse coverage of the information to be extracted may be better candidates for using *PromD* or *Filtered Scan*. Thus, to estimate the characteristics of an execution strategy, we need to identify key database-specific factors. An important challenge is that, unlike in the relational world, where we have histograms or statistics to estimate the necessary query execution properties, here we do not know a priori the database characteristics. Naturally, processing an entire text collection with all available information extraction systems in order to gather necessary database-specific statistics is not feasible or desirable for large text databases (e.g., for the Web at large), and so we need to build effective alternative methods to gather these statistics.

Estimating the execution time of a query execution strategy is relatively simple, using statistics such as the average time to retrieve, filter, and process a document, along with the total number of documents expected to be retrieved and processed [12, 13, 14]. These statistics vary depending on both the information extraction systems of choice (e.g., an extraction system that constructs a complete syntactic parse tree of the input documents is likely to be slower than an extraction system that returns tuples based on, say, the frequency of entity co-occurrences) and the document retrieval strategies (e.g., the time to retrieve and filter a document using *Filtered Scan* is likely to be higher than the time to retrieve a document using *Scan*). So, we must consider both these factors when deriving the database-specific statistics.

Estimating the output quality of a query execution strategy, on the other hand, is a more challenging task. We designed a sampling-based estimation method that considers an execution strategy "as a whole" and identifies various database-specific statistics for the entire strategy [12]. Specifically, for each information extraction system and each document retrieval strategy, we derive statistics on the average number of tuples and of *good* tuples that the extraction system generates after processing a document retrieved using the document retrieval strategy of choice. During estimation, these statistics can be extrapolated to the number of documents that the strategy will process and we can then estimate the expected output quality of the execution strategy. To derive these salient database-specific statistics, a critical observation is that document retrieval strategies conceptually divide a text database into disjoint sets of documents: the (tuple-rich) documents that match the *PromD* queries, and the rest of the documents, which are less likely to result in useful extracted tuples. Using stratified sampling, we can derive the necessary statistics from each set [12]. Given an appropriately constructed document sample, most of the database statistics can be automatically derived. However, a key new challenge arises when gathering statistics on the average number of *good* tuples. We will discuss this challenge later in Section 4.

The notion of output quality of an execution strategy is novel to this query optimization problem. Building an effective query optimization approach requires not only an understanding of how query execution strategies—as a single unit—differ, but also an in-depth understanding of the impact of each component of an execution strategy on the overall execution. With this in mind, we performed an in-depth study of query executions in an extraction-based scenario. As argued earlier, query optimization involves making choices for three important components, namely, information extraction systems, document retrieval strategies, and join algorithms. Therefore, knowing how these components work, both stand-alone and together, is crucial to query processing design and comprehension. In particular, when studying each component of the strategy our goal is to capture the impact of a component (and the parameters thereof) on the overall execution characteristics. Towards this goal, we built analytical models for each component that provide a concise view of its behavior. Specifically, we derived models for the output quality as a function of the information extraction systems and their parameters (e.g., knob settings), the document retrieval strategies, as well as the join algorithms. By understanding the impact on the execution time and output quality of a query execution strategy's components, we can also understand—and thus, predict—the characteristics of the overall query execution strategy [13, 14].

To summarize, given a structured query we explore the space of candidate execution strategies as discussed in Section 2. We estimate the key characteristics of each of the candidate strategies using the techniques reviewed above. Finally, to guide the choice of appropriate query execution strategies, we consider the user-specified preferences for execution efficiency and output quality.

4. DISCUSSION

Traditional relational optimizers focus on minimizing the time to execute a given query. In contrast, a query optimizer for processing structured queries over text databases must consider both execution efficiency *and* output quality. Thus, designing key components for a query optimizer in the context of text databases introduces interesting research directions, some of which we discuss next.

4.1 Automated Quality Evaluation

At the heart of the estimation processes discussed above lies the critical task of determining whether an observed tuple is correct or not. This task is important for two purposes: (a) to evaluate the output of the extraction systems and, in turn, the performance of the query optimizer during evaluation, and (b) to derive database-specific statistics used during query processing. To carry out this task, manually inspecting each tuple is, of course, tedious and prohibitively time-consuming.

As one possible automated verification approach, we can resort to using external *gold sets* to identify all the good tuples among a set of extracted tuples; tuples that are present in the gold set are good tuples, and the rest are bad tuples. However, in many real-life applications the extraction system needs to extract previously unseen tuples that simply do not appear in any gold set.

Earlier work has looked into the problem of verifying a tuple by gathering evidence from a given text database. In general, these approaches assign a confidence score to a tuple based on this evidence [1, 7, 17] and, using an appropriate threshold for the confidence score, we can determine whether a tuple is good or bad. In our earlier work [12, 13, 14], we also resorted to a partially automated verification approach. Specifically, we first manually define a small number of "high-precision" natural-language patterns for each relation. Then, to decide whether an extracted tuple is good or not, we instantiate the high-precision patterns for the relation with the attribute values for the tuple, and search for instances of the instantiated patterns using the database's search interface. We consider the presence of an instantiated pattern in a database as strong evidence of tuple correctness.

EXAMPLE 4. *Consider the task of extracting a* Headquarters(Company, Location) *relation from a text database, where a tuple* $\langle c, \ell \rangle$ *indicates that* c *is a company whose headquarters are located in* ℓ*. To verify tuples of the* Headquarters *relation, one possible template we can define is "*$\langle$*LOCATION*$\rangle$*-based* $\langle$*ORGANIZATION*$\rangle$*. Thus, the tuple* $\langle$*Microsoft Corp., Redmond*$\rangle$*, results in an instantiated pattern "Redmond-based Microsoft Corp.," which we issue as a query to the database's search interface.*

While this template-based verification task allows for some automation in the tuple-verification process, generating *enough* high-precision patterns for each relation is tedious and may be difficult to achieve. And using a restricted set of patterns is undesirable as it may result in few or no tuples being marked as correct and, in turn, may introduce some bias in the derived output quality.

Automated verification at a large scale is therefore a hard problem. Ideally, we would like to automatically verify a tuple based on the evidence gathered from the underlying database in a reliable manner. Approaches that leverage small-scale manual annotation (e.g., using on-demand services like Amazon's Mechanical Turk[2]) to improve scalable automatic verification techniques [21] are promising directions for improving the current state of the art.

[2] `http://www.mturk.com`

4.2 Query Formulation: Capturing User Needs

As argued earlier, candidate execution strategies for a given query may differ substantially in their execution efficiency and their output quality. Furthermore, different users may have different preferences regarding the desired execution efficiency and output quality of a query execution. Based on this observation, we can design query paradigms that reflect whether users are after output quality, efficiency, or an appropriate balance between these two query-execution characteristics. We briefly review some query processing paradigms that we have considered.

Threshold-based Model: Sometimes users may be after some minimum output quality, for which they desire a query execution that is as fast as possible. In [13, 14], we presented a query paradigm where users specify their preferences in terms of the minimum number of good tuples desired, as well as the maximum number of bad tuples that they are willing to tolerate. These thresholds then guide the query optimization process, which identifies query execution strategies that meet these user requirements and then picks the fastest among these candidate strategies. This relatively "low-level" query paradigm can be used as a building block for higher-level paradigms (e.g., users might request some minimum precision or recall, under some constraint on execution time, or some minimum value for a combination of precision and recall).

Efficiency-Quality "Mixture" Model: In [12], we presented a query paradigm where users can specify the desired balance between execution efficiency and output quality. Such "high-level" user preferences can allow users to explore a database without having to know the exact output quality thresholds as is required in the *Threshold-based Model* discussed above.

Score-based Model: Sometimes extraction system report each extracted tuple together with an extraction "score" that reflects the extraction system's confidence in the correctness of the extracted tuple. Given such score-based setting, users may be after a relatively small number of high-ranking tuples, where the tuple ranking is determined by the tuple confidence scores as exported by the extraction systems; furthermore, these tuples should be produced as fast as possible. In [15], we presented a query paradigm where users can specify the desired number of high-ranking tuples. This score-based model allows users to quickly receive a few desirable tuples, while avoiding uninteresting tuples that have low confidence scores.

The alternative query paradigms discussed above have relative strengths and weaknesses, particularly in terms of how natural and useful they are from a user's perspective. An important direction for future work is to conduct user studies to understand the relative merits of these paradigms, and also to help define additional

ones that have not yet been explored.

5. CONCLUSION

In this paper, we presented an overview of our ongoing SQoUT project, with the general goal of processing structured queries over relations extracted from text databases. We identified and tackled a variety of problems that occur in our query optimization setting for information extraction. Specifically, we identified a rich space of query execution strategies and critical query execution characteristics, namely, efficiency and output quality. We also discussed methods to estimate these query execution characteristics, to build, in turn, a robust query optimizer for our text-based scenario. Our query optimizer takes into account user-specific requirements for execution efficiency and output quality, and chooses an execution strategy for each query in a principled, cost-based manner.

6. ACKNOWLEDGMENTS

This work has been performed in collaboration with AnHai Doan. This material is based upon work supported by a generous gift from Microsoft Research, as well as by the National Science Foundation under Grants No. IIS-0811038 and IIS-0643846. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Microsoft Research or the National Science Foundation.

7. REFERENCES

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *DL*, 2000.

[2] E. Agichtein and L. Gravano. Querying text databases for efficient information extraction. In *ICDE*, 2003.

[3] S. Brin. Extracting patterns and relations from the world wide web. In *WebDB*, 1998.

[4] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In *IAAI*, 1999.

[5] W. Cohen and A. McCallum. Information extraction from the World Wide Web (tutorial). In *KDD*, 2003.

[6] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: An architecture for development of robust HLT applications. In *ACL*, 2002.

[7] D. Downey, O. Etzioni, and S. Soderland. A probabilistic model of redundancy in information extraction. In *IJCAI*, 2005.

[8] O. Etzioni, M. J. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in KnowItAll (preliminary results). In *WWW*, 2004.

[9] D. Ferrucci and A. Lally. UIMA: An architectural approach to unstructured information processing in the corporate research environment. In *Natural Language Engineering*, 2004.

[10] P. G. Ipeirotis, E. Agichtein, P. Jain, and L. Gravano. To search or to crawl? Towards a query optimizer for text-centric tasks. In *SIGMOD*, 2006.

[11] P. G. Ipeirotis, E. Agichtein, P. Jain, and L. Gravano. Towards a query optimizer for text-centric tasks. *ACM Transactions on Database Systems*, 32(4), Dec. 2007.

[12] A. Jain, A. Doan, and L. Gravano. Optimizing SQL queries over text databases. In *ICDE*, 2008.

[13] A. Jain and P. G. Ipeirotis. A quality-aware optimizer for information extraction. *ACM Transactions on Database Systems*, 2009. To appear.

[14] A. Jain, P. G. Ipeirotis, A. Doan, and L. Gravano. Join optimization of information extraction output: Quality matters! In *ICDE*, 2009. To appear.

[15] A. Jain and D. Srivastava. Exploring a few good tuples from text databases. In *ICDE*, 2009. To appear.

[16] I. Mansuri and S. Sarawagi. A system for integrating unstructured data into relational databases. In *ICDE*, 2006.

[17] M. Paşca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and searching the world wide web of facts - step one: The one-million fact extraction challenge. In *WWW*, 2007.

[18] R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw-Hill, 2002.

[19] F. Reiss, S. Raghavan, R. Krishnamurthy, H. Zhu, and S. Vaithyanathan. An algebraic approach to rule-based information extraction. In *ICDE*, 2008.

[20] W. Shen, A. Doan, J. Naughton, and R. Ramakrishnan. Declarative information extraction using Datalog with embedded extraction predicates. In *VLDB*, 2007.

[21] V. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *KDD*, 2008.

Domain Adaptation of Information Extraction Models

Rahul Gupta
IIT Bombay
grahul@cse.iitb.ac.in

Sunita Sarawagi
IIT Bombay
sunita@iitb.ac.in

ABSTRACT

Domain adaptation refers to the process of adapting an extraction model trained in one domain to another related domain with only unlabeled data. We present a brief survey of existing methods of retraining models to best exploit labeled data from a related domain. These approaches that involve expensive model retraining are not practical when a large number of new domains have to be handled in an operational setting. We describe our approach for adapting record extraction models that exploits the regularity within a domain to jointly label records without retraining any model.

1. INTRODUCTION

The construction of models employed by information extractors is an expensive process requiring tedious effort either in collecting labeled data or hand coding the models. In many cases, it is possible to substantially reduce this effort if a model from a related domain is available. For example, we might find a model for extracting people names from news articles, while we are actually interested in extracting people name mentions in emails. Or, we find a model to identify the polarity of sentiment about home appliances and we are interested in determining the sentiment about audio equipment. In both these cases, although our target domain is related to the original domain, it has a systematic difference so that a blind application of the model is not expected to provide high accuracy. Even within the same domain, an extraction model often needs to be applied on unstructured sources that within themselves display a regularity not foreseeable at the time of model creation. For example, a model trained to extract fields of citation records can be improved significantly by exploiting the regularity of multiple strings from the same web page.

Such forms of domain adaptation will be essential in any large scale information extraction system involving multiple kinds of extraction tasks on evolving and open-ended sources. While domain adaptation is applicable both for manually-coded and machine learning models, in this article we concentrate on machine learning models. We will present an overview of the main techniques that have emerged recently from machine learning and natural language processing communities, and then present an overview of our research in the area.

2. BASICS OF LEARNING-BASED IE MODELS

Many IE tasks, including entity extraction, relationship extraction, and sentiment extraction, are formulated as feature-based prediction models. These models predict a label $\mathbf{y}$ from a space $\mathcal{Y}$ given an input $\mathbf{x}$ based on a feature vector $\mathbf{f}(\mathbf{x}, \mathbf{y}) \in \mathcal{R}^K$ that maps any $(\mathbf{x}, \mathbf{y})$ pair to a vector of K reals. The feature vector is defined by the user and provides a convenient abstraction for capturing many varied kinds of clues known to aid the extraction task. The model associates a weight vector $\mathbf{w}$ corresponding to the feature vector $\mathbf{f}$ and the predicted label is simply the $\mathbf{y}$ with the highest value of $\mathbf{w} \cdot \mathbf{f}(\mathbf{x}, \mathbf{y})$. We briefly illustrate how this works for different kind of extraction tasks. More details can be found in this survey [11].

In sentiment extraction, the space $\mathcal{Y}$ of possible predictions is positive or negative and an entry in the feature vector is typically the counts of occurrences of a particular word or frequent word bigram in the input document $\mathbf{x}$.

In relationship extraction, the input $\mathbf{x}$ is a sentence and two marked entity strings in it. The task is to predict the kind of relationship that exists between the entity pair. The space $\mathcal{Y}$ consists of possible relationship types as defined by the user and a special label "other" indicating none of the above. The feature vector entries capture various syntactic and semantic relationships between the strings, such as the bag of words between the two mentions, the parts-of-speech information of words adjacent to the mention, and so on.

In entity extraction, the task is to label words in a sentence with one of a fixed set of entity types. The prediction $\mathbf{y}$ is therefore a vector of length n for an input sentence of n words and thus the space $\mathcal{Y}$ of possible labels is m^n where m is the number of possible entity types. Instead of explicitly searching over this exponential sized space, we assume that feature vector $\mathbf{f}(\mathbf{x}, \mathbf{y})$ decomposes as a sum of local features that apply over label pairs of adjacent words. This decomposition is exploited for efficient inference over the space of variables $\mathbf{y}$. The local feature vector entries consist of various

properties of a word and its neighboring words. Typical properties useful for entity extraction are the case pattern of the word, its orthographic type, match in a dictionary of known types, its part of speech, and so on.

2.1 Training the weight vector w

During training we are given a labeled set SRC = $\{(\mathbf{x}_\ell, \mathbf{y}_\ell)\}_{\ell=1}^{N}$ consisting of correct input output pairs and our goal is to find the value of $\mathbf{w}$ that will minimize error on future inputs. This training objective is expressed as:

$$\min_{\mathbf{w}} \sum_{\ell} \text{loss}(\mathbf{x}_\ell, \mathbf{y}_\ell, \mathbf{w}, \mathbf{f}) + C||\mathbf{w}||^\gamma \quad (1)$$

where $\text{loss}(\mathbf{x}_\ell, \mathbf{y}_\ell, \mathbf{w}, \mathbf{f})$ is a function that measures for input $\mathbf{x}_\ell$ the error in predicting the label $\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, \mathbf{y})$ given that its correct label is $\mathbf{y}_\ell$. A popular form of the loss function is $\mathbf{w} \cdot \mathbf{f}(\mathbf{x}_\ell, \mathbf{y}_\ell) - \log \sum_{\mathbf{y}} exp(\mathbf{w} \cdot \mathbf{f}(\mathbf{x}_\ell, \mathbf{y}))$. This form of the loss function is both efficient to train and has been found to generalize well to future instances when they follow the same distribution as the training data. The second term $C||\mathbf{w}||^\gamma$ (with γ usually 1 or 2) prevents the model from overfitting $\mathbf{w}$ to the labeled set.

2.2 Domain adaptation

In domain adaptation we are faced with the following situation: there is labeled dataset SRC from one or multiple domains but on the domain on which we need predictions we only have an unlabeled pool DEST. Various existing domain adaptation techniques tend to retrain the model after seeing the examples in DEST [13, 8, 1, 12, 3, 2]. We first give an overview of these methods in Section 3. Next, we give a summary of our method for domain adaptation in record extraction that does not require retraining. Such methods are more useful in settings where we need to handle many target domains and retraining the model for each domain is expensive.

3. CURRENT DOMAIN ADAPTATION TECHNIQUES

The key idea in the various methods of adapting the labeled examples in SRC to provide high accuracy on DEST is to choose a representation of the SRC examples that make them close to the DEST distribution. We discuss three proposed methods of achieving this goal.

3.1 Select relevant examples

One strategy to make the examples in SRC relevant to classifying examples in DEST is to differentially weight examples in SRC such that higher weights are assigned to examples that are similar to our target examples. Let β_i denote the weight assigned to the ith example in SRC. The training objective (Eq. 1) is modified such that the loss of the i-th example is multiplied by β_i. We now discuss how to set β_i.

The ideal weight of each example $\mathbf{x}$ in SRC should be the ratio of its probability in the target and the source domains. As shown in [13], this is the optimum way to align the source distribution to the target distribution. However, this approach is not practical because we do not have a probability distribution over the examples in each domain. Estimating the distribution through the samples is difficult because in general, $\mathbf{x}$ has many dimensions. Therefore, a number of methods have been proposed to estimate the β values directly without first estimating the probability of $\mathbf{x}$ in each of the domains.

A mean matching method proposed in [8] assigns weights β_i to the examples in SRC such that the mean of the weighted examples in SRC matches the means of the examples in DEST. In contrast, [1] uses another classifier to estimate the probability that an example is from the target distribution as against the source distribution. In training this classifier, the examples in DEST are treated as positive examples and the ones in SRC are treated as negative examples.

3.2 Remove irrelevant features

The above method of weighting entire examples is not effective when a few features cause the two domains to differ systematically from each other. For example, if we have a feature called "Is capitalized word" in the source domain but in the target domain every letter is capitalized, then no instance weighting scheme can align the two distributions. In such kinds of mismatch a more effective method of domain adaptation is to differentially weight features instead of examples. A method proposed in [12] is to assign a weight λ_j to each feature j that is equal to the difference in the expected value of the feature in the two domains. The model is then retrained by adding a third term $\sum_j \lambda_j |w_j|$ to the training objective in Equation 1 that penalizes features with large λ_j values so that their role in the final classification is minimized. This method has been shown to provide significant accuracy gains on entity extraction tasks with varying training and test domains [12].

3.3 Add related features

A third strategy proposed in [3, 2] is to add new features to the target domain by aligning them to anchor features in the labeled source domain. Anchor features are those that are frequent in the two domains and are strongly correlated to the class labels. As an example, consider the task of sentiment extraction where the source domain has labeled book reviews and the target domain has unlabeled home appliance reviews. Words like "good", "dissatisfied" and "excellent" that are present in both the domains and strongly correlated with the class labels in the source domain are good candidates for such anchor features. Once a set of anchor features is determined, we find the set of those features from the target and source domains that are strongly correlated with each anchor feature. For example, with the anchor feature "excellent", in the book domain we might find words like "engaging" and in the appliance domain words like "reliable" strongly associated with it. This establishes a correspondence between features "engaging" and "reliable". We refer the reader to [3, 2] for more details on how such a correspondence is quantified

and exploited during model retraining.

4. DOMAIN ADAPTATION FOR RECORD EXTRACTION

We now present our approach for domain adaptation which does not do any model retraining. Our approach uses the key observation that record labelings inside a domain tend to have regularity in many aspects. For example, in a citation labeling task, records in the same list tend to use the same ordering of labels. They also tend to use the same font/formatting style for specific labels, such as bold Titles in one domain, or italicized Titles in another. Thus, regularities tend to exist in each domain, although the nature of each such regularity might vary from domain to domain. To illustrate further, Table 1 lists citation records from two domains. All labelings in the first domain start with Author, while those in the second domain start with a Title. All the titles in the second domain are also hyperlinks to the paper. We call each such regularity-rich aspect a *property*.

Properties are not confined to citation labeling tasks. Infact, they exist and can be exploited in any domain whose records enjoy regularity. Consider the task of extracting products, where each catalog corresponds to a domain. In one catalog, each record might end a product price with USD (thus showing regularity), while in another, prices might be listed with EUR. Table 2 lists more properties for these two tasks. Table 1 illustrates the regularity of some of the properties on the citation labeling task.

The key idea behind using properties for domain adaptation is as follows. If we use our error-prone base model on each record in DEST independently, the noisy output labelings will generally not enjoy regularity simply because regularity in the domain is not captured at all, coupled with the errors made by the model. However, if we jointly label all records in DEST together, and provide an extra incentive for the output labelings to be regular with respect to the affected set of properties, many of the errors made earlier will be corrected. As an example, again consider the property *First Label* that returns the first label in a citation record. If under the base model, a majority but not all records in DEST exhibit regularity, then the extra incentive in our scheme will push the remaining records to conform with the other records and take on the correct first label.

Keeping this high-level picture in mind, we describe our approach with the following components:

1. A collection of instance-labeling pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$. Instances and their labelings are usually structured, e.g. $\mathbf{x}_i$ can be a citation record and $\mathbf{y}_i$ its bibliographic segmentation. Each $\mathbf{y}_i$ is probabilistically modeled using a feature-based prediction model as discussed in Section 2. These models are also popularly known as Markov Random Fields (MRFs) in machine learning literature [9]. From Section 2 recall that the scoring function for assigning labeling $\mathbf{y}_i$ to $\mathbf{x}_i$ is $\mathbf{w} \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i)$, which decomposes over the parts c of the MRF as $\mathbf{w} \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{y}) = \sum_c \mathbf{w} \cdot \mathbf{f}_c(\mathbf{x}_i, \mathbf{y}_c)$. For sentence-like records, a part c is usually a word or a pair of adjacent words, as explained in the entity extraction task in Section 2.

2. A set P of properties where each property $p \in P$ includes in its domain a subset $\mathcal{D}_p$ of MRFs and maps each labeling $\mathbf{y}$ of an input $\mathbf{x} \in \mathcal{D}_p$ to a discrete value from its range $\mathcal{R}'p$. These properties predominantly take only one value for records in a fixed domain. The dominant value however can vary from domain to domain. For tractability, we assume that each property decomposes over parts of the MRF. We discuss decomposable properties in Section 4.1.

3. A clique potential function $\mathrm{C}p(\{p(\mathbf{x}_i, \mathbf{y}_i)\}_{\mathbf{x}_i \in \mathcal{D}_p})$ for each property p. We call it a clique potential because each property p is thought of as a clique, and each member MRF in $\mathcal{D}_p$ a vertex of that clique. This potential is maximized when all the member MRFs get labelings with the same property value. By including these potentials in our objective, we can encourage conformity of properties across labelings of member MRFs. Various symmetric potential functions are discussed in Section 4.2.

Our aim is now to jointly label the N records so as to maximize the sum of the individual MRF specific scores from our trained model, and the clique potentials coupling these MRFs via their property functions. This is given by:

$$\max_{(\mathbf{y}_1,\ldots,\mathbf{y}_N)} \sum_{i=1}^{N} \mathbf{w} \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i) + \sum_{p \in P} \mathrm{C}p(\{p(\mathbf{x}_i, \mathbf{y}_i)\}_{i \in \mathcal{D}_p}) \tag{2}$$

Equation 2 is NP-hard to optimize even for simple cases. Hence, we look at an approximate optimization method instead. Our method uses the well-known paradigm of message passing in a cluster graph [5]. Message passing breaks the computation into two kinds of efficient subroutines: one that exploits the decomposability of the properties, and the other uses algorithms tailored to the specific symmetric potentials being used.

In the subsequent sections, we discuss decomposable properties, symmetric clique potentials, and the joint labeling computation algorithm.

4.1 Decomposable Properties

A property maps a $(\mathbf{x}, \mathbf{y})$ pair to a discrete value in its range. Table 2 gives examples of some properties for various record extraction tasks.

For tractability, we consider only *decomposable properties*, viz. properties that can be broken over the parts c of the MRF of labeling $\mathbf{y}$, just like our base model $\mathbf{w} \cdot \mathbf{f}$. We formally describe decomposable properties as:

DEFINITION 4.1. *A decomposable property $p(\mathbf{x}, \mathbf{y})$ is composed out of component level properties $p(\mathbf{x}, \mathbf{y}_c, c)$*

Domain	Record	p_1	p_2	p_3	p_4
1	Bhardwaj, P. (2001). Delegating Pricing Decisions. *Marketing Science* 20(2). 143-169.	Author	'.'	Venue	Volume
1	Balasubramaniam, S. and P. Bhardwaj (2004). When not all conflict is bad: Manufacturing marketing conflict and strategic incentive design. *Management Science* 50(4). 489-502.	Author	'.'	Venue	Volume
1	Bhardwaj, P. and S. Balasubramaniam (2005). Managing Channel Profits: The Role of Managerial Incentives. Forthcoming *Quantitative Marketing and Economics*.	Author	'.'	Venue	End
2	A Simulator for estimating Railway Line Capacity. *In APORS - 2003.*	Title	Start	Venue	Year
2	Scheduling Loosely Connected Task Graphs. *Journal of Computer and System Sciences , August 2003 .*	Title	Start	Venue	Month
2	Devanagari Pen-written Character Recognition. *In ADCOM - 2001 .*	Title	Start	Venue	Year

Table 1: Illustration of properties $p_1, \ldots, p_4$ applied to records from two bibliographic domains.

Id	Property $p(\mathbf{x}, \mathbf{y})$	Range	Decomposable?
p_1	First non-Other label in $\mathbf{y}$	$Y \setminus \{\text{Other}\}$	Yes
p_2	Token before Title in $\mathbf{y}$	All seen tokens $\cup$ {Start,NoTitle}	Yes
p_3	First non-Other label after Title in $\mathbf{y}$	$Y \setminus \{\text{Other}\} \cup \{\text{End,NoTitle}\}$	Yes
p_4	First non-Other label after Venue in $\mathbf{y}$	$Y \setminus \{\text{Other}\} \cup \{\text{End,NoVenue}\}$	Yes
p_5	Does Title appear after Author in $\mathbf{y}$?	Boolean $\cup$ {NoAuthor,NoTitle}	No
p_6	Number of Titles in $\mathbf{y}$	$\mathbb{N} \cup \{0\}$	No
p_7	Label of fixed token t in $\mathbf{y}$	Y	Yes
p_8	HTML tag containing ProductName in $\mathbf{y}$	All HTML tags	Yes
p_9	Token after Price in $\mathbf{y}$?	USD,GBP,EUR,CAD etc.	Yes
p_{10}	Lowest common ancestor of ProductName and Price	Seen DOM XPaths	No

Table 2: Examples of properties for Bibliographic record extraction and Product extraction. Y is the set of all labels.

defined over parts c of $\mathbf{y}$. $p : (\mathbf{x}, \mathbf{y}_c, c) \mapsto \mathcal{R}_p \cup \{\emptyset\}$ where the special symbol $\emptyset$ means that the property is not applicable to $(\mathbf{x}, \mathbf{y}_c, c)$. $p(\mathbf{x}, \mathbf{y})$ is composed as:

$$p(\mathbf{x}, \mathbf{y}) \triangleq \begin{cases} \emptyset & \text{if } \forall c : p(\mathbf{x}, \mathbf{y}_c, c) = \emptyset \\ v & \text{if } \forall c : p(\mathbf{x}, \mathbf{y}_c, c) \in \{v, \emptyset\} \\ \perp & \text{otherwise.} \end{cases} \quad (3)$$

The first case occurs when the property does not fire over any of the parts of $\mathbf{y}$. The last case occurs when $\mathbf{y}$ has more than one parts where the property has a valid value but the values are different. The new range $\mathcal{R}'p$ now consists of $\mathcal{R}_p$ and the two special symbols $\perp$ and $\emptyset$.

We show that even with decomposable properties we can express many useful types of regularities in labeling multiple MRFs arising in domain adaptation.

Example 1. Consider a property that expresses regularity in the order of labels in a collection of bibliography records. Let $\mathbf{x}$ be a bibliographic record and $\mathbf{y}$ its labeling. Define property p_3, which returns the first non-Other label in $\mathbf{y}$ after a Title. A label 'End' marks the end of $\mathbf{y}$. So $\mathcal{R}_p$ contains 'End' and all labels except Other. So when p_3 is applied to a part c labeled Title, it returns the first non-Other label in $\mathbf{y}$ after c. Thus,

$$p_3(\mathbf{x}, y_c, c) \triangleq \begin{cases} \beta & (y_c = \text{Title}) \text{ and } (y_{c+i} = \beta) \text{ and} \\ & (y_{c+j} = \text{Other},\ j = 1, \ldots, i-1) \\ \text{End} & (y_c = \text{Title}) \text{ and } (\mathbf{y} \text{ ends at } c) \\ \emptyset & y_c \neq \text{Title} \end{cases}$$

Therefore,

$$p_3(\mathbf{x}, \mathbf{y}) \triangleq \begin{cases} \emptyset & \mathbf{y} \text{ has no Title} \\ \beta & \beta \text{ is the first non-Other label after } \textbf{each} \\ & \text{Title in } \mathbf{y} \\ \perp & \text{otherwise} \end{cases}$$

Example 2. Consider a property, denoted by p_2, whose range is the space of tokens. This property returns the identity of the token before a Title in $\mathbf{y}$. So,

$$p_2(\mathbf{x}, y_c, c) \triangleq \begin{cases} x_{c-1} & y_c = \text{Title and } (c > 0) \\ \text{'Start'} & y_c = \text{Title and } (c = 0) \\ \emptyset & y_c \neq \text{Title} \end{cases}$$

Therefore,

$$p_2(\mathbf{x}, \mathbf{y}) \triangleq \begin{cases} \emptyset & \text{No Title in } \mathbf{y} \\ \text{'Start'} & \text{The only Title in } \mathbf{y} \text{ is at the start} \\ t & \text{All Titles in } \mathbf{y} \text{ preceded by token } t \\ \perp & \mathbf{y} \text{ has multiple Titles with different} \\ & \text{preceding tokens} \end{cases}$$

4.2 Symmetric Clique Potentials

A symmetric clique potential depends only on the number of clique vertices taking a property value v, denoted by n_v, and not on the identity of those vertices. In other words, such a potential is invariant under any permutation of its arguments and the potential's value

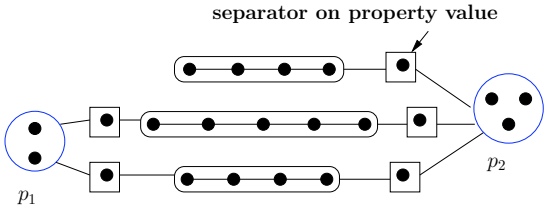


Figure 1: Cluster graph for a toy example with three chain-shaped MRF instances and two properties.

is derived from the histogram of counts $\{n_v | \forall v \in V\}$, where V is the set of possible property values. We denote this histogram by the vector $\mathbf{n}$. An associative potential is maximized when $n_v = n$ for some v, i.e. one value is given to all the clique vertices.

Three popular associative clique potential families are listed in Table 3. We specifically discuss a very prominent symmetric clique potential — Potts potential, which we also use in our experiments.

Potts Potential

The Potts potential corresponds to the negative Gini index of the property values at the clique vertices:

$$\mathrm{C}^{\mathrm{Potts}} = \mathrm{C}(n_1, \ldots, n_{|V|}) = \frac{\lambda}{n} \sum_{v \in V} n_v^2 \qquad (4)$$

where n is the number of vertices. Potts potential counts (upto a constant) the number of clique edges, both of whose end vertices have the same property value. Potts potential have been extensively used in statistical physics in the garb of Ising models, in image pixel labeling tasks e.g. [4], in associative Markov networks [15] to model associative labeling tasks, and in skip-chain MRFs [14] to model non-local associative dependencies between repeated occurrences of a word.

4.3 Joint Record Labeling

The individual MRFs coupled with symmetric clique potentials form a natural cluster graphical model, such as the toy model shown in Figure 1. To compute the labelings of all MRFs jointly, we perform message passing on this cluster graph.

Message passing on the cluster graph can be seen as an optimization-by-parts heuristic to solve Equation 2. In a single iteration of this heuristic, each cluster computes its own belief of what its own labeling should be and passes this message to the neighboring clusters, which in turn compute their beliefs about the labels of their member nodes. These computations are interdependent because the clusters have overlapping members.

In our setup, this leads to computing messages from MRFs to property cliques and vice versa. Complete message computation details are provided in [7]. We only provide the outline here. First, to compute a message from an MRF to a property clique, we look at the message computation algorithm *inside* the MRF and fold-in the property. Folding is possible because the properties are decomposable in the same way as the MRF model. Second, to compute the reverse message, we invoke highly efficient potential-specific combinatorial algorithms at the clique. Some such algorithms are presented in [6] and [7] for a variety of potentials.

After sufficient rounds of interaction via message passing, each MRF reports its best labeling independently.

5. EXPERIMENTAL RESULTS

We demonstrate the domain adaptability of our approach and show that using a good set of properties can bring down the test error significantly.

We focus on the bibliographic information extraction task, where the aim is to adapt a base model across widely varying publications pages of authors. Our dataset consists of 433 bibliographic entries from the webpages of 31 authors, hand-labeled with 14 labels such as Title, Author, Venue, Location and Year. Bibliographic entries across different authors differ in various aspects like label-ordering, missing labels, punctuation, HTML formatting and bibliographic style.

Varying fractions of 31 domains were used to train a base model. We used standard extraction features in a window around each token, along with label transition features [10].

For our collective labeling framework, we use properties $p_1, \ldots, p_4$ from Table 2. We use Potts potential for each property, with $\lambda = 1$. Some of these properties, e.g. p_3, operate on non-adjacent labels, and thus are not Markovian. This can be easily rectified by making 'Other' an extension of its predecessor label, e.g. an 'Other' segment after 'Title' can be relabeled as 'After-Title'.

The performance results of the collective model with the above properties versus the baseline model are presented in Table 4. For the test domains, we report token F1 accuracy of the important labels — Title, Author and Venue. F1 accuracy of a label l is the harmonic mean of the precision and recall of l, defined as:

$$\mathrm{Prec}(l) = \frac{\#\text{ tokens correctly marked } l \text{ by the model}}{\#\text{ tokens marked } l \text{ by the model}}$$

$$\mathrm{Recall}(l) = \frac{\#\text{ tokens correctly marked } l \text{ by the model}}{\#\text{ tokens with true label } l}$$

F1 accuracies reported in Table 4 are averaged over five trials. The collective model leads to up to 25% reduction in the test error for Venue and Title, labels for which we had defined related properties. Figure 2 shows the error reduction on individual test domains for one particular split when five domains were used for training and 26 for testing. The errors are computed from the combined token F1 scores of Title, Venue and Author. For some domains the errors are reduced by more than 50%. Collective inference increases errors in only two domains. In those domains, a majority of the labelings output by the base model take on wrong property values. Thus by encouraging conformity, the

Name	Form	Remarks
MAX	$\max_v f_v(n_v)$	f_v is a non-decreasing function
SUM	$\sum_v f_v(n_v)$	f_v non-decreasing. Includes the Potts potential $= \frac{\lambda}{n}\sum_v n_v^2$
MAJORITY	$f_a(\mathbf{n})$, where $a = \text{argmax}_v n_v$	f_a is typically linear

Table 3: Various families of symmetric clique potentials. $\mathbf{n} = (n_1, \ldots, n_{|V|})$ is the histogram of property values in the clique.

	Title		Venue		Author	
Train %	Base	CI	Base	CI	Base	CI
5	70.7	74.8	58.8	62.5	74.1	74.3
10	78.0	82.1	69.2	72.2	75.6	75.9
20	85.8	88.6	76.7	78.9	80.7	80.7
30	91.7	93.0	81.5	82.6	87.7	88.0
50	92.3	94.2	83.5	84.5	89.4	90.0

Table 4: Token-F1 of the Collective and Base Models on the Bibliographic Extraction Task

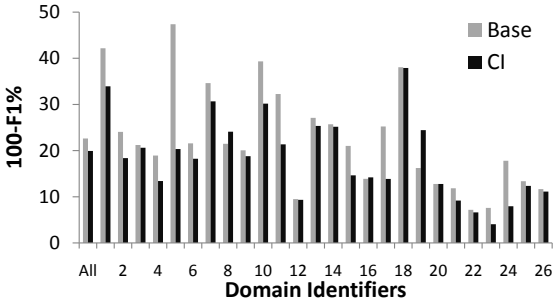


Figure 2: Per-domain error for the base and collective inference (CI) model

remaining labelings also take on the wrong value causing a slight dip in accuracy.

6. CONCLUSION

We presented a mini-survey of domain adaptation approaches for information extraction. We summarized our proposed approach that jointly labels records in a target domain without retraining any model. Our framework encourages records to jointly take labelings that are conformant with respect to a set of domain-independent properties. We also presented the joint-labeling algorithm for the new graphical model that results from our setup. Finally, we demonstrated our framework on a bibliographic task, where we showed significant gains by exploiting intra-domain regularity.

7. REFERENCES

[1] S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, 2007.

[2] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.

[3] J. Blitzer, R. McDonald, and F. Pereira. Domain Adaptation with Structural Correspondence Learning. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[5] J. Duchi, D. Tarlow, G. Elidan, and D. Koller. Using combinatorial optimization within max-product belief propagation. In *Advances in Neural Information Processing Systems (NIPS 2006)*, 2007.

[6] R. Gupta, A. A. Diwan, and S. Sarawagi. Efficient inference with cardinality-based clique potentials. In *Proceedings of the 24th International Conference on Machine Learning (ICML), USA*, 2007.

[7] R. Gupta and S. Sarawagi. A generalized framework for collective inference with applications in domain adaptation, Under Preparation.

[8] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf. Correcting Sample Selection Bias by Unlabeled Data. In *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2007. MIT Press.

[9] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML-2001)*, Williams, MA, 2001.

[10] F. Peng and A. McCallum. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL*, pages 329–336, 2004.

[11] S. Sarawagi. Information extraction. *FnT Databases*, 1(3), 2008.

[12] S. Satpal and S. Sarawagi. Domain adaptation of conditional probability models via feature subsetting. In *ECML/PKDD*, 2007.

[13] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, pages 227–244, 2000.

[14] C. Sutton and A. McCallum. Collective segmentation and labeling of distant entities in information extraction. Technical Report TR # 04-49, University of Massachusetts, July 2004. Presented at ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields.

[15] B. Taskar, V. Chatalbashev, and D. Koller. Learning associative markov networks. In *Twenty First International Conference on Machine Learning (ICML04), Banff, Canada.*, 2004.

The YAGO-NAGA Approach to Knowledge Discovery

Gjergji Kasneci, Maya Ramanath, Fabian Suchanek, Gerhard Weikum
Max Planck Institute for Informatics
D-66123 Saarbruecken, Germany
kasneci,ramanath,suchanek,weikum@mpi-inf.mpg.de

ABSTRACT

This paper gives an overview on the YAGO-NAGA approach to information extraction for building a conveniently searchable, large-scale, highly accurate knowledge base of common facts. YAGO harvests infoboxes and category names of Wikipedia for facts about individual entities, and it reconciles these with the taxonomic backbone of WordNet in order to ensure that all entities have proper classes and the class system is consistent. Currently, the YAGO knowledge base contains about 19 million instances of binary relations for about 1.95 million entities. Based on intensive sampling, its accuracy is estimated to be above 95 percent. The paper presents the architecture of the YAGO extractor toolkit, its distinctive approach to consistency checking, its provisions for maintenance and further growth, and the query engine for YAGO, coined NAGA. It also discusses ongoing work on extensions towards integrating fact candidates extracted from natural-language text sources.

1. INTRODUCTION

Universal, comprehensive knowledge bases have been an elusive AI goal for many years. Ontologies and thesauri such as OpenCyc, SUMO, WordNet, or UMLS (for the biomedical domain) are achievements along this route. But they are typically focused on intensional knowledge about semantic classes. For example, they would know that mathematicians are scientists, that scientists are humans (and mammals and vertebrates, etc.); and they may also know that humans are either male or female, cannot fly (without tools) but can compose and play music, and so on. However, the currently available ontologies typically disregard the extensional knowledge about individual entities: instances of the semantic classes that are captured and interconnected in the ontology. For example, none of the above mentioned ontologies knows more than a handful of concrete mathematicians (or famous biologists etc.). Today, the best source for extensional knowledge is probably Wikipedia, providing a wealth of knowledge about individual entities and their relationships. But most of this knowledge is only latent, by being embedded in the natural-language text of Wikipedia articles or, in the best case, reflected in the semistructured components of Wikipedia: infoboxes and the category system.

A comprehensive knowledge base should know all individual entities of this world (e.g., Nicolas Sarkozy), their semantic classes (e.g., Sarkozy isa Politician), relationships between entities (e.g., Sarkozy presidentOf France), as well as validity times and confidence values for the correctness of such facts. Moreover, it should come with logical reasoning capabilities and rich support for querying. The benefits from solving this grand challenge would be enormous. Potential applications include but would not be limited to:

1. a machine-readable, formalized encyclopedia that can be queried with high precision like a semantic database;

2. an enabler for semantic search on the Web, for detecting entities and relations in Web pages and reasoning about them in expressive (probabilistic) logics;

3. a backbone for natural-language question answering that would aid in dealing with entities and their relationships in answering who/where/when/ etc. questions;

4. a key asset for machine translation (e.g., English to German) and interpretation of spoken dialogs, where world knowledge provides essential context for disambiguation;

5. a catalyst for acquisition of further knowledge and largely automated maintenance and growth of the knowledge base.

To illustrate the first two points, consider the following difficult "knowledge queries" that a student, journalist, or researcher may pose to the Web:

Q1: Which Grammy winners were born in Europe?
Q2: Which French politicans are married to singers?
Q3: Which Nobel prize winners had an academic advisor who graduated from the same university?
Q4: Give me a comprehensive list of HIV drugs that inhibit proteases (a specific family of enzymes).

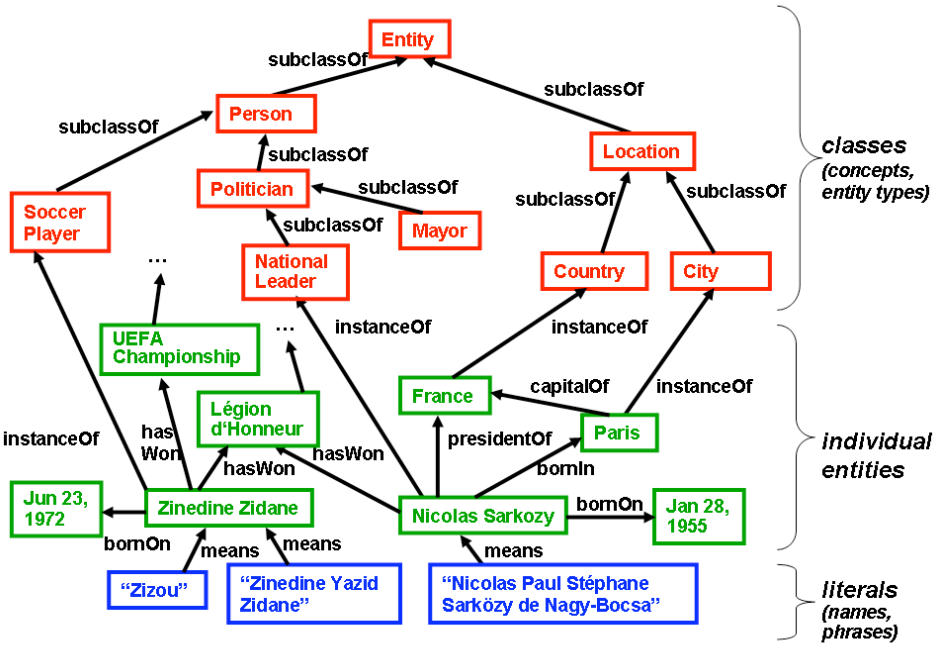


Figure 1: Excerpt of the YAGO Knowledge Base

Regardless of how well these information needs are formulated as keyword queries or question phrases, current search engines would hardly produce good answers. In the best case, the user would have to browse through many potentially relevant pages and manually compile the bits and pieces of the desired answers. A large knowledge base would support precise query formulation (not necessarily in natural language) that explicitly indicates entities and relations, and would provide the best answers in a concise manner.

This paper gives an overview of the YAGO-NAGA[1] approach to automatically building and maintaining a conveniently searchable, large and highly accurate knowledge base, by applying information-extraction (IE) methods to Wikipedia and other sources of latent knowledge. The project started in summer 2006 at the Max Planck Institute for Informatics, with continuous enhancements and extensions.

The YAGO knowledge base represents all facts in the form of unary and binary relations: classes of individual entities, and pairs of entities connected by specific relationship types. This data model can be seen as a typed graph with entities and classes corresponding to nodes and relations corresponding to edges. It can also be interpreted as a collection of RDF triples with two adjacent nodes and their connecting edge denoting a (subject, predicate, object) triple. Figure 1 shows an excerpt of the knowledge base. The knowledge base is publicly available at `http://www.mpi-inf.mpg.de/~suchanek/yago`. It can be queried for knowledge discovery by the NAGA search engine. An online demo is accessible at `http://www.mpi-inf.mpg.de/~kasneci/naga`.

Section 2 outlines the architecture of YAGO. Section 3 presents the extractor toolkit for building the YAGO core knowledge base. Section 4 presents the consistency checking methods, which ensure the high accuracy of YAGO's facts. Section 5 discusses our ongoing work on how YAGO can be automatically maintained and further grown. Section 6 presents the NAGA model and system for querying YAGO and ranking search results.

2. SYSTEM ARCHITECTURE

The YAGO architecture is depicted in Figure 2. In contrast to many other IE projects, YAGO emphasizes high accuracy and the consistency of the harvested knowledge rather than aiming for high recall (coverage) of facts. YAGO primarily gathers its knowledge by integrating information from Wikipedia and WordNet. It performs rule-based IE on the infoboxes and category system of Wikipedia, and reconciles the resulting facts with WordNet's taxonomical class system. This is done by performing consistency checks whenever a new fact is considered for addition to the knowledge base. This approach resulted in the *YAGO core knowledge base* [18, 19], currently containing 249,015 classes, 1,941,578 in-

[1]YAGO = Yet Another Great Ontology
NAGA = Not Another Google Answer

dividual entities, and about 19 million facts (instances of binary relations) for 93 different relations. Extensive sampling showed that the accuracy is at least 95 percent [19], and many of the remaining errors (false positives) are due to entries in Wikipedia itself (which we considered as ground truth).

As the rule-based core extractors are limited in coverage, YAGO can also employ pattern-, NLP- and learning-based IE techniques [1, 5, 15] on text sources such as Wikipedia texts, news articles, research literature, or Web pages of interest and clear style. These techniques, in combination with the diversity and mixed quality of the sources, introduce a higher risk of degrading in precision, and are computationally much more expensive. Therefore, the text-oriented harvesting of YAGO is carried out in two phases. The *gathering phase* employs recall-oriented IE methods, and aims at high throughput. The output is interpreted as a set of fact hypotheses. Subsequently, the *scrutinizing phase* assesses the hypotheses against the existing knowledge base, in a batched manner, and filters out facts that show high indication of being inconsistent with essential invariants and prior knowledge (e.g., that a person's birth place is unique and that certain cities are located in Europe so that an American-born person cannot be born in such a city).

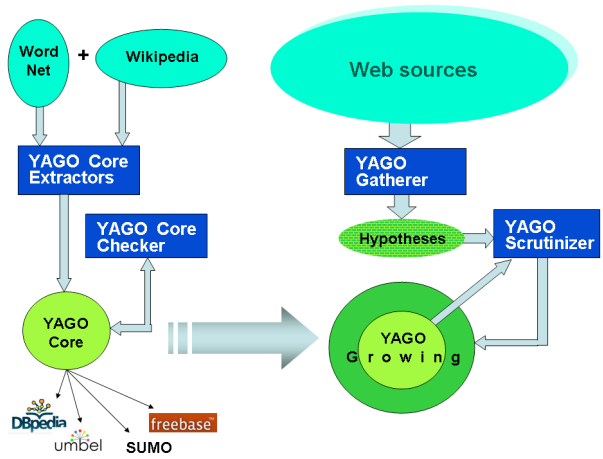


Figure 2: The YAGO Architecture

3. YAGO CORE EXTRACTORS

Wikipedia Infoboxes. Wikipedia has two kinds of - almost structured - information components on article pages that can be effectively harvested: the *infoboxes* and the *category system*. Infoboxes are collections of attribute name-value pairs. They are often based on templates and then reused for important types of entities such as countries, companies, scientists, music bands, sports teams, etc. For example, the infobox for Nicolas Sarkozy gives us data such as *birth_date = 28 January 1955*, *birth_place = Paris*, *occupation = lawyer*, and *alma_mater = University of Paris X: Nanterre*. YAGO uses a suite of rules for frequently used infobox attributes to extract and normalize the corresponding values. For example, the *spouse* attribute is mapped to the *marriedTo* relation, and the extracted fact then is *Nicolas Sarkozy marriedTo Carla Bruni*. YAGO does not attempt to extract all infobox attributes, as their "long tail" has a lot of naming diversity and noise (see [20, 21] for a more recent, universal attempt at harvesting infoboxes).

Wikipedia Categories. As for the category system, the Wikipedia community has manually placed (the article about) Nicolas Sarkozy into categories such as: *Presidents_of_France*, *Légion_d'honneur_recipients*, or *Alumni_of_Sciences_Po* (the Paris Institute of Political Studies). These give YAGO clues about instanceOf relations, and we can infer that the entity Nicolas Sarkozy is an instance of the classes PresidentsOfFrance, LégionD'HonneurRecipients, and AlumniOfSciencesPo. Occasionally, YAGO encounters a misleading category name that does not indicate a class membership (instanceOf). An example is the category *Hollywood_Walk_of_Fame*, which includes many actors and musicians. Semantically, however, these people are not instances of a class *Walk* (which would be a subclass of *Motion*), nor are they instances of some superclass *Awards* but rather awards winners which in turn would be a subclass of humans. YAGO employs linguistic processing (noun phrase parsing) and also some heuristics (e.g., the head word in the noun phrase should be in plural form), in order to cope with these pitfalls and achieve high accuracy in harvesting the category system.

Ongoing Work: Temporal Validity. The world is dynamic, and facts change with time. When we ran YAGO on Wikipedia in spring 2007, we extracted the fact *Jacques Chirac presidentOf France*, seemingly contradicting our current result *Nicolas Sarkozy presidentOf France*. To resolve this situation, we need temporal annotations for facts. To ensure that our knowledge model is not inflated with ad-hoc features, we decided to adapt the reification technique of RDF and use our standard binary-relation approach to represent validity times. This works as follows. Both of the above facts have identifiers, say *Id1* (for the fact about Chirac) and *Id2* (for the fact about Sarkozy), to which we can refer in other facts. We create additional facts like *Id1 ValidSince 17 May 1995*, *Id1 ValidUntil 16 May 2007*, and *Id2 ValidSince 17 May 2007*. This technique is general, it can also be used to represent arbitrary ternary and higher-arity relations in our model. Likewise, meta-relations such as *witnesses* for a fact - remembering sources that support a fact's validity, use the same representation.

In principle, YAGO could also handle divorces and identify spouses for specific time periods, but often this kind of information is not available in the semistructured parts of Wikipedia. Therefore, we have starting working on specialized text-oriented extractors with the specific target of detecting validity times. One problem in this task is that temporal statements often refer to relative timepoints (e.g., last Monday), have different

granularities such as "May 2007" vs. "17 May 2007", or are incomplete (e.g., showing either an "until" or "since" statement but not both, even for terminated intervals). To deal with these situations, we represent every timepoint as a pair of (earliest, latest) intervals and use plus/minus infinity for missing information in such an interval. As we find more accurate or complete time statements or gather more evidence for certain facts in the IE process, we can refine the temporal facts in the knowledge base [23].

4. YAGO CONSISTENCY CHECKERS

Rationale. YAGO pays particular attention to the consistency of its knowledge base. Solely performing IE on infoboxes and categories of Wikipedia may result in a large but incoherent collection of facts: redundant and potentially inconsistent, overly specific in some parts and left blank in others in an arbitrary way. For example, we may know that Nicolas Sarkozy is an instance of PresidentsOfFrance, but we may not be able to automatically infer that he is also an instance of Presidents and most likely also an instance of FrenchPeople. Likewise, the fact that he is a president (or minister of the interior or mayor to also include former positions) does not automatically tell us that he is a politician. To overcome these problems, YAGO intensively uses the WordNet thesaurus and integrates the facts from Wikipedia with the taxonomic backbone provided by WordNet, using techniques outlined in the rest of this section.

Class Hierarchy. YAGO initializes its class system by importing all WordNet classes and their hypernymy/hyponymy (superclass/subclass) relations. Each individual entity that YAGO discovers in Wikipedia needs to be mapped into at least one of the existing YAGO classes. If this fails, the entity and its related facts are not admitted to the knowledge base. Analogously, classes that are derived from Wikipedia category names such as PresidentsOfFrance need to be mapped, with a subclassOf relationship, to one or more proper superclasses such as Presidents or FrenchPeople. These procedures ensure that we can maintain a consistent knowledge base, where consistency eliminates dangling entities or classes and also guarantees that the subclassOf relation is acyclic. The empirically assessed accuracy of the subclassOf relation is around 97 percent.

Type Conditions. As all entities are typed by their assignment to one or more classes, binary relations have a type signature as well. For example, the isCEOof relation can be specified to have a domain BusinessPerson or simply Person and a range Company (where we denote relations as powerset functions). When the extractor produces a candidate fact like *Nicolas Sarkozy isCEOof France*, we can reject it because the second argument, France, is not a company. Similarly, the hypothesis *Nicolas Sarkozy marriedTo Élysée Palace* which may, perhaps, be incorrectly inferred from sentences such as "Nicolas Sarkozy loves his work with the Élysée Palace" (or from an incorrect interpretation of the infobox attribute Residence), is falsified by the type invariant that marriages are between persons.

Ongoing Work: Constraints on Relations. A very powerful constraint for the consistency of IE results is declaring a relation to be transitive and acyclic (or requiring that its transitive closure is acyclic). The class hierarchy is one important usage case; the locatedIn relation between geographic entities is another one. Going even further, we are considering support for functional dependencies, inclusion dependencies, and inverse relations. For example, a person can have only one birth place, which would allow us to prune out many spurious candidates for alternative birth places that we may see in Web pages. As for inclusion dependencies, we can derive *Paris locatedIn France* from the fact *Paris capitalOf France*, without necessarily seeing it in explicit form. Finally, the presidentOf fact would be an example for exploiting inverse relations. Once we accept that *Nicolas Sarkozy presidentOf France*, we could automatically infer that *France hasPresident Nicolas Sarkozy* (for the same time interval).

5. GROWING YAGO

Keeping YAGO Up-To-Date. We can maintain the YAGO knowledge base, with its current scope, by periodically re-running the extractors on Wikipedia and WordNet. It takes a few hours to build the entire knowledge base. When validity times or intervals can be properly inferred, this would retain previously compiled facts (e.g., former CEOs) as long as they do not cause any inconsistencies.

Adding Natural-Language Text Sources. Further growth of YAGO, beyond the harvesting of infoboxes and category systems, calls for text-oriented IE with more or less powerful NLP capabilities. Our own tool LEILA [17] can be employed for this purpose. It uses a dependency-grammar parser for deep parsing of natural-language sentences, with heuristics for anaphora resolution (e.g., pronouns referring to subjects or objects in a preceding sentence). This produces a tagged graph representation, whose properties can be encoded as features for a statistical learner (e.g., an SVM) that classifies fact candidates into acceptable facts vs. false hypotheses. LEILA provides reasonably good accuracy, but requires about a minute to process a long Wikipedia article on a standard PC, and works for one specific relation at a time. Simpler NLP methods, such as part-of-speech tagging (for word categories: nouns, verbs, etc.), are much faster but would have significantly lower precision. Statistics, like frequencies of witnesses, can be used to improve precision, but proper tuning of statistical thresholds is all but easy. The YAGO architecture supports all these approaches. However, with the open issues in understanding the three-way tradeoffs between precision, recall, and efficiency, we do not yet employ these techniques at large scale.

Ongoing Work. For growing YAGO we can leverage the existing YAGO core in several ways. We believe that the core contains more or less all entities of inter-

est and their most important classes. For example, all notable politicians, athletes, pop stars, movies, cities, rivers, etc. should have a Wikipedia entry, and thus are included in YAGO, too. Certainly, this does not hold for classes like computer scientists, medical drugs, etc. But we could incorporate other sources such as DBLP or UMLS, and adapt the YAGO extractors to them. With the YAGO core as semantic backbone, we can quickly test sentences, paragraphs, or entire Web pages as to whether they contain one or two interesting entities. And when aiming at a particular binary relation, we can exploit our type system: a sentence or paragraph is promising only if it contains two entities of the proper types. For example, for hypothesizing a fact of the isCEOof relation, a sentence must contain a person and a company to be worth undergoing deeper analysis. Testing the type of an entity is a fast lookup in the core knowledge base.

To preserve the consistency of YAGO when adding new facts gathered with "riskier" IE methods, we can utilize the type and constraint checkers that YAGO already has. For efficiency, we batch newly acquired facts and run the consistency checking procedures for several thousand hypotheses together, dropping those that violate vital checks or too many "soft constraints".

6. QUERYING YAGO BY NAGA

Query Language. For querying the YAGO knowledge base, we have designed a query language that builds on the concepts of SPARQL (the W3C standard for querying RDF data), but extends these capabilities by more expressive pattern matching. Our prototype system, NAGA (Not Another Google Answer) [10], implements this query language and provides a statistical ranking model for query results.

A query is a conjunction of *fact templates*, where each template would have to be matched by an edge and its incident nodes in the knowledge graph. For example, the first two example queries of Section 1 can be expressed as follows:

Q1: $x hasWonPrize GrammyAward,
$x bornIn $y,
$y locatedIn Europe
Q2: $x isa politician,
$x citizenOf France,
$x marriedTo $y,
$y isa singer

where $x and $y are variables for which we are seeking bindings so that all query patterns are matched together.

The relation names in the query can also be regular expressions, which have to be matched by an entire path in the knowledge graph. This is a powerful way of dealing with transitive relations and variants of relations where the user may not exactly know by which relations the entities of interest are connected. For example, if the *bornIn* relation actually refers to cities and the *locatedIn* relation captures a city-county-state-country hierarchy, we should replace the last condition in Q1 by the fact template *$y (locatedIn)* Europe*. And if we do not care whether the persons that we are looking for are born in Europe or are citizens of a European country, we may use the template *$y (citizenOf | bornIn | originatesFrom).(locatedIn)* Europe* instead of the last two conditions of Q1. Regular expressions are also helpful in dealing with the class hierarchy in a flexible way. In Q2 the relation label *isa* is actually a shorthand notation for the regular expression *instanceOf.(subclassOf)**, thus enabling ministers or mayors, which are subclasses of politicians, to be returned as query results.

The query language also provides support for formulating temporal conditions on the validity of the facts of interest. For example, if we want to retrieve all French presidents whose terms started in this millenium, we can phrase this as:

f: ($x presidentOf France),
(f since $t),
($t after 31 December 1999)

We are working on enhancing the query language to provide more elaborate capabilities for temporal queries. NAGA has further advanced features, most notably, for specifying relatedness queries among a set of entities [10, 11]. For example, the query:

connect (Nicolas Sarkozy, Zinedine Zidane,
Gerard Depardieu, Miles Davis)

asks for commonalities or other relationships among Sarkozy, the soccer player Zidane, the actor Depardieu, and the trumpet player Miles Davis. A possible answer (technically, a Steiner tree in the underlying knowledge graph) could be that all four are recipients of the French Légion_d'honneur order.

Ranking. Whenever queries return many results, e.g., hundreds of (mostly unimportant) politicians, we need ranking. NAGA employs a novel kind of *statistical language model (LM)* [12, 22] for this purpose, capturing the *informativeness* of a query result [10]: users prefer salient facts or interesting facts, e.g., Nicolas Sarkozy and not the mayor of a small provincial town. In addition, we need to consider the *confidence* that the result facts are indeed correct. Our IE methods assign a confidence weight to each fact f in the knowledge base based on the empirically assessed goodness of the extractor and the extraction target (e.g., rule-based for birthdates vs. linguistic for birth places) and the trustworthiness of the fact's witnesses s (i.e., sources from which it was extracted). One possible way of combining these aspects (among various options) would be:

$$confidence(f) = max\ \{accuracy(f,s) \times trust(s) \mid s \in witnesses(f)\}$$

where trustworthiness could be based on PageRank-style authority or on empirical assessment by experts (e.g., high for Wikipedia, low for blogs). The confidence in a query-result graph that consists of multiple facts is the product of the individual facts' confidence values, postulating statistical independence among the facts.

For informativeness, NAGA employs an LM for graph-structured data. In the following we give a simplified

explanation of the model introduced in [10]. Conceptually, we construct a statistical model for each possible result graph g with connected edges (facts) g_i, and consider the probability that the query q, consisting of fact templates q_i, was generated from g:

$$P[q|g] = \prod_i \lambda P[q_i|g_i] + (1-\lambda)P[q_i]$$

where we factorize over edges for tractability and use $P[q_i]$ for smoothing with parameter λ (analogously to standard LM's). Applying Bayes' rule, simplifying the resulting expression and omitting sub-expressions that do not influence the ranking of results, we obtain:

$$P[q|g] \sim \prod_i \frac{P[q_i|g_i]}{P[q_i]}$$

We can interpret $1/P[q_i]$ as an idf-style weighting of the individual subqueries (emphasizing the more selective patterns in the query). The main parameters to be estimated are the $P[q_i|g_i]$ values, which reflect informativeness for the given query. We use a "background corpus" for this purpose, either a large Web sample or the entirety of Wikipedia texts. We compute the number of witnesses for g_i, that is, the frequency of the two entities (or classes) in g_i co-occurring (in the same sentence, paragraph, or Web page). Analogously, the number of witnesses for q_i is the frequency of the non-variable parts of q_i occurring together. Our current implementation precomputes these statistics based on the Wikipedia corpus. With these ingredients we can finally set

$$P[q_i|g_i] \approx \frac{\#witnesses(g_i)}{\#witnesses(q_i)}$$

For example, as partial results to query Q1, famous Grammy winners such as Eric Clapton, Phil Collins, or Enya should be among the highest ranked results.

For the overall scoring of query results, NAGA uses a weighted linear combination of informativeness and confidence:

$$score(q,g) = \alpha \prod_i \frac{P[q_i|g_i]}{P[q_i]} + (1-\alpha) \prod_i confidence(g_i)$$

Ongoing Work: Personalization. The notion of informativeness is, strictly speaking, a subjective measure: an individual user wants to see a salient result that is also interesting to her. This calls for a *personalized ranking model* or at least a user-group-specific model. An elegant property of the LM approach pursued in NAGA is that we can easily compose multiple LM's using a probabilistic mixture model. We can estimate parameters of a user- or community-specific LM and combine this with a global LM, both models using the same mathematical structure but different parameters.

For the personalized LM, we monitor the history of queries and browsing interactions on the online knowledge base. A click on a fact is interpreted as positive feedback that the fact is interesting to the user, and this evidence is spread to the graph neighborhood, with exponential decay and attention to the edge types along which propagation is meaningful [7]. As an example, assume that a user has intensively explored epic movies and orchestral music, and then poses query Q1. The personalized ranking would prioritize European film-music composers such as Ennio Morricone, Hans Zimmer, or Javier Navarrete.

7. CONCLUSION

The YAGO core is publicly available and has been imported into and integrated with various other knowledge-management projects including DBpedia (`dbpedia.org`), SUMO (`www.ontologyportal.org`), UMBEL (`umbel.org`), and Freebase (`www.freebase.com`). Our ongoing work to improve YAGO mostly centers around making it larger while retaining its high accuracy. This entails deeper considerations on scalability issues, for example, by utilizing database-style query processing and optimization techniques, along the lines of [9].

YAGO shares many of its goals and methodologies with parallel projects along related lines. These include Avatar [14], Cimple/DBlife [6, 16], DBpedia [2], KnowItAll/TextRunner [8, 3, 4], Kylin/KOG [20, 21], and the Libra technology [13, 24] (and probably more). Together they form an exciting trend of leading research towards the elusive goal of machine-readable, comprehensive knowledge bases.

8. REFERENCES

[1] Eugene Agichtein: Scaling Information Extraction to Large Document Collections. IEEE Data Eng. Bull. 28(4), 2005

[2] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, Zachary G. Ives: DBpedia: A Nucleus for a Web of Open Data. ISWC/ASWC 2007

[3] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, Oren Etzioni: Open Information Extraction from the Web. IJCAI 2007

[4] Michael J. Cafarella, Christopher Re, Dan Suciu, Oren Etzioni: Structured Querying of Web Text Data: A Technical Challenge. CIDR 2007

[5] Hamish Cunningham: An Introduction to Information Extraction. In: Encyclopedia of Language and Linguistics, 2nd Edition, Elsevier, 2005

[6] Pedro DeRose, Warren Shen, Fei Chen, AnHai Doan, Raghu Ramakrishnan: Building Structured Web Community Portals: A Top-Down, Compositional, and Incremental Approach. VLDB 2007

[7] Minko Dudev, Shady Elbassuoni, Julia Luxenburger, Maya Ramanath, Gerhard Weikum: Personalizing the Search for Knowledge. PersDB 2008.

[8] Oren Etzioni, Michael J. Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen

Soderland, Daniel S. Weld, Alexander Yates: Unsupervised Named-Entity Extraction from the Web: An Experimental Study. Artif. Intell. 165(1), 2005

[9] Panagiotis G. Ipeirotis, Eugene Agichtein, Pranay Jain, Luis Gravano: Towards a Query Optimizer for Text-Centric Tasks. ACM Trans. Database Syst. 32(4), 2007

[10] Gjergji Kasneci, Fabian M. Suchanek, Georgiana Ifrim, Maya Ramanath, Gerhard Weikum: NAGA: Searching and Ranking Knowledge. ICDE 2008

[11] Gjergji Kasneci, Maya Ramanath, Mauro Sozio, Fabian M. Suchanek, Gerhard Weikum: STAR: Steiner Tree Approximation in Relationship-Graphs. ICDE 2009

[12] Xiaoyong Liu, W. Bruce Croft: Statistical Language Modeling for Information Retrieval. Annual Review of Information Science and Technology 39, 2004

[13] Zaiqing Nie, Yunxiao Ma, Shuming Shi, Ji-Rong Wen, Wei-Ying Ma: Web Object Retrieval. WWW 2007

[14] Frederick Reiss, Sriram Raghavan, Rajasekar Krishnamurthy, Huaiyu Zhu, Shivakumar Vaithyanathan: An Algebraic Approach to Rule-Based Information Extraction. ICDE 2008

[15] Sunita Sarawagi: Information Extraction. Foundations and Trends in Databases 2(1), 2008

[16] Warren Shen, AnHai Doan, Jeffrey F. Naughton, Raghu Ramakrishnan: Declarative Information Extraction Using Datalog with Embedded Extraction Predicates. VLDB 2007

[17] Fabian M. Suchanek, Georgiana Ifrim, Gerhard Weikum: Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents. KDD 2006

[18] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum: YAGO: a Core of Semantic Knowledge. WWW 2007

[19] Fabian Suchanek, Gjergji Kasneci, Gerhard Weikum: YAGO: A Large Ontology from Wikipedia and WordNet. Journal of Web Semantics, 2008

[20] Fei Wu, Daniel S. Weld: Autonomously Semantifying Wikipedia. CIKM 2007

[21] Fei Wu, Daniel S. Weld: Automatically Refining the wikipedia Infobox Ontology. WWW 2008

[22] ChengXiang Zhai, John D. Lafferty: A risk minimization framework for information retrieval. Inf. Process. Manage. 42(1), 2006

[23] Qi Zhang, Fabian M. Suchanek, Lihua Yue, Gerhard Weikum: TOB: Timely Ontologies for Business Relations. WebDB 2008

[24] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang, Wei-Ying Ma: Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. KDD 2006

Webpage Understanding: Beyond Page-Level Search

Zaiqing Nie Ji-Rong Wen Wei-Ying Ma

Web Search & Mining Group
Microsoft Research Asia
Beijing, P. R. China
{znie, jrwen, wyma}@microsoft.com

Abstract

In this paper we introduce the webpage understanding problem which consists of three subtasks: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling. The problem is motivated by the search applications we have been working on including Microsoft Academic Search, Windows Live Product Search and Renlifang Entity Relationship Search. We believe that integrated webpage understanding will be an important direction for future research in Web mining.

1. Introduction

The World Wide Web is a vast and rapidly growing repository of information, and various kinds of valuable semantic information are embedded in webpages. Some basic understanding of the structure and the semantics of webpages could significantly improve people's browsing and searching experience.

We have been working on novel Web applications beyond page-level search with different levels of webpage understanding granularity. Specifically,

Block-based Search: We segment a webpage into semantic blocks and label the importance values of the blocks using a block importance model [11]. Then the semantic blocks, along with their importance values, are used to build block-based Web search engines [1][3].

Object-Level Vertical Search: We extract and integrate all the Web information about a real world object/entity and generate a pseudo page for this object. These object pseudo pages are indexed to answer user queries, and users can get integrated information about a real-world object in one stop, instead of browsing through a long list of pages. Our object-level vertical search technologies have been used to build Microsoft Academic Search (http://libra.msra.cn) and Windows Live Product Search (http://products.live.com).

Entity Relationship Search: We have deployed an Entity Relationship Search Engine in the China search market called *Renlifang* (http://renlifang.msra.cn). In Renlifang, users can query the system about people, locations, and organizations and explore their relationships. These entities and their relationships are automatically mined from the text content on the Web (more than 1 billion Chinese webpages).

As we can clearly see, large-scale Web mining (especially webpage understanding) plays a critical role in the above search technologies. In this paper, we first introduce these search applications in detail to motivate the webpage understanding tasks: webpage segmentation, webpage structure labeling, webpage text segmentation and labeling. Then we formally define the webpage understanding problem. Finally we present integrated statistical models for these webpage understanding tasks.

2. Beyond Page-Level Search

Nowadays, major commercial search engines take a webpage as the basic information unit and return a list of pages as search results to users. *Is a webpage the only or best atomic unit for information search on the Web?* We have developed several novel search technologies and systems, in ways going beyond the current page-level search paradigm.

2.1 Block-based Search

The content of a webpage is usually much more diverse compared with a traditional plain text document and encompasses multiple regions with unrelated topics. Moreover, for the purpose of browsing and publication, non-content materials, such as navigation bars, decoration fragments, interaction forms, copyright notices, and contact information, are usually embedded in webpages. Instead of treating a whole webpage as a unit of retrieval, we believe that the characteristics of webpages make passage a more effective mechanism for information retrieval.

In [2], we propose a VIPS (**VI**sion-based **P**age **S**egmentation) algorithm to segment a webpage into multiple semantic blocks. VIPS makes use of page layout features such as font, color, and size to segment a page. It first extracts all suitable nodes from the tag-tree of the page, and then finds the separators between these nodes. Here,

Figure 1. VIPS segmentation of a sample webpage

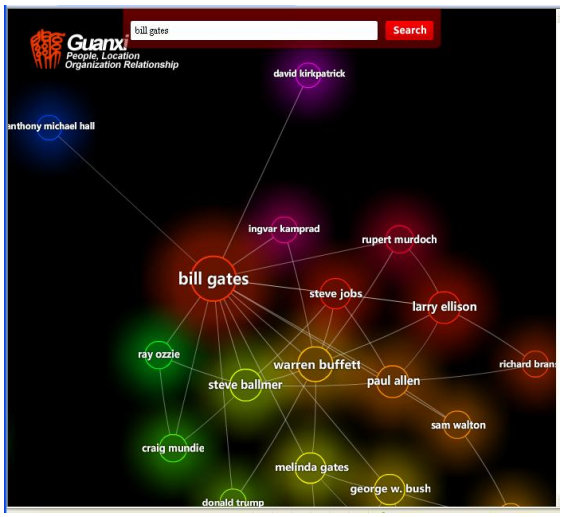


Figure 2. Automatically generated entity relationship graph for the query "Bill Gates" by our entity relationship search engine

separators denote the horizontal or vertical lines in a page that visually do not cross any node. Figure 1 shows the result of using VIPS to segment a sample CNN webpage.

After segmenting a webpage into semantic blocks, we compute the importance values of the blocks using a block importance model [11]. Then the semantic blocks, along with their importance values, are used to build block-based Web search engines with block-level link analysis [1] and block-based ranking [3] algorithms, and finally to improve the relevance of search results.

2.2 Object-Level Vertical Search

Much structured information about real-world objects is embedded in webpages and online databases. We explored a new paradigm to enable web search at the object level. We developed a set of technologies to automatically extract, integrate and rank structured Web objects [6][7][8][14], and then build powerful object-level vertical search engines for specific domains such as product search, academic search, and local search.

Information (e.g., attributes) about a web object is usually distributed in many web sources and within small segments of webpages. The task of an object extractor is to extract meta-data about a given type of objects from every webpage containing this type of objects. For example, for each crawled product webpage, we extract the *name, image*, *price* and *description* of each product from it using machine learning algorithms. If all of these product pages or just a portion of them are correctly extracted, we will have a huge collection of meta-data about real-world products that could be used for further knowledge discovery and query answering. Our statistical study on 51,000 randomly crawled webpages shows that about 12.6 percent are product pages. That is, there are about 1 billion product pages within a search index of 9 billion webpages.

However, how to extract product information from webpages generated by many (maybe tens of thousands of) different templates is non-trivial. One possible solution is that we first distinguish webpages generated by different templates, and then build an extractor for each template. We say that this type of solution is *template-dependent* (i.e., wrappers). However, accurately identifying webpages for each template is not a trivial task because even webpages from the same website may be generated by dozens of templates. Even if we can distinguish webpages, template-dependent methods are still impractical because the learning and maintenance of so many different extractors for different templates will require substantial human effort. By empirically studying webpages across websites about the same type of objects, we found strong template-independent features. The information about an object in a webpage is usually grouped together as an *object block*, as shown in Figure 4.

Using our Vision-based Page Segmentation (i.e., VIPS) algorithm and data record extraction technologies, we can automatically detect these object blocks, which are further segmented into atomic extraction units (i.e., HTML elements) called *object elements*. Each object element provides (partial) information about a single attribute of the web object. The web object extraction problem can be solved as a webpage structure labeling problem assuming we don't need to further segment the text content within the HTML elements [14].

With the Web object extraction and ranking technology, people can get integrated information about a real-world object in one stop, instead of browsing through a long list of pages. Our object-level vertical search technologies have been used to build Microsoft Academic Search and Windows Live Product Search. For more information about our object-level vertical search work, please refer to [6].

2.3 Entity Relationship Mining and Search

We have deployed an Entity Relationship Search Engine in the China search market called Renlifang. Currently Renlifang only serves in the Chinese language domain, and the knowledge is automatically mined from more than 1 billion crawled Chinese webpages.

Renlifang is a different kind of search engine, one that explores relationships between entities. In Renlifang, users can query the system about people, locations, and organizations and explore their relationships. These entities and their relationships are automatically mined from the text content on the Web. For each crawled webpage in Renlifang, the system extracts entity information and detects relationships, covering a spectrum of everyday individuals and well-known people, locations, or organizations.

Below we list the key features of Renlifang:

- **Entity Relationship Mining and Navigation**. Renlifang enables users to explore highly relevant information during searches to discover interesting relationships about entities associated with their query.
- **Expertise Finding**. For example, Renlifang could return a ranked list of people known for dancing or any other topic.
- **Web-Prominence Ranking**. Renlifang detects the popularity of an entity and enables users to browse entities in different categories ranked by their prominence on the Web during a given time period.
- **People Bio Ranking**. Renlifang ranks text blocks from webpages by the likelihood of being biography/description blocks.

Renlifang has been well received by Chinese Internet users and media with positive comments and millions of daily page-views in peak days. The English version of Renlifang is under development. In Figure 2, we show an automatically generated entity relationship graph using our English Renlifang prototype.

In entity relationship search, we need to extract the entity names and the related entities from both free text content and structured HTML elements within every webpage we crawled. So we need to use both page structure labeling and text segmentation and labeling technologies.

2.4 Categorization of Search Engines

Figure 3 shows a matrix about the categorization of search engines. Traditional web search engines can be classified as "page-level general search" engines, which simply crawl as many webpages as possible and treat each page as the basic

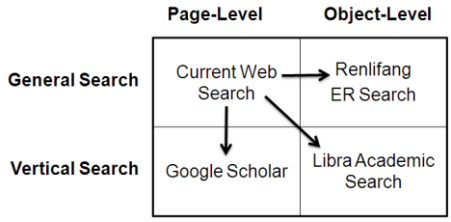


Figure 3. From Page-Level Search to Object-Level Search

retrieval unit. We see that there are two important trends towards next generation search engines. One trend is from general search to vertical search. We can restrict the search to a special domain and build page-level vertical search. For example, Google Scholar is an academic vertical search engine designed to help users to identify academic information. Another trend is from page-level to object-level. In some domains, data are more structured and uniform and it's relatively easy to define object schemas and conduct object extraction. Libra academic search is a typical application of object-level vertical search technologies. If we can automatically mine all types of entities and their relationships, we can build an object-level general search engine. This is actually the so-called Web Database dream, which aims to treat the whole Web as a huge database. The key here is to provide generic entity relationship mining and search technologies. Renlifang is a preliminary attempt towards this direction. For example, to extend Renlifang to support product relationship search, we only need to re-train the Named Entity Recognition model to optimize for product name extraction.

As we can see from the above search applications including block-based search, object-level vertical search and entity relationship search, some shallow understanding of the webpages will significantly improve users' browsing and searching experiences.

3. Problem Definition

In this section we define the webpage understanding problem which consists of three sub-tasks: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling.

3.1 Webpage Segmentation

To segment a webpage into semantically coherent units, the visual presentation of the page contains a lot of useful cues. Generally, a webpage designer would organize the content of a webpage to make it easy for reading. Thus, semantically coherent content is usually grouped together and the entire page is divided into regions for different content using explicit or implicit visual separators such as

Figure 4. A sample webpage with two similar data records

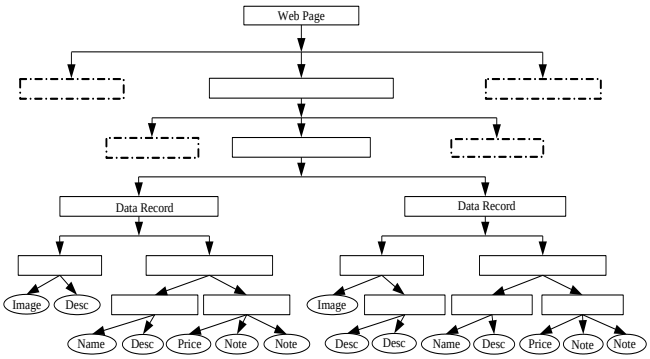


Figure 5. The vision-tree of the page in Figure 4.

lines, blank areas, images, font sizes, and colors [2]. Our goal is to derive this content structure from the visual presentation of a webpage.

We formally define the webpage segmentation problem below.

Definition 3.1 (Webpage Segmentation)*: Given a webpage, webpage segmentation is the task of partitioning the page at the semantic level and constructing a vision-tree for the page. Each node in the vision-tree will correspond to a block of coherent content in the original page (See Figure 1 for an example).*

Based on the definition, the output of webpage segmentation is the vision-tree of a webpage. Each node on this vision-tree represents a data region in the webpage, which is called a *block*. The root block represents the whole page. Each inner block is the aggregation of all its child blocks. All leaf blocks are atomic units (*i.e.*, elements) and form a flat segmentation of the webpage. Since vision-tree can effectively keep related content together while separating semantically different blocks from one another, we use it as the data representation format of the webpage segmentation results. Figure 5 is a vision-tree for the page in Figure 4, where we use rectangles to denote the inner blocks and use ellipses to denote the leaf blocks (or elements). Due to space limitations, the blocks denoted by dotted rectangles are not fully expanded.

3.2 Webpage Structure Labeling

After webpage segmentation, we will have a vision-tree representation of a webpage keeping semantically coherent content together as web blocks. The webpage structure labeling task is to assign semantic labels to the blocks on a webpage (i.e., nodes on vision-tree). For different applications, the semantic label space could be different. For example,

- For Web object extraction, the label space consists of a label called Object Block and several labels corresponding to the individual attribute names of the object (for example, the *name*, *image*, *price* and *description* of a product for sale). The web object extraction problem can be solved as a webpage structure labeling problem assuming we don't need to further segment the HTML elements which are the leaf nodes of the vision-tree [14].
- For the webpage main block detection application, the label space could consist of the following: Main Block, Navigation Bar, Copyright, Advertisement, etc.

Below we define the webpage structure labeling problem.

Definition 3.2 (Webpage Structure Labeling)*: Give a vision-tree of a page, let* $\mathrm{x}=\{\mathrm{x}_0,\mathrm{x}_1,\cdots,\mathrm{x}_N\}$ *be the features of all the blocks and each component* x_i *is a feature vector of one block, and let* $\mathrm{y}=\{\mathrm{y}_0,\mathrm{y}_1,\cdots,\mathrm{y}_N\}$ *be one possible label assignment of the corresponding blocks. The goal of webpage structure labeling is to compute maximum a posteriori (MAP) probability of* y *:*

$$\mathrm{y}^* = \arg\max p(\mathrm{y}\,|\,\mathrm{x})$$

3.3 Webpage Text Segmentation and Labeling

The page segmentation task segments a webpage into blocks and constructs a vision-tree for the webpage, and then the webpage structure labeling task detects the object blocks and labels HTML elements on the vision-tree using attribute names. However, how to effectively segment and label the text content inside HTML elements is still an open problem. As we pointed out in section 2, text segmentation and labeling are critical for entity relationship search engines which need to extract entity information from both the free text content and structured blocks of billions of crawled webpages. Since much of the text content on a webpage is often text fragments and not strictly grammatical, traditional natural language processing techniques that typically expect grammatical sentences, are no longer directly applicable.

In Figure 6, we show an example webpage containing local entity information. As we can see, the address information

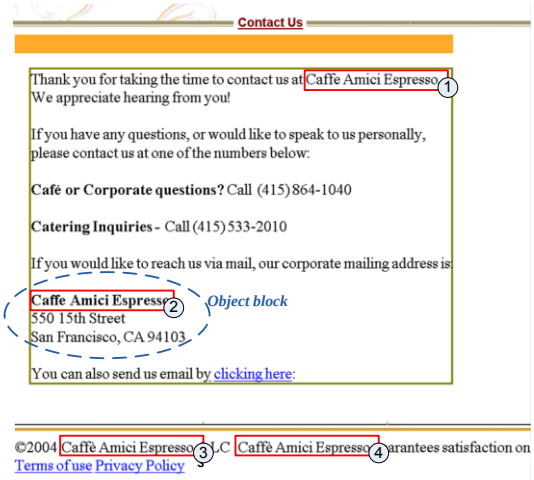


Figure 6. An example webpage containing local objects

of the local business on the webpage is regularly formatted in a visually structured block: the first line of the block contains the business name with bold font; the second line contains the street information; the third line contains the city, state and zip code. As we can see, the attributes in an object block are always short strings. It is quite difficult to identify business names correctly only with the structure (i.e., visual layout) information and text features of these short strings (e.g., regular expression and word emission probability). For example, there is not much evidence for "Caffe Amici Espresso" to be labeled as the business name in the object block shown in Figure 6. Fortunately, the business name is mentioned not only in the object block but also in the natural language sentences outside the object block, such as "Thank you for taking the time to contact us at Caffe Amici Espresso" and "Caffe Amici Espresso guarantees satisfaction on all products we sell".

We believe that if we could collect more information about an object, we can make better decisions on it. For example, it would be much more accurate and easier if we could label all the mentions of the business name "Caffe Amici Espresso" together, no matter where it appeared in the webpage: object blocks or natural language sentences.

Below we define the webpage text segmentation and labeling problem.

Definition 3.3 (Webpage Text Segmentation and Labeling): *Given a vision-tree of a page, let* $\mathrm{x} = \{\mathrm{x}_0, \mathrm{x}_1, \cdots, \mathrm{x}_N\}$ *be the features of all the word occurrences on the tree and each component* x_i *is a feature vector of one word occurrence. The goal of webpage text segmentation and labeling is to find the optimization segmentation and labeling* ***S*****:*

$$\mathbf{S}^* = \arg\max_{\mathbf{S}} p(\mathbf{S} \mid \mathbf{X})$$

4. Models for Webpage Understanding

In this section, we introduce models/algorithms for webpage understanding subtasks: webpage segmentation, structure labeling, and web text segmentation and labeling. In particular, in Section 4.4, we argue that joint optimization of the subtasks significantly improves the performance of the individual subtasks.

4.1 Webpage Segmentation

An intuitive way to segment a page is based on the layout of a webpage. This way, a webpage is generally separated into 5 regions: top, down, left, right and center [5]. The drawback of this method is that such a layout template cannot fit into all webpages. Furthermore, the segmentation is too rough to exhibit semantic coherence.

Compared with the above segmentation, Vision-based Page Segmentation (VIPS) excels in both an appropriate partition granularity and a coherent semantic aggregation. By detecting useful visual cues based on DOM structure, a tree-like vision-based content structure of a webpage is obtained. The granularity is controlled by *a degree of coherence* (DOC) which indicates how coherent each block is. VIPS can efficiently keep related content together while separating semantically different blocks from each other. Each block in VIPS is represented as a node in a tree. The root is the whole page; inner nodes are the top level coarser blocks; child nodes are obtained by partitioning the parent node into finer blocks; all leaf nodes consist of a flat segmentation of a webpage with an appropriate coherent degree. The stopping of the VIPS algorithm is controlled by a predefined DOC (PDOC), which plays a role as a threshold to indicate the finest granularity that we are satisfied with [2]. The segmentation only stops when the DOCs of all blocks are no smaller than the PDOC.

However, if we combine webpage segmentation and structure labeling together, we don't need to guess the predefined DOC to segment a webpage into blocks with a satisfied degree of granularity. The leaf nodes of the vision-tree could be the HTML elements, and we just need to assign labels to the nodes of the tree to detect the blocks we are interested in.

4.2 Webpage Structure Labeling

After the webpage segmentation task, a webpage is represented as a vision-tree, and the webpage structure labeling task becomes the task of assigning labels to the nodes on a vision-tree. We introduce a probabilistic model called Hierarchical Conditional Random Field (HCRF) model for webpage structure labeling.

For the page in Figure 4, the HCRF model is shown in Figure 7, where we also use rectangles to denote inner nodes and use ovals to denote leaf nodes. The dotted rectangles are for the blocks that are not fully expanded. Each node on the graph is associated with a random

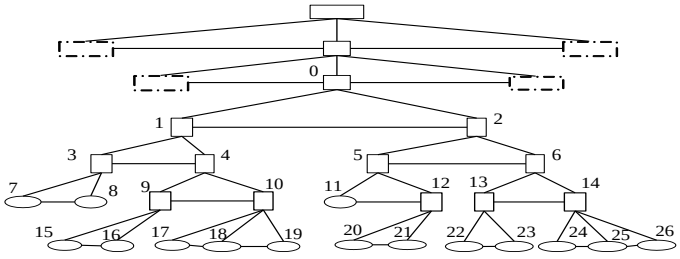


Figure 7. The HCRF model for the page in Figure 4.

variable Y_i. We currently model the interactions of sibling variables via a linear-chain, although more complex structure such as two-dimensional grid can also be used.

As a conditional model, HCRF can efficiently incorporate any useful features for webpage structure labeling. By incorporating hierarchical interactions, HCRF could incorporate long distance dependencies and achieve promising results [14].

4.3 Webpage Text Segmentation and Labeling

The existing work on text processing cannot be directly applied to web text understanding. This is because the text content on webpages is often not as regular as those in natural language documents and many of them are less grammatical text fragments. One possible method of using NLP techniques for web text understanding is to first manually or automatically identify logically coherent data blocks, and then concatenate the text fragments within each block into one string via some pre-defined ordering method. The concatenated strings are finally put into a text processing method, such as CRYSTAL [10] or Semi-CRF [9], to identify target information. [4][10] are two attempts in this direction.

It is natural to leverage the webpage structure labeling results to first concatenate the text fragments within the blocks generated by VIPS, and then use Semi-CRF to process the concatenated strings with the help of structure labeling results. However it would be more effective if we could jointly optimize the structure labeling task and the text segmentation and labeling task together.

4.4 Integrated Webpage Understanding

Now we have introduced the three subtasks of webpage understanding: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling. We argue that we need a unified model to jointly optimize these webpage understanding tasks. This is because with more semantic understanding of the text tokens we could perform better structure labeling, and with better structure labeling we can perform better page segmentation, and vice versa.

We don't have a unified model to integrate all the three subtasks yet, but we have done some initial work to jointly optimize two subtasks. Our recent work [15] shows that the joint optimization of webpage segmentation and structure labeling tasks improves the performance of both tasks, and we will introduce another recent work [12] on integrated webpage structure labeling and text segmentation and labeling below.

Given the data representation of the page structure and text strings, we can define the joint optimization problem formally as follows.

Definition 4.1 (Joint Optimization of Structure Understanding and Text Segmentation and Labeling): *Given a vision tree X, the goal of joint optimization of structure understanding and text segmentation and labeling is to find the optimal assignment of the node labels and text segmentations* ***(H, S)*****:*

$$(\mathbf{H},\mathbf{S})^{*} = \arg\max_{(\mathbf{S},\mathbf{H})} p(\mathbf{H},\mathbf{S} \mid \mathbf{X}) \cdot$$

Here, all the segmentation and labels of the leaf nodes on the vision tree are denoted as $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2 \ldots \mathbf{s}_i \ldots \mathbf{s}_{|\mathbf{S}|}\}$, and $\mathbf{H} = \{h_1, h_2 \ldots h_i \ldots h_{|\mathbf{X}|}\}$ represents the labels of the nodes on the vision-tree **X.**

This problem is too hard because the searching space is the Cartesian product of both label spaces. Fortunately, the negative logarithm of the posterior will be a convex function, if we use the exponential function as the potential function. Then we can use the coordinate-wise optimization to optimize **H** and **S** iteratively. In this manner, we can solve two simpler conditional optimization problems instead, i.e., we perform the structure understanding and text understanding separately and iteratively. As we introduced before, the state-of-the-art models for structure understanding and text understating are HCRF and Semi-CRF respectively. However, we need to make them interact with each other. Therefore, we extend them by introducing additional parameters. We extend the HCRF model by introducing text segment feature functions with the segmentation of the text strings as their input. The Semi-CRF model is extended by introducing both the label of the corresponding node on the vision-tree and the segmentation results over all the nodes on the vision-tree in the last iteration.

We evaluated the performance of the joint optimization using a local entity extraction task. The extraction results show that the accuracy of all the attributes of the joint optimization model are significantly better than optimizing webpage structure labeling and text segmentation and labeling separately.

5. CONCLUSION

Internet search engines process billions of webpages on a weekly basis, and the text content of these webpages are indexed to answer user queries. We believe that some shallow understanding of the webpages will significantly improve users' browsing and searching experiences. In this paper, we formally define the webpage understanding problem, which consists of three subtasks: webpage segmentation, webpage structure labeling, and webpage text segmentation and labeling. The solutions to the problem have to be template-independent because of its web-scale nature. In this paper, we introduce partially integrated statistical models for these webpage understanding tasks. However we believe that fully integrated webpage understanding models will be an important direction for future research in Web mining for search applications.

6. REFERENCES

[1] Deng Cai, Xiaofei He, Ji-Rong Wen, and Wei-Ying Ma. Block-Level Link Analysis. In Proceedings of SIGIR, 2004.

[2] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma. VIPS: a Vision-based Page Segmentation Algorithm. Microsoft Technical Report, MSR-TR-2003-79, 2003.

[3] Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Ying Ma. Block-based Web Search. In Proceedings of SIGIR, 2004.

[4] D. DiPasquo. Using HTML Formatting to Aid in Natural Language Processing on the World Wide Web. Senior Honors Thesis, Carnegie Mellon University, 1998.

[5] M. Kovacevic, M. Diligenti, M. Gori and V. Milutinovic. Recognition of Common Areas in a Webpage Using Visual Information: a possible application in a page classification. ICDM, 2002.

[6] Zaiqing Nie, Ji-Rong Wen and Wei-Ying Ma. Object-Level Vertical Search. Proc. of CIDR, 2007.

[7] Zaiqing Nie, Yunxiao Ma, Shuming Shi, Ji-Rong Wen, Wei-Ying Ma. Web Object Retrieval. In Proc. of WWW, 2007.

[8] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen and Wei-Ying Ma. Object-Level Ranking: Bringing Order to Web Objects. In Proc. of WWW, 2005.

[9] S. Sarawagi and W. W. Cohen. Semi-Markov. Conditional Random Fields for Information Extraction. Proc. of NIPS, 2004.

[10] S. Soderland. Learning to Extract Text-based Information from the World Wide Web. Proc. of SIGKDD, 1997.

[11] Ruihua Song, Haifeng Liu, Ji-Rong Wen, and Wei-Ying Ma. Learning Block Importance Models for Webpages. In *Proc. of WWW*, 2004.

[12] Chunyu Yang, Yong Cao, Zaiqing Nie, Jie Zhou, Ji-Rong Wen. Closing the Loop in Webpage Understanding. Proc. of CIKM, 2008.

[13] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang and Hsiao-Wuen Hon. Webpage Understanding: An Integrated Approach. Proc. of SIGKDD, 2007.

[14] Jun Zhu, Zaiqing Nie, Ji-Rong Wen, Bo Zhang and Wei-Ying Ma. Simultaneous Record Detection and Attribute Labeling in Web Data Extraction. Proc. of SIGKDD, 2006.

[15] Jun Zhu, Zaiqing Nie, Bo Zhang and Ji-Rong Wen. Dynamic Hierarchical Markov Random Fields for Integrated Web Data Extraction. Journal of Machine Learning Research. 9(Jul): 1583--1614, 2008.

Web-Scale Extraction of Structured Data

Michael J. Cafarella
University of Washington
mjc@cs.washington.edu

Jayant Madhavan
Google Inc.
jayant@google.com

Alon Halevy
Google Inc.
halevy@google.com

ABSTRACT

A long-standing goal of Web research has been to construct a unified Web knowledge base. Information extraction techniques have shown good results on Web inputs, but even most domain-independent ones are not appropriate for Web-scale operation. In this paper we describe three recent extraction systems that can be operated on the entire Web (two of which come from Google Research). The TEXTRUNNER system focuses on raw natural language text, the WEBTABLES system focuses on HTML-embedded tables, and the deep-web surfacing system focuses on "hidden" databases. The domain, expressiveness, and accuracy of extracted data can depend strongly on its source extractor; we describe differences in the characteristics of data produced by the three extractors. Finally, we discuss a series of unique data applications (some of which have already been prototyped) that are enabled by aggregating extracted Web information.

1. INTRODUCTION

Since the inception of the Web, the holy grail of web information extraction has been to create a knowledge base of all facts represented on the Web. Even if such a knowledge base were imperfect (and it certainly would be), its contents can be used for a variety of purposes, including answering factual queries (possibly performing simple inferences), expansion of keyword queries to improve recall, and assisting more specific extraction efforts. The vast majority of the work on information extraction has focused on more specific tasks, either by limiting the extraction to particular domains (e.g., extracting seminar announcement data from email in an academic department, or extracting corporate intelligence from news articles), or training extractors that apply to specific web sites. As such, these techniques cannot be applied to the Web as a whole.

In this paper we describe three extraction systems that began with the goal of being domain and site independent, and therefore apply to the entire Web. The systems target different kinds of resources on the Web: text, HTML tables and databases behind forms. Each of these systems has extracted a portion of what can some day become the unified Web knowledge base, and teaches us important lessons on the way to that goal. The first project is the TEXTRUNNER system [3], which operates on very large amounts of unstructured text, making very few assumptions about its target data. The second project is the more recent WEBTABLES system [8, 9], which extracts relational tables from HTML structures. The third project is the deep-web crawling project [21], which surfaces contents of backend databases that are accessible via HTML forms. We describe how the data extracted from each of these systems differ in content and the types of techniques that needed to be applied in each case. We also describe how each system contributes to computing a set of entities and relationships that are represented on the Web.

Finally, we look beyond the specific goal of creating a knowledge base of the Web and consider what kind of *semantic services* can be created in order to assist with a wide collection of tasks. We argue that by *aggregating* vast amounts of structured data on the Web we can create valuable services such as synonym finding, schema auto-complete and type prediction. These services are complimentary to creating a knowledge base of the Web and can be created even if the knowledge base is still under construction. We show early examples of such services that we created in our recent work.

2. TEXTUAL EXTRACTION

The first extraction system that we consider operates over very large amounts of unstructured text. Banko et al.'s TEXTRUNNER consumes text from a Web crawl and emits n-ary tuples [3]. It works by first linguistically parsing each natural language sentence in a crawl, then using the results to obtain several candidate tuple extractions. For example, TEXTRUNNER might process the sentence "Albert Einstein was born in 1879", find two noun phrases and a linking verb phrase, then create the tuple ⟨`Einstein, 1879`⟩ in the `was_born_in` relation. Finally, TEXTRUNNER applies extraction-frequency-based techniques to determine whether the extraction is accurate. Table 1 lists a few example TEXTRUNNER extractions.

There are a number of extractors that emit tuples from raw text inputs, including DIPRE, SNOWBALL, and KNOWITALL [1, 7, 14]. However, TEXTRUNNER has two additional unusual qualities that make it es-

Object 1	Relation	Object 2
einstein	discovered	relativity
1848	was_year_of	revolution
edison	invented	phonograph
einstein	died_in	1955

Table 1: A few example binary tuple extractions from TextRunner.

pecially apt for creating a Web-scale knowledge base. First, TextRunner was designed to operate in *batch* mode, consuming an entire crawl at once and emitting a large amount of data. In contrast, other systems have been on-demand query-driven systems that choose pages based on the user's revealed interest. The on-demand approach may seem more appealing because it promises to only perform relevant work, and because the system may be able to use the query to improve the output quality. However, the batch-oriented technique allows us to pre-compute good extractions before any queries arrive, and then index these extractions aggressively. In the query-driven approach, all of this work (possibly including downloading the text itself) must happen at query-time. Also, note that traditional search engines follow a similar batch-processing approach.

Second, TextRunner does not attempt to populate a given target relation or schema, but rather discovers them during processing. KnowItAll, Dipre, and Snowball all require some sort of target guidance from the user (e.g., a set of seed pairs or a set of extraction phrases). This kind of *open information extraction* is necessary if we want to extract the Web's worth of data without a strong model of Web contents (in the form of either a query load or perhaps some kind of general ontology). Further, this approach allows the extractor to automatically obtain brand-new relations as they appear over time.

This system has been tested in two large deployments. The initial TextRunner project ran on a general corpus of 9 million Web pages, extracting 1 million concrete tuples (of which 88% were accurate) [3]. In a more recent experiment, TextRunner was run on a much larger 500M page crawl, yielding more than 200M tuples that occurred at least two times (though detailed accuracy figures are not yet available) [2]. H-CRF is an extension to the TextRunner system that uses an approach based on conditional random fields to improve output quality [4].

TextRunner's open IE technique differs substantially from previous work. For example, the Dipre and Snowball projects accept a set of binary seed tuples, learn patterns that textually connect the two elements, then extend the set by downloading relevant Web pages and applying the learned patterns. The KnowItAll system extracts binary *is-a* relations (and in later work, other *n-ary* relations) from downloaded Web pages [14]. It worked by applying a handful of generic extraction phrases (due to Hearst [17]) like "X such as Y" and "X, including Y,..." to the pages, then distinguishing true from spurious extractions with a trained classifier that took as input frequency-statistics of the extraction phrases.

Several other systems occupy an interesting middle ground between raw text and structured data. The Yago system produced ontologies consisting of hypernyms and binary relations, extracted from WordNet and the structured parts of Wikipedia (such as the specialized "list pages" that put other Wikipedia entities into broad sets) [24]. It did not extract information from the Wikipedia article content. The Kylin system used Wikipedia infoboxes (essentially, small lists of attribute/value pairs associated with a single entity) to train textual extractors that were then applied to the article text [27]. The primary goal of the system was to make the infobox data more complete, especially for versions of Wikipedia in unpopular languages.

The data that TextRunner and other text-centric extractors emit differs in several ways from traditional relational-style data. First, the cardinality of text-derived tuples is heavily reliant on extractable natural language constructs: binary relations (generally two nouns and a linking relation) are much easier to extract and are more common than 3-ary or larger tuples. In contrast, relational tables make it easy to express data with many dimensions.

Second, the domains that appear in text-derived data will be different from data found in relations. In part, this difference is due to the above point about binary tuples being more "natural" than other tuple sizes in natural language text. But this difference in domains is also due to issues of human reading and writing style: it would be very boring to read a piece of text that exhaustively enumerates movie times, but scanning a table is quite easy.

Third, unlike the WebTables extractor we describe next, where each extraction is based on a *single* occurrence of a table on the Web, the TextRunner (as well as the KnowItAll) extractor outputs extractions that are found in *multiple* locations on the Web. This is necessary since the extractors that TextRunner relies on are linguistic extractors that are quite fallible. In contrast, the tables extracted by WebTables are "cleaner" after extraction than text-based tuples. Further, because an extracted relation is a fairly complicated object consisting of multiple tuples, it is harder to find exact replicas elsewhere on the Web and so frequency-based techniques are not obviously applicable.

Finally, the output data model from TextRunner differs slightly from the relational model. The individual dimensions of a TextRunner extraction are generally not labeled (e.g., we do not know that `Einstein` is in the `human` attribute, and that `1879` is in `birthyear`). But most relations (and extractions from the "hard-coded" text systems) do have this kind of attribute metadata. The situation is reversed in the case of the name of the relation. The `was_born_in` extraction serves as a rough name for a relation in which the

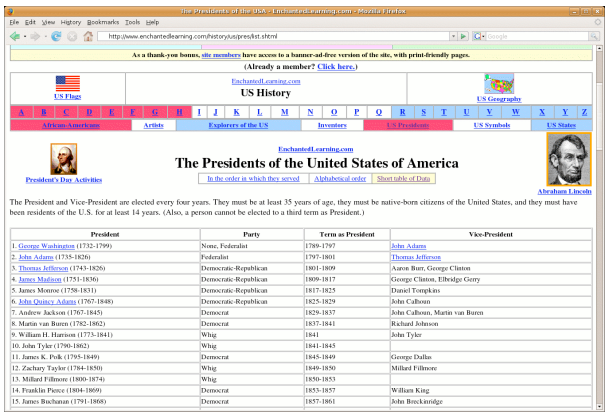


Figure 1: Example HTML Table.

derived tuple belongs to, but relational tables on the Web do not usually offer their relation name.

3. THE RELATIONAL WEB

The WebTables system [8, 9] is designed to extract structured data from the Web that is expressed using the HTML `table` tag. For example, the Web page shown in Figure 1 contains a table that lists American presidents[1]. The table has four columns, each with a domain-specific label and type (e.g., **President** is a person name, **Term as President** is a date range, etc) and there is a tuple of data for each row. Even though much of its metadata is implicit, this Web page essentially contains a small relational database that anyone can crawl.

Of course, not all `table` tags carry relational data. A huge number are used for page layout, for calendars, or other non-relational purposes. For example, in Figure 1, the top of the page contains a `table` tag used to lay out a navigation bar with the letters A-Z. Based on a human-judged sample of several thousand raw tables, we estimate that our general Web crawl of 14.1B tables contains about 154M true relational databases, or about 1.1% of the total. While the percentage is fairly small, the vast number of tables on the Web means that the total number of relations is still enormous. Indeed, the relational databases in our crawl form the largest database corpus we know of, by five orders of decimal magnitude.

Unfortunately, distinguishing a *relational* table from a *non-relational* one can be difficult to do automatically. Obtaining a set of good relational tables from a crawl can be considered a form of *open information extraction*, but instead of raw unstructured text the extractor consumes (messy) structured inputs. Our WebTables system uses a combination of hand-written and statistically-trained classifiers to recover the relational tables from the overall set of HTML tables. After running the resulting classifiers on a general Web crawl, WebTables obtains a huge corpus of structured materialized relational databases. These databases are a very useful source of information for the unified Web knowledge base.

Recovering relational databases from the raw HTML tables consists of two steps. First, WebTables attempts to filter out all the non-relational tables. Second, for all the tables that we believe to be relational, WebTables attempts to recover *metadata* for each.

WebTables executes the following steps when filtering out relational tables:

Step 1. Throw out *obviously* non-relational HTML tables, such as those consisting of a single row or a single column. We also remove tables that are used to display calendars or used for HTML form layout. We can detect all of these cases with simple hand-written detectors.

Step 2. Label each remaining tables as *relational* or *non-relational* using a trained statistical classifier. The classifier bases its decision on a set of hand-written table features that seem to indicate a table's type: the number of rows, the number of columns, the number of empty cells, the number of columns with numeric-only data, etc.

Step 1 of this process removes more than 89% of the total table set. The remaining HTML tables are trickier. Traditional schema tests such as constraint checking are not appropriate in a messy Web context, so our test for whether a table is *relational* is necessarily somewhat subjective. Based on a human-judged sample of several thousand examples, we believe a little more than 10% of the remaining tables should be considered relational.

We used a portion of this human-emitted sample to train the classifier in Step 2. That step's output (and hence, the output of the entire filtering pipeline) is an imperfect but still useful corpus of databases. The output retains 81% of the truly relational databases in the input corpus, though only 41% of the output is relational. That means WebTables emits a database of 271M relations, which includes 125M of the raw input's estimated 154M true relations (and, therefore, also includes 146M false ones).

After relational filtering, WebTables tries to recover each relation's metadata. Because we do not recover multi-table databases, and because many traditional database constraints (e.g., key constraints) cannot be expressed using the `table` tag, our target metadata is fairly modest. It simply consists of a set of labels (one for each column) that is found in the table's first row. For example, the table from Figure 1 has metadata that consists of `President`, `Party`, `Term as President`, and `Vice President`.

Because table columns are not explicitly typed, the metadata row can be difficult to distinguish from a table's data contents. To recover the metadata, we used a second trained statistical classifier that takes a table and emits a *hasmetadata*/ *nometadata* decision. It uses features that are meant to "reconstruct" type information for the table: the number of columns that have non-string data in the first row versus in the table's body, and various tests on string-length meant to detect whether the first value in a column is drawn from

[1] `http://www.enchantedlearning.com/history/us/pres/list.shtml`

the same distribution as the body of the column.

A human-marked sample of the relational filtering output indicates that about 71% of all true relations have metadata. Our metadata-recovery classifier performs well: it achieves precision of 89% and recall of 85%.

The materialized relational tables that we obtain from WebTables are both relatively clean and very expressive. First, the tabular structure makes individual elements easy to extract, and hence provides a very rich collection of entities for the Web knowledge base. Other data sources (such as the text collections used in TextRunner) require that we demarcate entities, relations, and values in a sea of surrounding text. In contrast, a good relational table explicitly declares most label endpoints through use of the table structure itself. Further, a relational table carries many pieces of metadata at once: each tuple has a series of dimensions that usually have attribute labels, and the mere presence of tuples in a table indicate set membership. However, unlike a TextRunner tuple, the label of the relation is rarely explicit. Finally, unlike the HTML form interfaces to deep-web databases (described in the next section), but like tuples derived by TextRunner, all of the dimensions in a relational table can be queried directly (forms may allow queries on only a subset of the emitted data's attributes).

Interestingly, in [25] it is shown that tables extracted by WebTables can be used to successfully expand instance sets that were originally extracted from free text using techniques similar to those in TextRunner. We also have initial results indicating that entities in the extracted tables can be used to improve the segmentation of HTML lists – another source for structured data on the Web.

4. ACCESSING DEEP-WEB DATABASES

Not all the structured data on the Web is published in easily-accessible HTML tables. Large volumes of data stored in backend databases are often made available to web users only through HTML form interfaces. For example, US census data can be retrieved by zipcode using the HTML form on the US census website[2]. Users retrieve data by performing valid form submissions. HTML forms either pose structured queries over relational databases or keyword queries over text databases, and the retrieved contents are published in structured templates, e.g., HTML tables, on web pages.

While the tables harvested by WebTables can potentially be reached by users by posing keyword queries on search engines, the contents behind HTML forms were for a long time believed to be beyond the reach of search engines – there are not many hyperlinks pointing to web pages that are results of form submissions and web crawlers did not have the ability to automatically fill out forms. Hence, the names Deep, Hidden, or Invisible Web have been used to collectively refer to the contents accessible only through forms. It has been speculated that the data in the Deep Web far exceeds that currently indexed by search engines [6, 16]. We estimate that there are at least 10 million potentially useful forms [20].

Our primary goal was to make the data in deep-web databases more easily accessible to search engine users. Our approach has been to *surface* the contents into web pages that can then be indexed by search engines (like any other web page). As in the cases of TextRunner and WebTables our goal was to develop techniques that would apply efficiently on large numbers of forms. This is in contrast with much prior work that have either addressed the problem by constructing mediator systems one domain at a time [12, 13, 26], or have needed site-specific wrappers or extractors to extract documents from text databases [5, 22]. As we discuss, the pages we surface contain tables from which additional data can be extracted for the Web knowledge base.

Over the past few years we have developed and deployed a system that has surfaced the contents of over a million of such databases, which span over 50 languages and over 100 domains. The surfaced pages contribute results to over a thousand web search queries per second on Google.com. In the rest of this section, we present an overview of our system. Further details about our system can be found in [21].

4.1 Surfacing Deep-Web databases

There are two complementary approaches to offering access to deep-web databases. The first approach, essentially a data integration solution, is to create vertical search engines for specific domains (e.g., cars, books, real-estate). In this approach we could create a mediator form for the domain at hand and semantic mappings between individual data sources and the mediator form. At web-scale, this approach suffers from several drawbacks. First, the cost of building and maintaining the mediator forms and the mappings is high. Second, it is extremely challenging to identify the domain (and the forms within the domain) that are relevant to a keyword query. Finally, data on the web is about everything and domain boundaries are not clearly definable, not to mention the many different languages – creating a mediated schema of everything will be an epic challenge, if at possible.

The second approach, surfacing, pre-computes the most relevant form submissions for all interesting HTML forms. The URLs resulting from these submissions can then be indexed like any other HTML page. Importantly, this approach leverages the existing search engine infrastructure and hence allows the seamless inclusion of Deep-Web pages into web search results; we thus prefer the surfacing approach.

The primary challenge in developing a surfacing approach lies in pre-computing the set of form submissions for any given form. First, values have to be selected for each input in the form. Value selection is trivial for select menus, but is very challenging for text boxes. Second, forms have multiple inputs and using a simple

[2] http://factfinder.census.gov/

strategy of enumerating all possible form submissions can be very wasteful. For example, the search form on cars.com has 5 inputs and a Cartesian product will yield over 200 million URLs, even though cars.com has only 650,000 cars on sale [11]. We present an overview of how we address these challenges in the rest of this section.

An overarching challenge in developing our solution was to make it scale and be domain independent. As already mentioned, there are millions of potentially useful forms on the Web. Given a particular form, it might be possible for a human expert to determine through laborious analysis the best possible submissions for that form, but such a solution would not scale. Our goal was to find a completely automated solution that can be applied to any form in any language or domain.

Selecting Input Values: A large number of forms have text box inputs and require valid inputs values for any data to be retrieved. Therefore, the system needs to choose a good set of values to submit in order to surface the most useful result pages. Interestingly, we found that it is not necessary to have a complete understanding of the semantics of the form in order to determine good candidate values for text inputs. We note that text inputs fall in one of two categories: generic search inputs that accept most keywords and typed text inputs that only accept values in a particular domain.

For search boxes, we start by making an initial prediction for good candidate keywords by analyzing the text on pages from that the form site that might be already indexed by the search engine. We use the initial set of keywords to bootstrap an iterative probing process. We test the form with candidate keywords and when valid form submissions result, we extract more keywords from the resulting pages. This iterative process continues until either new candidate keywords cannot be extracted or a pre-specified target is reached. The set of all candidate keywords can then be pruned to select a smaller subset that ensures diversity of the exposed database contents. Similar iterative probing approaches have been used in the past to extract text documents from specific databases [5, 10, 18, 22].

For typed text boxes, we attempt to match the type of the text box against a library of types that are extremely common across domains, e.g., zip codes in the US. Note that probing with values of the wrong type results in invalid submissions or pages with no results. We observed that even a library of just a few types can cover a significant number of text boxes.

Selecting Input Combinations: For HTML forms with more than one input, a simple strategy of enumerating the entire Cartesian product of all possible inputs will result in a very large number of URLs being generated. Crawling too many URLs will drain the resources of a search engine web crawler while also posing an unreasonable load on web servers hosting the HTML forms. Interestingly, when the Cartesian product is very large, it is likely that a large number of the form submissions result in empty result sets that are useless from an indexing standpoint.

To only generate a subset of the Cartesian product, we developed an algorithm that intelligently traverses the search space of possible input combinations to identify only the subset of input combinations that are likely to be useful to the search engine index. We introduced the notion of an input template: given a set of *binding* inputs, the template represents that set of all form submissions using only the Cartesian product of values for the binding inputs. We showed that only input templates that are *informative*, i.e., generate web pages with sufficiently distinct retrieved contents, are useful to a search engine index. Hence, given a form, our algorithm searches for informative templates in the form, and only generates form submissions from them.

Based on the algorithms outlined above, we find that we only generate a few hundred form submissions per form. Furthermore, we believe the number of form submissions we generate is proportional to the size of the database underlying the form site, rather than the number of inputs and input combinations in the form. In [21], we also show that we are able to extract large fractions of underlying deep-web databases automatically (without any human supervision) using only a small number of form submissions.

4.2 Extracting the extracted databases

By creating web pages, surfacing does not preserve the structure and hence the semantics of the data exposed from the underlying deep-web databases. The loss in semantics is also a lost opportunity for query answering. For example, suppose a user were to search for "used ford focus 1993". Suppose there is a surfaced used-car listing page for Honda Civics, which has a 1993 Honda Civic for sale, but with a remark "has better mileage than the Ford Focus". A simple IR index can very well consider such a surfaced web page a good result. Such a scenario can be avoided if the surfaced page had the annotation that the page was for used-car listings of Honda Civics and the search engine were able to exploit such annotations. Hence, the challenge in the future will be to find the right kind of annotation that can be used by the IR-style index most effectively.

When contents from a deep-web database are surfaced onto a web page, they are often laid out into HTML tables. Thus, a side-effect of surfacing is that there are potentially many more HTML tables that can then be recovered by a system like WebTables. In fact, HTML tables generated from the deep-web can potentially be recovered in a more informed way than in general. Surfacing leads to not one, but multiple pages with overlapping data laid out in identical tables. Recovering the tables collectively, rather than each one separately, can potentially lead to a complete extraction of the the deep-web database. In addition, mappings can be predicted from inputs in the form to columns in the recovered tables thereby resulting in recovering more of the semantics of the underlying data. Such deeper extraction and recovery is an area of future work.

In addition, the forms themselves contribute to interesting structured data. The different forms in a do-

main, just like different tables, are alternate representations for metadata within a domain. The forms can be aggregated and analyzed to yield interesting artifacts. Resources such as the ACSDb [9] can be constructed based on the metadata in forms. For example, as works such as [15, 23, 28] have shown, mediated schemas can be constructed for domains by clustering forms belonging to that domain. Likewise, form matching within a domain can be improved by exploiting other forms in the same same domain [19].

5. WEB-SCALE AGGREGATES

So far our discussion has focused on a knowledge base of facts from the Web. However, our experience has shown that significant value can be obtained from analyzing collections of metadata on the Web. Specifically, from the collections we have been working with (forms and HTML tables) we can extract several artifacts, such as: (1) a collection of forms (input names that appear together, values for select menus associated with input names), (2) a collection of several million schemata for tables, i.e., sets of column names that appear together, and (3) a collection of columns, each having values in the same domain (e.g., city names, zipcodes, car makes).

We postulate that we can build from these artifacts a set of *semantic services* that are useful for many *other* tasks. In fact, these are some of the services we would expect to build from a Web knowledge base, but we argue that we do not need a complete Web knowledge base in order to do so. Examples of such services include:

- Given an attribute name (and possibly values for its column or attribute names of surrounding columns) return a set of names that are often used as synonyms. In a sense, such a service is a component of a schema matching system whose goal is to help resolve heterogeneity between disparate schemata. A first version of such a service was described in [9].

- Given a name of an attribute, return a set of values for its column. An example of where such a service can be useful is to automatically fill out forms in order to surface deep-web content.

- Given an entity, return a set of possible properties (i.e, attributes and relationships) that may be associated with the entity. Such a service would be useful for information extraction tasks and for query expansion.

- Given a few attributes in a particular domain, return other attributes that database designers use for that domain (akin to a schema auto-complete). A first version of such a service was described in [9]. Such a service would be of general interest for database developers and in addition would help them choose attribute names that are more common and therefore avoid additional heterogeneity issues later.

6. CONCLUSION

We described three systems that perform information extraction in a domain-independent fashion, and therefore can (and have been) applied to the entire Web. The first system, TEXTRUNNER, extracts binary relationships between entities from arbitrary text and therefore obtains a very wide variety of relationships. TEXTRUNNER exploits the power of redundancy on the Web by basing its extractions on *multiple occurrences* of facts on the Web. The WEBTABLES system targets tabular data on the Web and extracts structured data that typically requires multiple attributes to describe and often includes numerical data that would be cumbersome to describe in text. In WEBTABLES, the fact that we have *multiple rows* in a table can provide further clues about the semantics of the data. Our deep-web crawl extracts an additional type of structured data that is currently stored in databases and available behind forms. The data extracted by the deep-web crawl requires additional effort to be fully structured, but the potential arises from the fact that we have *multiple tables* resulting from the same form. In all three cases, a side result of the extraction is a set of entities, relationships and schemata that can be used as building blocks for the Web knowledge base and for additional semantic services.

7. REFERENCES

[1] E. Agichtein, L. Gravano, J. Pavel, V. Sokolova, and A. Voskoboynik. Snowball: A Prototype System for Extracting Relations from Large Text Collections. In *SIGMOD*, 2001.

[2] M. Banko. Personal Communication, 2008.

[3] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web. In *IJCAI*, 2007.

[4] M. Banko and O. Etzioni. The Tradeoffs Between Open and Traditional Relational Extraction. In *ACL*, 2008.

[5] L. Barbosa and J. Freire. Siphoning hidden-web data through keyword-based interfaces. In *SBBD*, 2004.

[6] M. K. Bergman. The Deep Web: Surfacing Hidden Value. *Journal of Electronic Publishing*, 2001.

[7] S. Brin. Extracting Patterns and Relations from the World Wide Web. In *WebDB*, 1998.

[8] M. J. Cafarella, A. Halevy, Y. Zhang, D. Z. Wang, and E. Wu. Uncovering the Relational Web. In *WebDB*, 2008.

[9] M. J. Cafarella, A. Halevy, Y. Zhang, D. Z. Wang, and E. Wu. WebTables: Exploring the Power of Tables on the Web. In *VLDB*, 2008.

[10] J. P. Callan and M. E. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.

[11] Cars.com FAQ. http://siy.cars.com/siy/qsg/faqGeneralInfo.jsp#howmanyads.

[12] Cazoodle Apartment Search. http://apartments.cazoodle.com/.

[13] K. C.-C. Chang, B. He, and Z. Zhang. MetaQuerier over the Deep Web: Shallow Integration across Holistic Sources. In *VLDB-IIWeb*, 2004.

[14] O. Etzioni, M. Cafarella, D. Downey, S. Kwok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in KnowItAll (preliminary results). In *WWW*, 2004.

[15] B. He and K. C.-C. Chang. Statistical Schema Matching across Web Query Interfaces. In *SIGMOD*, 2003.

[16] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang. Accessing the Deep Web: A survey. *Communications of the ACM*, 50(5):95–101, 2007.

[17] M. A. Hearst. Automatic Acquisition of Hyponymns from Large Text Corpora. In *COLING*, 1992.

[18] P. G. Ipeirotis and L. Gravano. Distributed Search over the Hidden Web: Hierarchical Database Sampling and Selection. In *VLDB*, 2002.

[19] J. Madhavan, P. A. Bernstein, A. Doan, and A. Y. Halevy. Corpus-based Schema Matching. In *ICDE*, 2005.

[20] J. Madhavan, S. Jeffery, S. Cohen, X. Dong, D. Ko, C. Yu, and A. Halevy. Web-scale Data Integration: You can only afford to Pay As You Go. In *CIDR*, 2007.

[21] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy. Google's Deep-Web Crawl. In *VLDB*, 2008.

[22] A. Ntoulas, P. Zerfos, and J. Cho. Downloading Textual Hidden Web Content through Keyword Queries. In *JCDL*, 2005.

[23] A. D. Sarma, X. Dong, and A. Halevy. Bootstrapping pay-as-you-go data integration systems. In *SIGMOD*, 2008.

[24] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *WWW*, 2007.

[25] P. P. Talukdar, J. Reisinger, M. Pasca, D. Ravichandran, R. Bhagat, and F. Pereira. Weakly Supervised Acquisition of Labeled Class Instances using Graph Random Walks. In *EMNLP*, 2008.

[26] Trulia. http://www.trulia.com/.

[27] F. Wu and D. S. Weld. Autonomously Semantifying Wikipedia. In *CIKM*, 2007.

[28] W. Wu, C. Yu, A. Doan, and W. Meng. An Interactive Clustering-based Approach to Integrating Source Query Interfaces on the Deep Web. In *SIGMOD*, 2004.

Using Wikipedia to Bootstrap Open Information Extraction

Daniel S. Weld
Computer Science & Engineering
University of Washington
Seattle, WA-98195, USA
weld@cs.washington.edu

Raphael Hoffmann
Computer Science & Engineering
University of Washington
Seattle, WA-98195, USA
raphaelh@cs.washington.edu

Fei Wu
Computer Science & Engineering
University of Washington
Seattle, WA-98195, USA
wufei@cs.washington.edu

1. INTRODUCTION

We often use 'Data Management' to refer to the manipulation of relational or semi-structured information, but much of the world's data is unstructured, for example the vast amount of natural-language text on the Web. The ability to manage the information underlying this unstructured text is therefore increasingly important. While information retrieval techniques, as embodied in today's sophisticated search engines, offer important capabilities, they lack the most important faculties found in relational databases: 1) queries comprising aggregation, sorting and joins, and 2) structured visualization such as faceted browsing [29].

Information extraction (IE), the process of generating structured data from unstructured text, has the potential to convert much of the Web to relational form — enabling these powerful querying and visualization methods. Implemented systems have used manually-generated extractors (e.g., regular expressions) to "screen scrape" for decades, but in recent years machine learning methods have transformed IE, speeding the development of relation-specific extractors and greatly improving precision and recall. While the technology has led to many commercial applications, it requires identifying target relations ahead of time and the laborious construction of a labeled training set. As a result, supervised learning techniques can't scale to the vast number of relations discussed on the Web.

1.1 Open Information Extraction

In order to extract the widely-varying type of information on the Web, attention has recently turned to the broader goal of what Etzioni *et al.* call *open* information extraction [2, 11] — the task of scalably extracting information to fill an unbounded number of relational schemata, whose structure is unknown in advance. Open IE is distinguished from traditional methods on three dimensions [11]:

Input: Traditional, supervised approaches require a set of labeled training data in addition to the corpus for extraction; open IE uses domain-independent methods instead.

Target Schema: In traditional IE, the target relation is specified in advance; open IE automatically discovers the relations of interest.

Computational Complexity: The runtime of traditional methods is $O(D * R)$, where D denotes the number of documents in the corpus and R denotes the number of relations; in contrast, scalability to the Web demands that open IE scale linearly in D.

1.2 The Challenge

IE techniques are typically measured in terms of *precision*, the percentage of extracted items which are correct, and *recall*, the percentage of potentially correct tuples which are actually extracted. In most cases, there is a tradeoff between precision and recall (at 0% recall one has 100% precision), so researchers often look at the recall at some fixed precision (determined by task needs) or the *area under the precision / recall (P/R) curve*.

While several successful systems have tackled open IE [2, 27, 1, 5, 23], demonstrating respectable performance, they face a harder problem than traditional supervised methods. Specifically, the tradeoff between precision and recall is further opposed by the need for generality in the type of text accepted and the range of relations considered. The goal of fully open IE, eschewing manual tuning or the tedious construction of training sets, represents the extreme point on the generality spectrum. The challenge for the future, then, is to improve all three dimensions: precision, recall and generality. How far is it possible to go?

The remainder of this paper presents Kylin as a case study of open IE. We start by describing Kylin's use of Wikipedia to power the self-supervised training of information extractors. Then, in Section 3 we show how Wikipedia training can be seen as a *bootstrapping method* enabling extraction from the wider set of general Web pages. Not even the best machine-learning algorithms have production-level precision; Section 4 discusses how users — engaged in some other primary task — may be unobtrusively enticed to validate extractions, enlarging the training set, and thus improving accuracy over time. Section 5 concludes with observations gleaned from our experience with Kylin. Finally, Section 6 concludes by revisiting the challenges listed above.

2. SELF-SUPERVISED LEARNING

As we have said, information extraction is a process which generates a set of relational tuples by "reading" unstructured text. For example, the simplest case of IE, called named-entity recognition, extracts tuples of a unary relation, *e.g.*, `company(IBM)` from natural-language sentences. Supervised learning of extractors operates in two phases:

Training: Typically a large training corpus is cleaned and

Motto:	*Lux sit* (Latin)
Motto in English:	Let there be light[1]
Established:	1861
Type:	Public flagship Sea grant Space grant
Endowment:	$3.18 billion[2]
President:	Mark Emmert
Provost:	Phyllis Wise
Staff:	3,623

Figure 1: Rendered form of a portion (8 of 18 attributes) of the infobox for the University of Washington.

converted to a sequence of words, $\vec{\mathbf{w}}$. Each word is marked (*e.g.*, by a human) to show which words correspond to companies, thus providing a training set. One basic strategy is then to train a sequential probabilistic model, such as an HMM, deriving the parameters θ which maximize $P[\vec{\mathbf{w}}|\theta]$ by simply counting inter-state transitions and word-emissions per state. It is also possible to make use of unlabeled data by using the greedy *Baum-Welch* algorithm.

Extraction: Given a trained HMM with parameters, θ, and a sequence of test words, *e.g.*, $\vec{\mathbf{w}}$= "Today Amazon released its quarterly earnings," IE uses the dynamic-programming *Viterbi* algorithm to find the most likely state sequence π, i.e. the π that maximizes $P[\vec{\mathbf{w}}, \pi|\theta]$. Any words predicted to have been emitted from the HMM's "company" state are returned as tuples, *e.g.*, `company(Amazon)`.

But how can machine learning be used in the case of *open* IE? One method, incorporated into the TextRunner system [2, 1], learns patterns which indicate *general relationships* and extracts the relation as well as the arguments. We advocate an alternative approach: using Wikipedia to generate *relation-specific* training data for a broad set of thousands of relations.

2.1 Heuristic Generation of Training Data

There are many reasons why Wikipedia is an excellent corpus for training an open IE system. First, it is a comprehensive source of high-quality text about a very wide variety of topics. Secondly, it has a great deal of internal structure which can be exploited for learning [27]. In this paper, we restrict our attention to *infoboxes*, tabular summaries of an article's salient details which are included on a number of Wikipedia pages. For example, Figure 1 shows the infobox from the page on the University of Washington. Since this table is generated via stylesheet from XML source, it is relatively easy to process. When a page has such an infobox one can determine the *class* of the infobox (e.g., the `University` class). By aggregating the source code for a number of instances of an infobox class, one may deduce the most important *attributes* of the class; these correspond to relations whose first argument is an instance of the class. In our example, we have the year `established`, the `provost` and sixteen others.

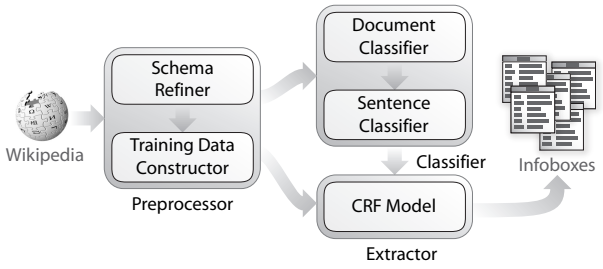


Figure 2: Architecture of Kylin's basic, self-supervised open information extraction system, before shrinkage is applied.

Kylin uses infoboxes to capture the schemata of the most important relations for summarizing the contents of Wikipedia articles. The main problem we address is how to populate those schemata from data extracted from the natural language text of articles *without* infoboxes. Kylin uses existing infoboxes for this second task as well, creating training sets for learning extractors for those relations [27]. The overall architecture of Kylin is shown in Figure 2; we discuss the main components below.

Preprocessor: The preprocessor selects and refines infobox-class schemata, choosing relevant relations; it then generates machine-learning datasets for training sentence classifiers and extractors. Refinement is necessary for several reasons. For example, *schema drift* occurs when authors create an infobox by copying one from a similar article and changing attribute values. If a new attribute is needed, they just make up a name, leading to schema and attribute duplication. Since this is a widespread problem, schema refinement clusters infobox classes which have similar schemata and names with low edit distance. Rare attributes are ignored.

Next, the preprocessor heuristically constructs two types of training datasets — those for sentence classifiers, and for CRF relation extractors. For each article with an infobox mentioning one or more target relations, Kylin tries to find a unique sentence in the article that mentions that attribute's value. The resulting labeled sentences form positive training examples for each relation; other sentences form negative training examples. If the attribute value is mentioned in several sentences, then one is selected heuristically.

Generating Classifiers: Kylin learns two types of classifiers. For each class of article being processed, a heuristic *document classifier* is used to recognize members of the infobox class. For each target relation within a class a *sentence classifier* is trained in order to predict whether a given sentence is likely to contain the attribute's value. For this, Kylin uses a maximum entropy model [17] with bagging. Features include a bag of words, augmented with part of speech tags.

Learning Extractors: Extracting attribute values from a sentence is best viewed as a sequential data-labeling problem. Kylin uses conditional random fields (CRFs) [15] with a wide variety of features (e.g., POS tags, position in the sentence, capitalization, presence of digits or special characters, relation to anchor text, etc.). Instead of training a single

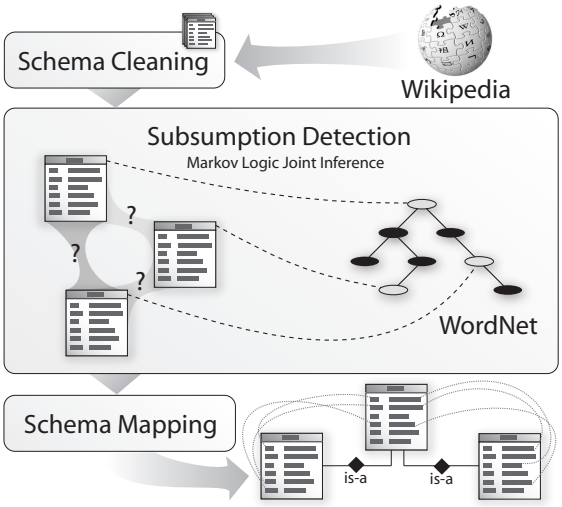


Figure 3: Architecture of Kylin Ontology Generator.

master extractor to clip all relations, Kylin trains a different CRF extractor for each relation, ensuring simplicity and fast retraining. This decomposition also enables easy parallelization for computational scalability.

2.2 Improving Recall with Shrinkage

Although Kylin performs well when it can find enough training data, it flounders on sparsely populated infobox classes — the majority of cases. Fortunately, there is a way to improve Kylin's performance through the use of shrinkage, a general statistical technique for improving estimators in the case of limited training data. McCallum et al. applied this technique for text classification in a hierarchy of classes by smoothing parameter estimates of a data-sparse child with its parent to get more robust estimates [16].

Similarly, we use shrinkage when training an extractor of an instance-sparse infobox class by aggregating data from its parent and child classes. For example, knowing that `Performer IS-A Person`, and `Performer.loc=Person.birth_plc`, we can use values from `Person.birth_plc` to help train an extractor for `Performer.loc`. The trick is finding a good subsumption hierarchy which relates attributes between parent and child classes. Unfortunately, Wikipedia does not contain such a taxonomy[1] Furthermore, previously-created taxonomies for Wikipedia, e.g. [18], don't contain the required relation-relation mappings between parent-child classes. Thus, we were led to devise our own taxonomy, which we did using a novel, autonomous process described below. After explaining the construction of this taxonomy, we describe our approach to shrinkage.

The Kylin Ontology Generator: The Kylin Ontology Generator (KOG) is an autonomous system that builds a rich taxonomy by combining Wikipedia infobox classes with WordNet using statistical-relational machine learning [28]. At the highest level KOG computes six different kinds of features, some metric and some Boolean: *similarity measures*, *edit history patterns*, *class-name string inclusion*, *category tags*, *Hearst patterns*, *search-engine statistics*, and *WordNet* mappings. These features are combined using statistical-relational machine learning, specifically joint inference over Markov logic networks [20], extending [21].

Figure 3 shows the architecture of KOG. First, its *schema cleaner* scans the infobox system to merge duplicate classes and relations, and infers the type signature of each relation. Then, the *subsumption detector* identifies the subsumption relations between infobox classes, and maps the classes to WordNet nodes. Finally, the *schema mapper* builds relation mappings between related classes, especially between parent-child pairs in the subsumption hierarchy [28].

KOG's taxonomy provides an ideal base for the shrinkage technique, as described below.

Shrinkage Using the KOG Ontology: Given a sparse target infobox class C, Kylin's shrinkage module searches upwards and downwards through the KOG taxonomy to aggregate training data from parent and children classes. The overall shrinkage procedure is as follows:

1. Given a class C, query KOG to collect the related class set: $S_C = \{C_i | path(C, C_i) \leq l\}$, where l is the preset threshold for path length. Currently Kylin only searches strict parent/child paths without considering siblings. Take the `Performer` class as an example: its parent `Person` and children `Actor` and `Comedian` could be included in S_C.
2. For each relation $C.r$ (e.g., `Performer.loc`) of C:
   (a) Query KOG for the mapped relation $C_i.r_j$ (e.g., `Person.birth_plc`) for each C_i.
   (b) Assign weight w_{ij} to the training examples from $C_i.r_j$ and add them to the training dataset for $C.r$. Note that w_{ij} may be a function both of the target relation $C.r$, the related class C_i, and C_i's mapped relation $C_i.r_j$.
3. Train the CRF extractors for C on the new training set.

With shrinkage, Kylin learns much better extractors, especially in classes with only a few instances containing infoboxes. For example, the area under the precision and recall curve for the `performer` class (which had 44 instances) improved by 57% after applying shrinkage from `person` (1201 examples), `actor` (8738 examples) and `comedian` (106 examples) [26]. Most of this improvement comes from increased recall, but precision gets a small boost as well.

3. BOOTSTRAPPING TO THE WEB

Even when Kylin does learn an effective extractor there are numerous cases where Wikipedia has an article on a topic, but the article simply doesn't have much information to be extracted. Indeed, a long-tailed distribution governs the length of articles in Wikipedia — around 44% of articles are marked as stub pages — indicating that much-needed

[1]Wikipedia's category system is not directly useful in this regard, for several reasons. First, the category system is too flat. Second, there are many administrative tags mixed into the categories. Third, the "content-bearing" tags are typically conjunctive, e.g. `Jewish physicist`, which require significant processing to decompose into an orthogonal taxonomy.

information is missing. Additionally, facts that are stated using uncommon or ambiguous sentence structures also hide from the extractors. This section shows how to extend the previously-described methods to support extraction from the broader Web.

The challenge for this approach — as one might expect — is maintaining high precision and recall. Since Kylin's extractors have been trained on the somewhat idiosyncratic Wikipedia corpus, they may not extend smoothly to the Web. After considering the roots of the problem, we discuss the extensions necessary to successfully bootstrap.

- **Reduced Recall When Extracting from the Web:** In many cases, the language used on Wikipedia is stylized and looks very different from pages on the general Web. For example, on Wikipedia a person's date of birth and his professions are often expressed in the first sentence, which begins with the person's name, then contains the birth date in parentheses, then the verb *is*, and finally a list of professions. When trained on such data, an extractor fails to extract from sentences of different style. For example, the extractor might not be able to extract a person's birthdate from the sentence "Obama was born on August 4, 1961."
- **Reduced Precision When Extracting from the Web:** Extractors trained on Wikipedia pages are often unable to discriminate irrelevant information. For example, a Kylin extractor for a person's birthdate is trained on a set of pages which all have that person as their primary subject. Such extractors become inaccurate when applied to a page which compares the lives of *several* people — even if the person in question is one of those mentioned.

While Kylin's self-supervised approach allows it to learn extractors for a comprehensive range of relations, using only Wikipedia as a training source also limits its ability to extract from the broader Web. In response, we developed two techniques which dramatically improved its precision and recall when extracting from the general Web.

3.1 Generalizing Extractors with Retraining

Our first idea is to harvest additional training data from the *outside* Web for improved training, which we call *retraining*. The challenge is automatically identifying relevant sentences given the sea of Web data. For this purpose, Kylin utilizes TextRunner, an open IE system [1], which extracts semantic relations $\{r|r = \langle obj_1, predicate, obj_2 \rangle\}$ from a crawl of about 100 million Web pages. Importantly for our purposes, TextRunner's crawl includes the top ten pages returned by Google when queried on the title of every Wikipedia article.

For each target relation within an infobox, Kylin queries to identify and verify relevant sentences that mention the attribute's value. Sentences passing the verification are labeled as extra positive training examples. Kylin also identifies the phrases (predicates) which are harbingers of each target relation. For example, "was married to" and "married" are identified for the "person.spouse" relation. These harbingers are used to eliminate potential false negative training examples generated from the Wikipedia data.

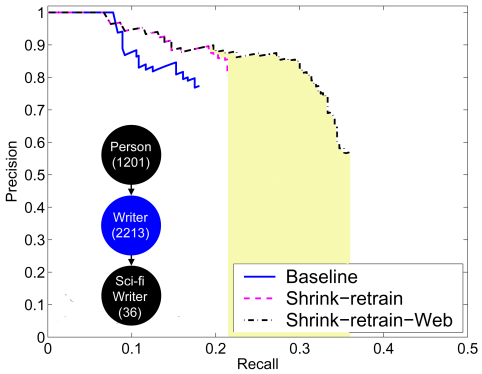


Figure 4: Using Kylin's retrained extractors to extract from the Web results in a substantial improvement to the area under the P/R curve — even in infobox classes, like `writer`, which have thousands of instances.

By adding new positive examples and excluding potential false negative sentences, retraining generates a cleaned and augmented training dataset which improves Kylin's performance. When used together with shrinkage, the improvement in recall is enormous. For example, on the `performer` class, recall improved by 73% on Wikipedia data, and it doubled on Web data. In other, more instance-sparse classes, the improvement in recall was even higher, as much as 1755% [26].

3.2 Selecting Quality Sources

Our second method, targeted towards increasing precision, is about deciding if an extraction from a Web page is indeed relevant to the subject of a Wikipedia article. We view this as an information-retrieval problem which we solve by carefully selecting and weighting extractions from Web pages. This requires a number of steps: First, Kylin generates a set of relevant queries and utilizes a general Web search engine, namely Google, to identify a set of pages which are likely to contain the desired information. These pages are fetched, and Kylin's extractors are applied to all content. Each extraction is then weighted using a combination of factors, including the rank of the page, the extraction confidence (score computed by the CRF), and the distance between the current sentence and the closest sentence containing the name of the Wikipedia article.

Our experiments revealed that the CRF's confidence is a poor choice for scoring different extractions of the same relations, but that the rank of the source page in response to our queries, and especially the distance between the extracted text and the closest sentence containing the name of the article are extremely useful. A weighted combination performed best, and roughly doubled precision of Web extractions for a variety of classes [26].

Extraction results from the Web are later combined with extraction results from Wikipedia. Higher weight is given to extractions from Wikipedia, because it is still likely that extractions from Wikipedia will be more precise. That is, in Wikipedia we can be more certain that a given page is highly relevant, is of higher quality, has a more consistent structure, for which Kylin's extractors have been particularly trained.

Integrating evidence from both Wikipedia and the greater Web further helps Kylin's performance. For example, on the

performer class, the area under the P/R curve improved 102%; for classes which are even more sparse, the improvement can be ten times as high [26]. Recall can even be improved on classes with many training instances; for example, Figure 4 shows the improvement on the writer class which has 2213 instances with infoboxes.

4. COMMUNAL CORRECTION

While some of the CRF extractors learned by Kylin have extremely high precision, in most cases precision is well below that of humans. Thus, a *fully* automated process for producing high-reliability, structured data (*e.g.*, as would be required in order to to add extracted data back into Wikipedia infoboxes) may be untenable. Instead, Kylin aims to amplify *human* effort towards this task. Figure 5 shows our first mockup design of this interface. In a series of interviews with members of the Wikipedia community, informal design reviews, and a final online user study we refined, explored and evaluated the space of interfaces, focusing on the design dimensions listed below.

4.1 Key Issues for Correction Interfaces

Contribution as a Non-Primary Task: Although tools already exist to help expert Wikipedia editors quickly make large numbers of edits [7], we instead want to enable contributions by the long tail of users *not yet contributing* [24]. In other words, we believe that pairing IE with *communal content creation* will be most effective if it encourages contributions by people who had not otherwise planned to contribute. This means that we must treat contributing as a *non-primary task* — encouraging contributions from people engaged in some other primary activity.

Inviting Contributions: Any system based in community content creation must provide an incentive for people to contribute. Bryant et al. report that newcomers become members of the Wikipedia community by participating in peripheral, yet productive, tasks that contribute to the overall goal of the community [4]. Given this behavior, our goal is to make the opportunity to contribute sufficiently salient that people will try it, but not so visible as to make the interface obtrusive or coercive. We designed three new interfaces to explore this tradeoff.

Presenting Ambiguity Resolution in Context: As shown in Figure 5 there are two plausible locations in an article for presenting each potential extraction: near the article text from which the value was extracted or proximal to the infobox where the data is needed. Presenting information at the latter location can be tricky because the contributor cannot verify information without knowing the context. Furthermore, varying the way that context is presented can dramatically affect user participation.

4.2 Preliminary User Study

We evaluated the effectiveness of our interfaces in stimulating edits as well as users' perception of interface intrusiveness in a novel study, deployed via Google Adwords and Yahoo Keyword Advertising [14]. While we developed

Figure 5: User interface mockup. Casual users are presented with a standard Wikipedia page highlighting a single attribute value; an ignorable popup window allows the user to verify the extraction if she wishes.

some interface designs which yielded even higher participation rates, the "winner" of our study was an icon design that was deemed relatively unobtrusive and yet which led people to voluntarily contribute an average of one fact validation for every 14 page visits. Validations had a precision of 90%. By validating facts multiple times from different visitors, we believe we can achieve very high precision on extracted tuples. Preliminary experiments also show that by adding these new tuples as additional training examples, Kylin will keep increasing extractor performance.

5. LESSONS LEARNED

Our experience in building the Kylin system and running it over the constantly-changing Wikipedia yields several lessons.

5.1 Approaches to Open IE

In traditionally, relational databases, rigid schemata are defined before any data is added; indeed, a well-defined schema facilitates the expression of queries. But if one wishes to extract a wide variety of information from the Web, for example a large fraction of data from Wikipedia, then human preengineering of such schemata (and associated construction of labeled training data for each relation) is impossible.

In response, we focus on what Etzioni *et al.* term *open* information extraction — algorithms which can handle an unbounded number of relations, eschew domain-specific training data, and scale linearly to handle Web-scale corpora. Besides Kylin, only a few open IE systems have yet been built, with the first being TextRunner [2, 1]. We believe that all open IE systems may be placed into two groups, which we term *relational-targeting* and *structural-targeting* methods. Kylin uses a relational approach, but to our knowledge all other open IE systems use structural targeting.

Relational Targeting: Learning a relation-specific extractor is the most common technique in traditional IE; indeed, the example in the beginning of Section 2 trained an HMM where one state corresponded to words naming an instance of the company relation. The first challenge for this method is acquiring enough training data for a comprehensive set of relations. This paper has shown how Kylin's self-supervised

approach uses Wikipedia infoboxes to heuristically assemble training sets. We estimate that Wikipedia contains 5000 infobox classes, each of which has approximately 10 attributes. Thus, while Kylin can't truly handle an *unbounded* number of relations, it seems capable of learning 50,000, which may be sufficient.

We contrast our approach with systems such as Yago [22], which also use Wikipedia as a corpus, but learn a fixed set of a dozen relations using substantial manual effort. While systems such as DBLIFE [8] scale relatively well, they aren't "open" either, requiring manually-crafted rules for extraction.

Structural Targeting: An alternative approach is to build a *general* extraction-engine which looks for some form of relation-independent structure on Web pages and uses this to extract tuples. A postprocessing step is often used to normalize the extractions, determining the precise relation and entities which have been extracted. There are many different forms of structure which may be used in this process. For example, TextRunner [2, 1] and Knext [23] exploit grammatical structure over sentences. Hearst patterns operate on phrases within sentences [13, 10]. Other work uses HTML structure, such as the DOM tree, to extract lists and tables [9, 10, 12, 5]

While open IE systems using structural targeting can truly handle an unbounded number of relations, such generality often comes at the cost of lower precision when compared to relationally-targeted extractors. This brings us back to the challenge described in Section 1.2: can open IE methods really provide generality along with high precision and recall. We think the answer is "yes." Both structural and self-supervised relational approaches have made impressive progress. And importantly, the different approaches are complementary, as we demonstrate with Kylin's retraining methodology. Future work may focus on other ways to combine these approaches.

5.2 Scalability of Relation-Specific Open IE

To scale extraction to a large number of relations from a large number of pages, parallelization is important and already we see information extraction algorithms leveraging map-reduce-style data processing frameworks [6]. Yet, parallelization alone is not sufficient. For example, Kylin learns a relation-specific extractor for each attribute of each infobox class in Wikipedia — potentially more than 50,000 extractors in total. It would be impractical to run every extractor on each Wikipedia page, let alone each page on the Web! Fortunately, Kylin's hierarchical approach for classifying Wikipedia pages and sentences, alleviates the problem when the extraction corpus is Wikipedia itself. However, it's unclear that this approach can be extended to the general Web.

5.3 Integrating Evidence

High precision and recall requires collecting information from many sources. In fact, leveraging the redundancy of content on the Web, utilizing structural properties of Web sites or their contents, and leveraging relevant databases [3] is often crucial for resolving extraction ambiguities. One method accomplishing this integration is joint inference [19, 25]. Additionally, the need to decompose extractors for reasons of efficiency further increases the amount of evidence being collected. The biggest challenge for this task is likely managing the tradeoff between computational blowup (if exact approaches are attempted) and greatly reduced accuracy (if approximations such as probabilistic independence are made).

Kylin learns a myriad of extractors independently, but what we consider as one of its main contributions, is the realization that a careful integration of extractors exploiting various structural properties of content and sites, can lead to dramatic improvements in precision and recall. Kylin automatically infers a taxonomy underlying Wikipedia data and utilizes this taxonomy to help extraction, even though the taxonomy itself is not its goal. When combined, techniques such as shrinkage, retraining, and Web-source selection enable Kylin to perform well, even when targeting the long tail of rarely occurring relations. We are also looking to improve Kylin's integration abilities through the use of probabilistic CFGs. This may provide a cleaner way to integrate page and sentence classification with extraction; this method also enables joint inference. As always, scalability is likely to be the defining challenge.

5.4 The Role of Humans

In the near future, we are unlikely to achieve extremely high precision extraction without human involvement. Researchers have explored four main ways for involving humans in the process: 1) humans write rule-based extraction procedures, 2) humans label training data for supervised learning of extractors, 3) humans validate candiate extractions, regardless of how they are produced, and 4) humans manually aggregate structured information (*e.g.*, as in Freebase). We believe, that in many cases, one can best utilize the rare human resource by combining self-supervised learning with crowd-sourced validation. The study described in Section 4.2 shows that high participation rates are possible.

6. CONCLUSION

By converting unstructured, natural-language text to relational form, information extraction enables many powerful Data Management techniques. However, in order to scale IE to the Web, we must focus on *open* IE — a paradigm that tackles an unbounded number of relations, eschews domain-specific training data, and scales computationally [2, 11]. This paper describes Kylin, which uses self-supervised learning to train relationally-targeted extractors from Wikipedia infoboxes. We explained how shrinkage and retraining allow Kylin to improve extractor robustness, and we demonstrate that these extractors can successfully mine tuples from a broader set of Web pages. Finally, we argued that the best way to utilize human efforts is by inviting humans to quickly validate the correctness of machine-generated extractions.

We distill several lessons from our experience. Perhaps our most important observation contrasts two approaches to open IE: relational *vs.* structural targeting. While Kylin primarily uses the relational approach, we argue that the best chance for jointly optimizing extraction generality, precision, and recall will be to further combine relational and structural approaches.

Acknowledgments: We thank Eytan Adar, Saleema Amershi, Mike Cafarella, AnHai Doan, Oren Etzioni, Krzysztof Gajos, James Fogarty, Chloé Kiddon, Shawn Ling, Kayur Patel, and Stefan Schoenmackers for valuable discussions. Evgeniy Gabrilovitch and Ken Schmidt greatly facilitated our user study. This work was supported by NSF grant IIS-0307906, ONR grant N00014-06-1-0147, SRI CALO grant 03-000225, the WRF / TJ Cable Professorship and a grant from Yahoo! Inc.

7. REFERENCES

[1] M. Banko and O. Etzioni. The tradeoffs between traditional and open relation extraction. *Proceedings of ACL08*, 2008.

[2] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the Web. *Proceedings of IJCAI07*, 2007.

[3] Kedar Bellare and Andrew McCallum. Learning extractors from unlabeled text using relevant databases. *Proceedings of IIWeb08 Workshop*, 2007.

[4] S. Bryant, A. Forte, and A. Bruckman. Becoming Wikipedian: transformation of participation in a collaborative online encyclopedia. *Proceedings of GROUP05*, 2005.

[5] Michael J. Cafarella, Alon Halevy, Yang Zhang, Daisy Zhe Wang, and Eugene Wu. Webtables: Exploring the power of tables on the web. *Proceedings of VLDB08*, 2008.

[6] C. Chu, S. Kim, Y. Lin, Y. Yu, G. Bradski, A. Ng, and K Olukotun. Map-reduce for machine learning on multicore. *Proceedings of NIPS06*, Vancouver, Canada, 2006.

[7] D. Cosley, D. Frankowski, L. Terveen, and J. Riedl. Suggestbot: Using intelligent task routing to help people find work in wikipedia. *Proceedings of IUI07*, January 2007.

[8] P. DeRose, X. Chai, B. Gao, W. Shen, A. Doan, P. Bohannon, and J. Zhu. Building community wikipedias: A human-machine approach. *Proceedings of ICDE08*, 2008.

[9] R. Doorenbos, O. Etzioni, and D. Weld. A scalable comparison-shopping agent for the World-Wide Web. *Proceedings of AGENTS97*, pages 39–48, Marina del Rey, California, 1997.

[10] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, 165(1):91–134, 2005.

[11] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the Web. *Communications of the ACM*, **51**(12) 2008.

[12] Wolfgang Gatterbauer, Paul Bohunsky, Marcus Herzog, Bernhard Krüpl, and Bernhard Pollak. Towards domain-independent information extraction from web tables. *Proceedings of WWW07*, 2007.

[13] M. Hearst. Automatic acquisition of hyponyms from large text corpora. *Proceedings of COLING92*, 1992.

[14] R. Hoffmann, S. Amershi, K. Patel, F. Wu, J. Fogarty, and D. S. Weld. Amplifying community content creation with mixed-initiative information extraction. *Proceedings of CHI09*, 2009.

[15] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of ICML01*, Edinburgh, Scotland, May 2001.

[16] Andrew K. McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. Jude W. Shavlik, editor, *Proceedings of ICML98*, pages 359–367, Madison, US, 1998. Morgan Kaufmann Publishers, San Francisco, US.

[17] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. *Proceedings of the IJCAI99 Workshop on Machine Learning for Information Filtering*, 1999.

[18] S. Ponzetto and M. Strube. Deriving a large scale taxonomy from Wikipedia. *Proceedings of AAAI07*, pages 1440–1445, 2007.

[19] Hoifung Poon and Pedro Domingos. Joint inference in information extraction. *Proceedings of AAAI08*, pages 913–918, 2007.

[20] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 2006.

[21] Rion Snow, Daniel Jurafsky, and A. Ng. Semantic taxonomy induction from heterogenous evidence. *Proceedings of ACL06*, 2006.

[22] F. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge - unifying WordNet and Wikipedia. *Proceedings of WWW07*, 2007.

[23] B. Van Durme and L.K. Schubert. Open knowledge extraction through compositional language processing. *Symposium on Semantics in Systems for Text Processing*, 2008.

[24] J. Voss. Measuring wikipedia. *International Conference of the International Society for Scientometrics and Informetrics*, 2005.

[25] Michael Wick, Khashayar Rohanimanesh, Karl Schultz, and Andrew McCallum. A unified approach for schema matching, coreference and canonicalization. *Proceedings of KDD08*, 2008.

[26] Fei Wu, Raphael Hoffmann, and Danel S. Weld. Information extraction from Wikipedia: Moving down the long tail. *Proceedings of KDD08*, 2008.

[27] Fei Wu and Daniel Weld. Autonomouslly semantifying Wikipedia. *Proceedings of CIKM07*, Lisbon, Porgugal, 2007.

[28] Fei Wu and Daniel Weld. Automatically refining the Wikipedia infobox ontology. *Proceedings of WWW08*, 2008.

[29] K. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. *Proceedings of SIGCHI03*, 2003.

Modeling and Querying Probabilistic XML Data*

Benny Kimelfeld
IBM Almaden Research Center
kimelfeld@us.ibm.com

Yehoshua Sagiv†
The Hebrew University of Jerusalem
sagiv@cs.huji.ac.il

1 Introduction

We survey recent results on modeling and querying probabilistic XML data. The literature contains a plethora of probabilistic XML models [2, 13, 14, 18, 21, 24, 27], and most of them can be represented by means of *p-documents* [18] that have, in addition to ordinary nodes, *distributional* nodes that specify the probabilistic process of generating a random document. The above models are families of p-documents that differ in the *types* of distributional nodes in use.

The focus of this survey is on the tradeoff between the ability to express real-world probabilistic data (in particular, by taking correlations between atomic events into account) and the efficiency of query evaluation. We concentrate on two important issues. The first is the ability to efficiently translate a p-document of one family into that of another. The second is the complexity of query evaluation over p-documents (under the usual semantics of querying probabilistic data, e.g., [4, 9, 10]). It turns out that efficient evaluation of a large class of queries (i.e., twig patterns with projection and aggregate functions) is realizable in models where distributional nodes are probabilistically independent. In other models, the evaluation of a query with projection is very often intractable. In comparison, very simple conjunctive queries are intractable over probabilistic models of relational databases, even when the tuples are probabilistically independent [9, 10].

To handle the limitation exhibited by the above tradeoff, various approaches have been proposed. The first is to allow query answers to be *approximate* [18], which makes the evaluation of twig patterns with projection tractable in the most expressive family of p-documents, among those considered. This tractability, however, does not carry over to non-monotonic queries, such as twig patterns with negation or aggregation. The approach presented in [7] combines the assumption about the independence of distributional nodes with the assertion of (a fixed set of) *constraints*, thereby deriving a model capable of representing complex correlations between atomic events, but not at the expense of efficiency.

2 Probabilistic XML

We model XML documents as unranked and unordered trees. Each node v has a *label* and a unique *identifier* that are denoted by $\lambda(v)$ and $id(v)$, respectively. When representing an XML document as a tree, a node corresponds to either an *element*, an *attribute* or a *value* (i.e., PCDATA or the value of an attribute). Accordingly, the label of a node corresponds to either an element name (i.e., tag), an attribute name or a value. Trees of the above form are called *documents*.

A *probabilistic XML space* (abbr. *px-space*) $\tilde{\mathscr{D}}$ is a pair (Ω, p), where Ω is a nonempty and finite set of documents, and $p : \Omega \to \mathbb{Q}^+$ maps every document $d \in \Omega$ to a positive rational number $p(d)$, such that $\sum_{d\in\Omega} p(d) = 1$. The set Ω is the *sample space* of $\tilde{\mathscr{D}}$, and p is the *probability distribution*. The documents of Ω are also called *possible worlds* (or *samples*). We identify $\tilde{\mathscr{D}}$ with its sample space Ω; for example, we write $d \in \tilde{\mathscr{D}}$ instead of $d \in \Omega$.

Typically, a px-space describes uncertainty in many parts of the data. Hence, its sample space may be too large for allowing an explicit representation. Next, we describe a compact representation of a px-space $\tilde{\mathscr{D}}$ by means of a p-document, which is (a description of) a probabilistic process that generates a random document $d \in \tilde{\mathscr{D}}$ with probability $p(d)$.

2.1 The P-Document Model

A *p-document* is a tree $\tilde{\mathscr{P}}$ that consists of two types of nodes. *Ordinary* nodes are those described at the beginning of Section 2, namely, each one has a label and a unique identifier. Ordinary nodes may appear in the documents of the sample space. *Distributional* nodes, on the other hand, are only used for defining the probabilistic process that generates random documents (but they do not actually occur in those doc-

*Database Principles Column. Column editor: Leonid Libkin, School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, UK. E-mail: libkin@inf.ed.ac.uk.

†The work of this author was supported by The Israel Science Foundation (Grant 893/05).

uments). In Section 3, several types of distributional nodes are defined. For now, it is sufficient to realize that each distributional node v has a probability distribution over subsets of its children. In the probabilistic process that generates a random document, v randomly chooses a subset of its children according to the distribution specified for v. In a p-document, the root and leaves are ordinary nodes.

As an example, Figure 1 shows a p-document $\tilde{\mathscr{P}}$. Each ordinary node v is represented by the string $id(x).\lambda(v)$ (e.g., "3.member"). Distributional nodes are depicted as rounded-corner rectangles. The type of a distributional node is indicated by the string (e.g., ind or mux) appearing inside the rectangle. These types are discussed in Section 3.

Given a p-document $\tilde{\mathscr{P}}$, a random document is generated in two steps. First, each distributional node randomly chooses a subset of its children. Note that the choices of different nodes are not necessarily probabilistically independent. All the unchosen children and their descendants (even descendants that have been chosen by their own parents) are deleted. The second step removes all the distributional nodes. If an ordinary node u remains, but its parent is removed, then the new parent of u is the lowest ordinary node v of $\tilde{\mathscr{P}}$, such that v is a proper ancestor of u. Note that when distributional nodes have distributional children, it may happen that the same document is obtained from two different applications of the first step. For additional details, see [18, 19].

3 Concrete Models

3.1 Types of Distributional Nodes

To obtain a concrete p-document, every distributional node should have a specific probability distribution of choosing a subset of its children. We define five types of distributional nodes, each one is characterized by a different way of describing that probability distribution. A node v of type ind specifies for each child w, the probability of choosing w. This probability is independent of the other choices of children. As a special case, a node v of type det always (deterministically) chooses all of its children. If the type of v is mux, then choices of different children are mutually exclusive. That is, v chooses at most one of its children, and it specifies the probability of choosing each child (so the sum of these probabilities is at most 1). A node v of type exp specifies the probability distribution explicitly. That is, v lists subsets of children and their probabilities of being chosen. We assume that the probabilities specified in a p-document are all non-zero, because it is useless to consider choices of children that have a zero probability of occurring.

In the above four types, the underlying assumption is that the choices of different distributional nodes are probabilistically independent. The next type makes it possible to introduce correlations between choices of nodes. When distributional nodes of type cie appear in a p-document $\tilde{\mathscr{P}}$, it means that $\tilde{\mathscr{P}}$ has independent random Boolean variables $e_1, \ldots, e_m$, called *event variables*. For each variable e_i, the p-document $\tilde{\mathscr{P}}$ specifies the probability $p(e_i)$ that e_i is **true**. Each node v of type cie specifies for every child w, a conjunction[1] $\alpha^v(w) = a_1 \wedge \cdots \wedge a_{l_w}$, where each a_j is either e_i or $\neg e_i$ for some $1 \leq i \leq m$ (note that different cie nodes can share common event variables). When generating a random document, values for $e_1, \ldots, e_m$ are randomly picked out, and a child is chosen if its corresponding conjunction is satisfied.

Next, we discuss how to compute the probability of a possible world d that belongs to the px-space defined by a p-document $\tilde{\mathscr{P}}$. Consider an execution of the probabilistic process that generates the subtree s of $\tilde{\mathscr{P}}$ in the first step and then produces d in the second step. For each distributional node v of s, such that the type of v is not cie, let p_v be the probability that v chooses exactly the children that it has in s. It is easy to compute p_v from the probability distribution specified for v. For the cie nodes, we have to compute the probability p_e of all the truth assignments τ to the event variables, such that for every cie node v of s the following holds. For all children w of v, if w appears in s, then τ satisfies $\alpha^v(w)$; otherwise, it does not. Let $p(s)$ be the product of p_e and all the p_v. Note that $p(s)$ is the probability that each distributional node of s chooses *exactly* the children that it has in s. Equivalently, it is the probability of getting s at the end of the first step. The probability of the possible world d is the sum of the probabilities $p(s)$ over all the subtrees s that yield d at the end of the second step. Note that computing each of p_e, $p(s)$ and the probability of d is generally intractable. But if there are no cie nodes in $\tilde{\mathscr{P}}$, then these three probabilities can be computed efficiently.

3.2 Families of P-Documents

We denote by $\mathsf{PrXML}^{\{\text{type}_1,\text{type}_2,\ldots\}}$ the family of all the p-documents, such that the types of their distributional nodes are among those listed in the superscript. For example, the p-documents of $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ use only ind and mux nodes.

The simple case of using a distributional node is when both its parent and children are ordinary. Then, the role of the distributional node is to choose ordinary children for its ordinary parent. Sometimes, however, we can obtain more complex probability

[1] We assume that the conjunction is satisfiable, that is, it does not include an event variable and its negation.

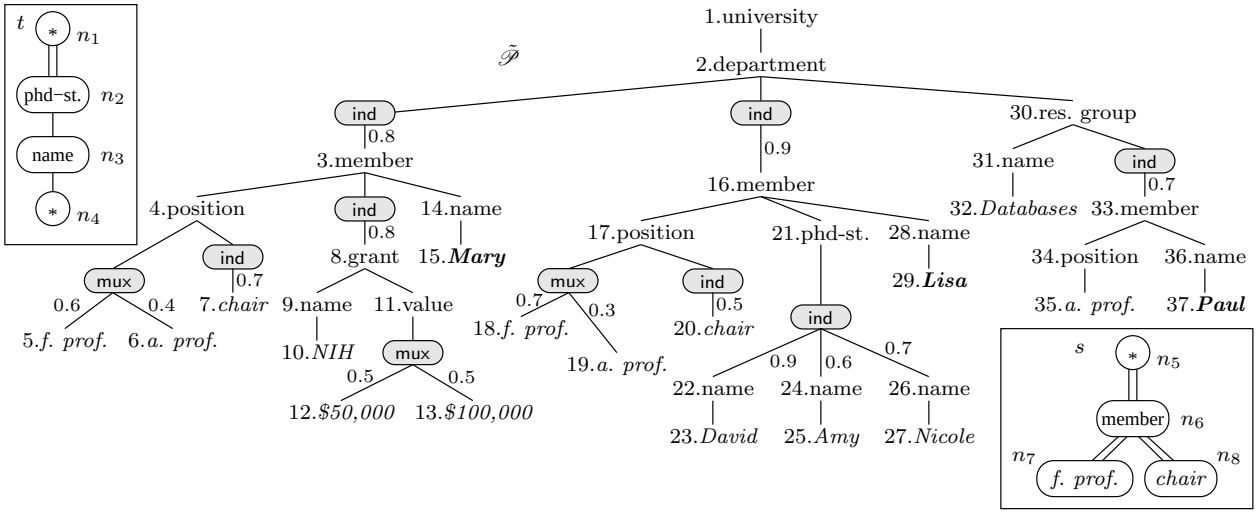


Figure 1: A p-document $\tilde{\mathscr{P}}$ and twigs t and s

distributions (over the space of documents) by constructing hierarchies of distributional nodes.

Formally, a p-document $\tilde{\mathscr{P}}$ is *distributional-hierarchy free* (abbr. DHF) if every distributional node of $\tilde{\mathscr{P}}$ has only ordinary children. As an example, the p-document $\tilde{\mathscr{P}}$ of Figure 1 is DHF. If $\mathscr{F}$ is a set of p-documents, then $\mathscr{F}_{|\not\perp}$ denotes the restriction of $\mathscr{F}$ to its DHF p-documents.

3.3 Expressiveness of Models

We have five types of distributional nodes and the option of either allowing or forbidding hierarchies. It gives more than fifty combinations—do all of them create families of p-documents that are inherently different from one another? In this section, we compare different families in terms of their expressiveness, using the notion of translations. A family $\mathscr{F}_1$ can be *translated* into a family $\mathscr{F}_2$ if there is an algorithm that accepts as input a p-document $\tilde{\mathscr{P}}_1$ of $\mathscr{F}_1$ and constructs as output a $\tilde{\mathscr{P}}_2$ of $\mathscr{F}_2$, such that $\tilde{\mathscr{P}}_1$ and $\tilde{\mathscr{P}}_2$ describe the same px-space. The translation is *efficient* if the algorithm runs in polynomial time.

As a simple example, we compare the families $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ and $\mathsf{PrXML}^{\{\mathsf{mux}\}}$. There is no translation of $\mathsf{PrXML}^{\{\mathsf{mux}\}}$ into $\mathsf{PrXML}^{\{\mathsf{ind}\}}$, because ind nodes cannot express mutually exclusive choices. In particular, consider the p-document $\tilde{\mathscr{P}}_1$ of Figure 2(a). Note that rectangles and circles are distributional and ordinary nodes, respectively; furthermore, for each child of a distributional node, the probability of choosing it is written next to its incoming edge. $\tilde{\mathscr{P}}_1$ is in $\mathsf{PrXML}^{\{\mathsf{mux}\}}$ and it creates exactly two possible worlds that have the sets of nodes $\{v, u_1\}$ and $\{v, u_2\}$. No p-document of $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ can yield these two possible worlds without also generating a third one that has the set of nodes $\{v, u_1, u_2\}$.

Figure 4 shows how to translate each ind node in a p-document of $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\perp}$ to several mux nodes (note that the probability of each w_i remains the same). Hence, $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\perp}$ is efficiently translatable into $\mathsf{PrXML}^{\{\mathsf{mux}\}}$.

However, $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ cannot be translated into $\mathsf{PrXML}^{\{\mathsf{mux}\}}$. To see why, consider the p-document

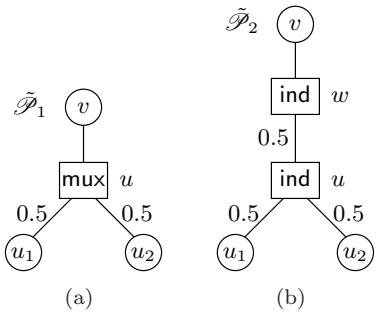


Figure 2: P-documents: (a) $\tilde{\mathscr{P}}_1$ and (b) $\tilde{\mathscr{P}}_2$

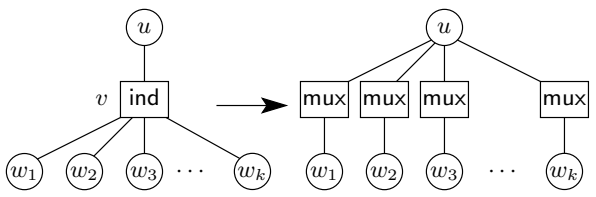


Figure 4: Translating $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\not\perp}$ to $\mathsf{PrXML}^{\{\mathsf{mux}\}}$

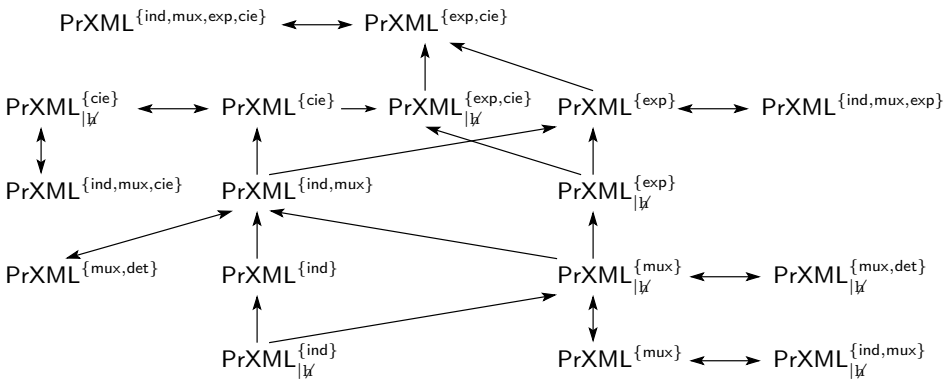


Figure 3: Efficient translations between families of p-documents

$\tilde{\mathscr{P}}_2 \in \mathsf{PrXML}^{\{\mathsf{ind}\}}$ of Figure 2(b). We define $E(x)$ as the event "node x appears in some possible world of $\tilde{\mathscr{P}}_2$." The events $E(u_1)$ and $E(u_2)$ have the same prior probability, namely, 0.25. But the conditional probability of $E(u_1)$, given the occurrence of $E(u_2)$, is 0.5 (because the appearance of u_2 in a possible world implies that node u has been chosen by its parent w). Suppose that some p-document $\tilde{\mathscr{P}}' \in \mathsf{PrXML}^{\{\mathsf{mux}\}}$ generates the same px-space as $\tilde{\mathscr{P}}_2$. A simple case analysis shows that in the px-space defined by $\tilde{\mathscr{P}}'$, the events $E(u_1)$ and $E(u_2)$ are either mutually exclusive or independent. Hence, no p-document of $\mathsf{PrXML}^{\{\mathsf{mux}\}}$ can generate the same px-space as $\tilde{\mathscr{P}}_2$.

Thus, the families $\mathsf{PrXML}^{\{\mathsf{ind}\}}$ and $\mathsf{PrXML}^{\{\mathsf{mux}\}}$ are incomparable in terms of expressive power. Note that both families are subsets of $\mathsf{PrXML}^{\{\mathsf{ind,mux}\}}$. Interestingly, the family $\mathsf{PrXML}^{\{\mathsf{ind,mux}\}}$ is efficiently translatable into $\mathsf{PrXML}^{\{\mathsf{mux,det}\}}$, as illustrated in Figure 5. The converse is trivial, because a det node is a special case of an ind node.

A thorough study of translations between families of p-documents is done in [1]. The results are summarized in Figure 3. An arrow means that there is an efficient translation in the specified direction. The figure is complete in the sense that if there is no directed path from a family $\mathscr{F}_1$ to another family $\mathscr{F}_2$, then an efficient translation does not exist.

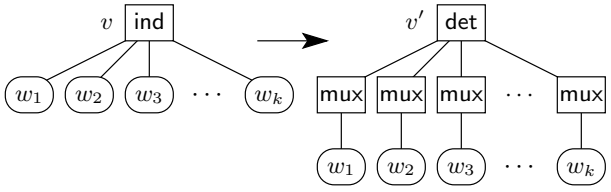


Figure 5: Translating $\mathsf{PrXML}^{\{\mathsf{ind,mux}\}}$ into $\mathsf{PrXML}^{\{\mathsf{mux,det}\}}$

3.4 Object and Value Semantics

Thus far, we have used the *object-based semantics*, namely, two documents cannot be the same if one has a node id that does not appear in the other. Sometimes, node ids are not important per se and, hence, the *value-based semantics* might be more suitable. Under this semantics, two documents are deemed the same if they are isomorphic. Formally, d_1 and d_2 are *isomorphic* if there is a one-to-one correspondence h between the nodes of d_1 and those of d_2, such that h preserves the tree structure and the labels, but not necessarily the ids.

When working with the value-based semantics, the intrinsic measure of uncertainty associated with a document d is the probability that a random possible world is isomorphic to d. In comparison, under the object-based semantics, we are interested in the probability that a random possible world is d itself. This distinction gives rise to the notion of a *v-translation* that transforms a p-document $\tilde{\mathscr{P}}_1$ to $\tilde{\mathscr{P}}_2$, such that the two generate isomorphic px-spaces. In the previous section, we actually discussed *o-translations* that are founded on the object-based semantics. Note that an o-translation is also a v-translation, but the converse is not necessarily true. Therefore, the existence of a directed path in Figure 3 means that there is an efficient v-translation. It is not known whether Figure 3 is complete for efficient v-translations. Notwithstanding, [1] shows that in many cases, the lack of a directed path indicates that there is no efficient v-translation. The main open problem is the following: Is $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ efficiently v-translatable into $\mathsf{PrXML}^{\{\mathsf{mux,det}\}}$, or at least into $\mathsf{PrXML}^{\{\mathsf{cie}\}}$?

3.5 Previously Studied Models

The family $\mathsf{PrXML}^{\{\mathsf{ind,mux}\}}$ is the ProTDB model of [21]. This model has also been studied in [7, 19].

The probabilistic XML model[2] of [27] is a subset of $\mathsf{PrXML}^{\{\mathsf{mux},\mathsf{det}\}}$, where mux nodes (called "probability nodes") have only det nodes as children (called "possibility nodes") and det nodes have only ordinary children (called "XML nodes").

The model of probabilistic XML that was investigated in [2, 24] is $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. The "simple probabilistic trees" of [2] are actually the family $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{\not\Downarrow}$.

The work of [14] introduced a model of probabilistic XML graphs, where each node explicitly specifies the probability distribution over its possible sets of children. Restricting their XML graphs to trees yields a sub-family of $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{\not\Downarrow}$. The same is true for [13] if we restrict their intervals to points.

4 Querying Probabilistic XML

In this section, we survey results on query evaluation over probabilistic XML. The focus is on queries that are based on twig patterns [3, 5]. Formally, a *twig* is a tree t with *child* and *descendant* edges, which are depicted by single and double lines, respectively, in the rectangular boxes of Figure 1. A *match* of a twig t in a document d is a mapping μ from the nodes of t to those of d, such that μ maps root to root, nodes to nodes, child edges to edges, and descendant edges to paths (with at least one edge). In addition, each node n of the twig has a unary condition $c_n(\cdot)$, and $\mu(n)$ must satisfy this condition, namely, $c_n(\mu(n))$ should evaluate to **true**. The simplest conditions are $\lambda(\mu(n)) = l$ (i.e., the label of $\mu(n)$ is l) and **true**; the latter is also known as the *wildcard* and denoted by the symbol $*$. In the twigs t and s of Figure 1, these simple conditions appear as a label or a $*$ inside each node.

The conventional semantics of evaluating a twig t (or any other type of query) is to find the matches of t and their probabilities (e.g., [9, 10]). That is, we need to compute a function p over all the mappings μ, such that $p(\mu)$ is the probability that μ is a match of t in a random possible world. The set of answers comprises all matches μ, such that $p(\mu) > 0$. Given a twig t, a p-document $\tilde{\mathscr{P}}$ and a mapping μ, the probability $p(\mu)$ can be computed efficiently, even if there are cie nodes, based on the following observation. Let s be the minimal subtree of $\tilde{\mathscr{P}}$ that includes the nodes in the image of μ. Observe that $p(\mu)$ is the same as the probability that each distributional node of s chooses *at least* the children that it has in s. (In comparison, the probability $p(s)$ defined at the end of Section 3.1 is that of choosing *exactly* the children that appear in s.) For example, consider the twig t and the p-document $\tilde{\mathscr{P}}$ of Figure 1. Let μ be the match defined by $\mu(n_1) = 1$, $\mu(n_2) = 21$, $\mu(n_3) = 26$ and $\mu(n_4) = 27$. The probability of μ is $0.9 \cdot 0.7 = 0.63$.

However, computing a query that involves projection is not so easy. For example, there are extremely simple conjunctive queries that are intractable over probabilistic relational databases, even if choices of distinct tuples are assumed to be independent [9, 10]. The ultimate usage of projection is to apply the Boolean interpretation. In the case of a twig t and an ordinary document d, it means to determine whether d *satisfies* t, denoted by $d \models t$; that is, to decide whether there is a match of t in d.

For a given p-document $\tilde{\mathscr{P}}$, we use $\mathscr{P}$ (i.e., without the tilde sign) to denote the random variable that represents a possible world of $\tilde{\mathscr{P}}$. Evaluating a Boolean twig t over $\tilde{\mathscr{P}}$ amounts to computing $\Pr(\mathscr{P} \models t)$, which is the probability that a random possible world of $\tilde{\mathscr{P}}$ satisfies t. By definition, $\Pr(\mathscr{P} \models t)$ is the sum of probabilities of all possible worlds d, such that d satisfies t.

Consider, for example, the p-document $\tilde{\mathscr{P}}$ and the twig t of Figure 1. A possible world of $\tilde{\mathscr{P}}$ satisfies t if it contains either node 23, node 25 or node 27. For each of these three nodes alone, we can compute the probability that it appears in a random possible world, as explained above. But the sum of these three probabilities is not what we are looking for, because these are not disjoint events (i.e., some possible worlds include all three nodes while others include only one or two of them).

In general, evaluating Boolean twigs over p-documents is a hard problem. In [18], it is shown that every nontrivial Boolean twig has an intractable data complexity over p-documents of $\mathsf{PrXML}^{\{\mathsf{cie}\}}$. By definition, a Boolean twig is *trivial* if it has only one node (i.e., it is a condition on the root of the document) or it contains a node with an unsatisfiable condition (i.e., equivalent to **false**). Recall that $\mathrm{FP}^{\#\mathrm{P}}$ is the class of functions that are efficiently computable using an oracle to some function in[3] #P.

Theorem 4.1 [18] *The evaluation of every nontrivial Boolean twig over* $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ *is* $\mathrm{FP}^{\#\mathrm{P}}$*-complete.*

In contrast to Theorem 4.1, [19] shows that over $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$ (i.e., the ProTDB model of [21]), every Boolean twig can be efficiently evaluated under data complexity. In [18], this result is generalized

[2] In the probabilistic documents of [27], the root is distributional. We can assume that a dummy ordinary node is added for compliance with the definition of p-documents.

[3] The functions of #P [26] correspond to NP machines and they count, for a given input, the number of accepting paths. By using an oracle to a #P-hard (or $\mathrm{FP}^{\#\mathrm{P}}$-hard) function, one can efficiently solve the entire polynomial hierarchy [25].

to the family $\mathsf{PrXML}^{\{\mathsf{exp}\}}$. Thus, from Figure 3 and Theorem 4.1, it follows that the family $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ is the maximal one, among those considered in the previous section, that allows efficient evaluation of Boolean twigs.

In [7], it is shown that tractable data complexity carries over to *c-formulae*, which are rather complex queries with aggregate functions. In particular, c-formulae are obtained by mutually nesting twigs and comparisons involving aggregate functions. A simple example of an atomic c-formula is $(\mathsf{count}(s)\ \theta\ R)$, where s is a *selector*, R is a rational number and θ is one of the operators $<, >, \leq, \geq, =$ and $\neq$. The selector s is a twig that computes a set of node ids, when given a document as input. The aggregate function count is applied to the set computed by s and the result is compared with R.

The aggregate functions min and max are also allowed in c-formulae, provided that labels are interpreted as numeric values. Another aggregate function is ratio. The simple atomic c-formulae that use ratio have the form $(\mathsf{ratio}(s,t)\ \theta\ R)$, where s, θ and R are as above, and t is a Boolean twig. The function $\mathsf{ratio}(s,t)$ is interpreted in a given document d as the ratio $|U|/|S|$, where S is the set of nodes that are selected by s, and U is the subset of S comprising all nodes u, such that the subtree of d rooted at u satisfies t.

Theorem 4.2 [7] *Let q^A be a c-formula that uses the aggregate functions* count, ratio, min *and* max. *The evaluation of q^A over $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ is in polynomial time.*[4]

In [7], it is also shown that Theorem 4.2 does not generalize to the aggregate functions sum and avg. For instance, it is intractable to compute the probability of the following event: the total sum (or the average) of all the numeric labels in a random possible world is zero. In fact, this probability cannot even be efficiently approximated unless[5] NP=RP.

In comparison to Theorem 4.2, the evaluation algorithms of [18, 19] apply only to twigs (which are a subclass of c-formulae), but they are more efficient. In particular, the algorithms of [18, 19] are *fixed-parameter tractable*[6] [11, 22], whereas that of [7] is not.

[4]Similarly to [23], the numerical operands of the query q^A are not assumed to be fixed; rather, they are given as part of the input.

[5]Note that NP=RP implies that the whole polynomial hierarchy is recognizable by an efficient randomized algorithm with a bounded two-sided error (BPP) [28].

[6]In the function that gives the running time, the size of the query effects only the constant, but not the degree of the polynomial.

Finally, in [7, 18, 19], it is shown how to apply their results to non-Boolean queries, namely, projection can be used (in the type of queries they consider), but not necessarily in a total manner.

4.1 Approximate Query Evaluation

Let t be a twig. In the context of evaluating t over a p-document, a *fully polynomial randomized approximation scheme* (*FPRAS*) for t is a randomized algorithm A that, given a p-document $\tilde{\mathscr{P}}$ and an $\epsilon > 0$, returns a number $A(\tilde{\mathscr{P}}, \epsilon)$, such that[7]

$$\Pr\left((1-\epsilon)p \leq A(\tilde{\mathscr{P}}, \epsilon) \leq (1+\epsilon)p\right) \geq \frac{3}{4},$$

where $p = \Pr(\mathscr{P} \models t)$. Moreover, the running time of A is polynomial in $\tilde{\mathscr{P}}$ and in $1/\epsilon$.

In [18], the Monte-Carlo approximation technique of [17] is used (as similarly done in [10]) in order to show that twigs can be efficiently approximated over the maximal family considered in Figure 3.

Theorem 4.3 [18] *Every twig has an FPRAS over $\mathsf{PrXML}^{\{\mathsf{exp,cie}\}}$.*

By combining a simple twig and any aggregate function (among those considered above), it is possible to get a query that cannot be efficiently approximated over $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ (for all $\epsilon > 0$), unless NP=RP. This is proved rather easily by using the following observation. It is NP-hard to test, for a given p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{cie}\}}$, whether the following event has a non-zero probability: the twig a/b is *not* satisfied by a random possible world of $\tilde{\mathscr{P}}$.

4.2 Enumerating Matches of Twigs

By Theorem 4.2, twig queries can be efficiently evaluated under data complexity. This is no longer true under *query-and-data* complexity (i.e., both the p-document and the query are given as input). The reason for that is the following. It is NP-complete to determine *non-emptiness*, that is, to decide whether there is at least one match of a twig t in some possible world of $\mathscr{P}$, where both t and $\tilde{\mathscr{P}}$ are part of the input; moreover, this is true even if $\tilde{\mathscr{P}}$ is in $\mathsf{PrXML}^{\{\mathsf{mux}\}}$ and the twig t has no descendant edges [18]. Observe that this result is about matches of t, which means that there is no projection.

Query-and-data complexity is instrumental in analyzing the efficiency of an algorithm relative to the output size. We consider the enumeration of all μ and their probabilities $p(\mu)$, where μ is a match of the twig t in some possible world of $\tilde{\mathscr{P}}$ (hence,

[7]Note that the choice of the reliability factor 3/4 is arbitrary, since for a given $\delta > 0$, one can enhance the reliability to $(1-\delta)$ by taking the median of $O(\log \delta)$ trials [15].

$p(\mu) > 0$). Under query-and-data complexity, the number of matches can be exponential in the size of the input. Consequently, "polynomial time in the size of the input" is not a suitable yardstick. Instead, other measures of efficiency are used in the literature. The common one is *polynomial total time*, namely, the running time is polynomial in the combined size of the input and the output. A stronger notion is *enumeration in incremental polynomial time* [16], namely, the ith answer is generated in time that is polynomial in the size of the input and that of the previous $i-1$ answers. The above NP-completeness result of [18] implies that it is intractable to enumerate all the matches μ of a given twig t in the possible worlds of a p-document $\tilde{\mathscr{P}}$. Rather surprisingly, [19] shows that this task (and even a generalized one) can be done efficiently if *maximal* matches (rather than the ordinary *complete* matches) are allowed.

Formally, a *partial match* of a twig t in a document d is a match of some t' in d, where t' is a subtree of t that has the same root as t. In [19], the following problem was considered. Given a twig pattern t, a p-document $\tilde{\mathscr{P}}$ and a threshold $p \in (0,1]$, enumerate all the *maximal matches w.r.t. p*, namely, all the partial matches μ, such that *(1)* the probability of μ is at least p, *and (2)* no partial match $\mu' \neq \mu$ with probability at least p subsumes μ.

As an example, consider the p-document $\tilde{\mathscr{P}}$ and the twig s of Figure 1. Let $p = 0.4$. The partial matches μ_1, μ_2 and μ_3 are defined as follows.

- $\mu_1(n_5) = 1$, $\mu_1(n_6) = 3$.
- $\mu_2(n_5) = 1$, $\mu_2(n_6) = 3$, $\mu_2(n_7) = 5$.
- $\mu_3(n_5) = 1$, $\mu_3(n_6) = 3$, $\mu_3(n_7) = 5$, $\mu_3(n_8) = 7$.

Note that μ_1 is subsumed by μ_2, and both μ_1 and μ_2 are subsumed by μ_3. The partial matches μ_1, μ_2 and μ_3 have the probabilities 0.8, $0.8 \cdot 0.6 = 0.48$ and $0.8 \cdot 0.6 \cdot 0.7 = 0.336$, respectively. Hence, μ_1 is not maximal w.r.t. p, since it is subsumed by μ_2 which has a probability above p. But μ_2 is maximal, because it is only subsumed by μ_3 which has a probability smaller than p. In [19], the following is proved.

Theorem 4.4 [19] *The following is in incremental polynomial time. Given a p-document $\tilde{\mathscr{P}} \in \mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$, a twig t and a threshold $p > 0$, enumerate all the maximal matches of t in $\tilde{\mathscr{P}}$ w.r.t. p.*

Whether Theorem 4.4 can be generalized to either $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ or $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ is yet unknown. We note that the operation of *maximally joining probabilistic relations* [20] is essentially the probabilistic version of *full disjunctions* [12]. This operation is very similar to that of enumerating the maximal matches of a twig in a p-document, but its general case is intractable [20] (whereas full disjunctions are computable with polynomial delay [6]).

5 Asserting Constraints

The type cie of distributional nodes is unique in its ability to represent correlations between choices of children at different distributional nodes. However, as discussed in the previous section, this type extremely limits the ability to evaluate queries that involve projection. In contrast, the other types (namely, exp, ind, mux and det) allow efficient evaluation of highly expressive queries. But they also entail an inherent assumption of probabilistic independence. This assumption often severely limits the ability to model real-life data.

As an example, consider again the document $\tilde{\mathscr{P}}$ of Figure 1. This p-document can be thought of as a fragment of a probabilistic database that represents the result of screen scraping a university Web site. In particular, each probability embodies the degree of certainty, in making some specific choice, as determined during the screen-scraping process. In addition to these probabilities, available information about the university can imply intricate correlations between different choices. For example, suppose that the following facts are known to hold.

1. Every department with three or more members has a chair (and there is at most one chair).
2. A chair must be a full professor.
3. At least 95% of the associate professors have at most two Ph.D. students.

This information implies that many possible worlds of $\tilde{\mathscr{P}}$ are actually inconceivable. Moreover, rather intricate correlations exist between the entities of $\tilde{\mathscr{P}}$. For instance, given that Lisa is a faculty member, there are three individuals that can be chosen as her students. These choices, however, are not probabilistically independent, because if Lisa is an associate professor, then it is most likely that she has no more than two students. Moreover, the second fact implies that these choices also depend on whether Lisa is the chair. Lastly, the first two facts imply that the academic rank of Lisa probabilistically depends on the likelihood that Mary is the chair, and also on whether Paul is a member of the department (because if there are three members, then there is a higher chance that Lisa is the chair and, hence, a full professor).

One way of incorporating the available information is to use cie nodes and correlate them by means of shared event variables. However, it is not clear how (and whether) this can be done efficiently (i.e., the

resulting p-document should not be too large). Moreover, even if that could be done efficiently, using cie nodes has its own limitations (as discussed earlier). A similar problem exists in other known models, such as those based on *Bayesian networks* (where approximating the probability of simple events is intractable [8]). A direct and convenient approach is proposed in [7], namely, representing the probability space of possible worlds by means of a *PXDB*.

A *PXDB* is a px-space that is represented by a pair $(\tilde{\mathscr{P}}, \mathscr{C})$, where $\tilde{\mathscr{P}}$ is a p-document of $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ (which, by Figure 3, effectively allows all the types of distributional nodes, except for cie) and $\mathscr{C}$ is a set of *constraints*, such as the above three facts. The px-space $\tilde{\mathscr{D}}$ that is described by $(\tilde{\mathscr{P}}, \mathscr{C})$ is *well defined* if there is a nonzero probability that a random document of $\tilde{\mathscr{P}}$ satisfies $\mathscr{C}$. In that case, $\tilde{\mathscr{D}}$ is the subspace of $\tilde{\mathscr{P}}$ that comprises all the possible worlds that satisfy each of the constraints (and, as usual, $\mathscr{D}$ is the random variable associated with $\tilde{\mathscr{D}}$). In particular, for a document d that satisfies $\mathscr{C}$ (denoted by $d \models \mathscr{C}$), the probability $\Pr(\mathscr{D} = d)$ is given by

$$\Pr(\mathscr{D} = d) = \Pr(\mathscr{P} = d \mid \mathscr{P} \models \mathscr{C}) = \frac{\Pr(\mathscr{P} = d)}{\Pr(\mathscr{P} \models \mathscr{C})}.$$

The above three facts about the university of Figure 1 can be easily expressed as c-formulae that use the aggregate functions count and ratio (recall that c-formulae were discussed in Section 4). The importance of this observation lies in Theorem 4.2. In particular, it is shown in [7] how Theorem 4.2 can be used for obtaining the following result.

Theorem 5.1 [7] *Let $\mathscr{C}$ be a fixed set of c-formulae that use the aggregate functions* count, ratio, min *and* max. *The following three tasks can be performed efficiently, given a PXDB $\tilde{\mathscr{D}} = (\tilde{\mathscr{P}}, \mathscr{C})$.*

- *Testing well-definedness of $\tilde{\mathscr{D}}$.*
- *Evaluating a twig (or a c-formulae with the above aggregate functions) over $\tilde{\mathscr{D}}$.*
- *Sampling $\tilde{\mathscr{D}}$.*

Sampling a PXDB $\tilde{\mathscr{D}}$ is the task of emulating $\tilde{\mathscr{D}}$ by randomly generating a document of $\tilde{\mathscr{D}}$, such that the probability of generating each document d is equal to $\Pr(\mathscr{D} = d)$.

Observe that Theorem 5.1 makes the limiting (yet necessary) assumption that the set $\mathscr{C}$ of constraints is fixed (however, the numerical values that appear in $\mathscr{C}$ are given as part of the input). Therefore, one can effectively utilize this result when the correlations between the represented entities are expressed by a small set of facts (e.g., as in the above example of a university).

6 Concluding Remarks

The families $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ and $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ (of [2, 24]) exhibit a clear tradeoff between the efficiency of query evaluation and the ability to model correlations between probabilistic choices. $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ is the most expressive family among those that do not have cie nodes (see Figure 3). In this family, highly expressive queries can be evaluated efficiently. But this is achieved at the expense of assuming that choices of children by different distributional nodes are independent. In comparison, $\mathsf{PrXML}^{\{\mathsf{cie}\}}$ can express correlations between distributional nodes by means of shared event variables; however, evaluation of queries with projection (even very simple ones) is intractable.

Approximate query evaluation partly surmounts the limitation entailed by the above tradeoff. Specifically, for twig queries with projection, efficient (multiplicative) approximate evaluation is realizable in the most expressive family, namely, $\mathsf{PrXML}^{\{\mathsf{exp},\mathsf{cie}\}}$. But this solution is possible only because twig queries are monotonic. In particular, query evaluation becomes inapproximable if negation can be applied to branches (e.g., "find all departments that do not have a chair"). The PXDB model takes a completely different approach. It describes correlations in a p-document of $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ in terms of a fixed set of constraints (phrased as c-formulae), rather than by many specific dependencies among distributional nodes (as can be done in $\mathsf{PrXML}^{\{\mathsf{cie}\}}$). This is a natural approach, because correlations are quite frequently a facet of integrity constraints. Interestingly, the PXDB approach demonstrates the following phenomenon. When a dependency-free probabilistic data model is coupled with a powerful query language, it becomes a realistic framework capable of expressing complex correlations among entities, without sacrificing efficiency.

The above tractability results for $\mathsf{PrXML}^{\{\mathsf{exp}\}}$ hold only under data complexity. Under query-and-data complexity, query evaluation becomes hard even for projection-free twigs over $\mathsf{PrXML}^{\{\mathsf{mux}\}}$. Nevertheless, [19] shows that one can enumerate in incremental polynomial time all the maximal matches of a twig (w.r.t. a threshold) in a p-document of $\mathsf{PrXML}^{\{\mathsf{ind},\mathsf{mux}\}}$.

In [2, 24], various aspects of managing probabilistic XML are studied. Their work is couched in the value-based semantics and the focus is on the family $\mathsf{PrXML}^{\{\mathsf{cie}\}}$; in addition, [2] also considers $\mathsf{PrXML}^{\{\mathsf{ind}\}}_{|\mathscr{V}|}$. They model and investigate updates in probabilistic XML. For example, an insertion into a given document is defined by a triple (t, n, d), where t is a twig pattern, n is a node of t and d is a tree that needs

to be added to each node v, such that some match of t in the given document maps n to v. In the setting of probabilistic data, an update modify the possible worlds, and the goal is to represent them by a new p-document. The work of [2, 24] shows how it can be done. Other problems studied in [24] are those of determining whether two p-documents are equivalent, and eliminating random possible worlds characterized by probabilities that are too low. Finally, they consider the problem of applying cardinality constraints to a given p-document and representing the result by means of a new p-document. Their cardinality constraints are a limited, order-unaware form of DTD constraints.

The *PIXml* model of [13] describes probabilistic choices similarly to $\mathsf{PrXML}^{\{\mathsf{exp}\}}_{|\not\perp}$. However, PIXml significantly deviates from p-documents in two aspects. First, the probability space as well as the possible worlds are represented by directed acyclic graphs, rather than trees. Second, the probabilities of choosing subsets of children are defined by intervals, rather than exact values.

References

[1] S. Abiteboul, B. Kimelfeld, Y. Sagiv, and P. Senellart. On the expressiveness of probabilistic XML models. Submitted for a journal publication, 2008.

[2] S. Abiteboul and P. Senellart. Querying and updating probabilistic information in XML. In *EDBT*, pages 1059–1068, 2006.

[3] S. Amer-Yahia, S. Cho, L. V. S. Lakshmanan, and D. Srivastava. Minimization of tree pattern queries. In *SIGMOD*, pages 497–508, 2001.

[4] O. Benjelloun, A. D. Sarma, A. Y. Halevy, M. Theobald, and J. Widom. Databases with uncertainty and lineage. *VLDB J.*, 17(2):243–264, 2008.

[5] N. Bruno, N. Koudas, and D. Srivastava. Holistic twig joins: optimal XML pattern matching. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, pages 310–321. ACM, 2002.

[6] S. Cohen, I. Fadida, Y. Kanza, B. Kimelfeld, and Y. Sagiv. Full disjunctions: Polynomial-delay iterators in action. In *VLDB*, pages 739–750. ACM, 2006.

[7] S. Cohen, B. Kimelfeld, and Y. Sagiv. Incorporating constraints in probabilistic XML. In *PODS*, pages 109–118, 2008.

[8] P. Dagum and M. Luby. Approximating probabilistic inference in bayesian belief networks is NP-hard. *Artif. Intell.*, 60(1):141–153, 1993.

[9] N. N. Dalvi and D. Suciu. The dichotomy of conjunctive queries on probabilistic structures. In *PODS*, pages 293–302, 2007.

[10] N. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *VLDB J.*, 16(4):523–544, 2007.

[11] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.

[12] C. A. Galindo-Legaria. Outerjoins as disjunctions. In *SIGMOD*, pages 348–358. ACM Press, 1994.

[13] E. Hung, L. Getoor, and V. S. Subrahmanian. Probabilistic interval XML. In *ICDT*, pages 361–377, 2003.

[14] E. Hung, L. Getoor, and V. S. Subrahmanian. PXML: A probabilistic semistructured data model and algebra. In *ICDE*, pages 467–478, 2003.

[15] M. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.

[16] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.

[17] R. M. Karp, M. Luby, and N. Madras. Monte-carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 10(3):429–448, 1989.

[18] B. Kimelfeld, Y. Kosharovsky, and Y. Sagiv. Query efficiency in probabilistic XML models. In *SIGMOD Conference*, pages 701–714, 2008.

[19] B. Kimelfeld and Y. Sagiv. Matching twigs in probabilistic XML. In *VLDB*, pages 27–38, 2007.

[20] B. Kimelfeld and Y. Sagiv. Maximally joining probabilistic data. In *PODS*, pages 303–312. ACM, 2007.

[21] A. Nierman and H. V. Jagadish. ProTDB: Probabilistic data in XML. In *VLDB*, pages 646–657, 2002.

[22] C. H. Papadimitriou and M. Yannakakis. On the complexity of database queries. *Journal of Computer and System Sciences*, 58(3):407–427, 1999.

[23] C. Ré and D.Suciu. Efficient evaluation of HAVING queries on a probabilistic database. In *DBPL*, volume 4797 of *Lecture Notes in Computer Science*, pages 186–200. Springer, 2007.

[24] P. Senellart and S. Abiteboul. On the complexity of managing probabilistic XML data. In *PODS*, pages 283–292, 2007.

[25] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 21(2):316–328, 1992.

[26] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[27] M. van Keulen, A. de Keijzer, and W. Alink. A probabilistic XML approach to data integration. In *ICDE*, pages 459–470. IEEE Computer Society, 2005.

[28] S. Zachos. Probabilistic quantifiers and games. *Journal of Computer and System Sciences*, 36(3):433–451, 1988.

On Query Algebras for Probabilistic Databases*

Christoph Koch

Department of Computer Science
Cornell University, Ithaca, NY
koch@cs.cornell.edu

Abstract

This article proposes a core query algebra for probabilistic databases. In essence, this core is part of the query languages of most probabilistic database systems proposed so far, but is sometimes hidden in complex language definitions. We give a formal definition of the algebra and illustrate it by examples. We then survey the current state of knowledge regarding the expressive power and complexity of this core.

1 Introduction

The emerging research area of probabilistic databases has attracted much interest and excitement recently. It is still quite early in the development of this field, and the community has not yet converged upon a standard set of features or use cases that probabilistic databases and their query languages should support. However, a body of foundational knowledge on query languages for probabilistic databases is forming rapidly. The aim of this article is to give a concise summary of this foundation.

We will focus on extracting and discussing a core query algebra that arguably can be found completely or mostly implemented in most probabilistic database systems developed so far, including MystiQ [9], Trio [21], MayBMS [3, 15], and MCDB [12]. This algebra is *probabilistic world-set algebra* [5, 14, 15, 16].

Agreeing on relational algebra as a core language for relational database systems was one of the foundations of their success: It has facilitated the development of a widely agreed-upon terminology which allowed the database research community to make rapid progress; but it is also the interface between query optimization and query evaluation at the very heart database systems. Part of the rationale for proposing a core algebra for probabilistic databases is, of course, the hope that it will help us replicate our previous success with relational databases in the field of probabilistic databases.

It seems proper to start the search for a such a core with the definition of *design desiderata* for probabilistic database query languages. Ours are the following:

1. Efficient query evaluation.

2. The right degree of expressive power. The language should be powerful enough to support important queries. On the other hand, it should not be too strong, because expressiveness generally comes at a price: high evaluation complexity and infeasibility of query optimization. Can a case be made that some language is in a natural way a probabilistic databases analog of the relationally complete languages (such as relational algebra) – an expressiveness yardstick?

3. Genericity. The semantics of a query language should be independent from details of how the data is represented. Queries should behave in the same way no matter how the probabilistic data is stored. This is a basic requirement that is even part of the traditional definition of what constitutes a query (cf. e.g. [1]), but it is nontrivial to achieve for probabilistic databases [5, 4]. Genericity is key to making the language applicable to many different database systems that internally represent data in different ways.

4. The ability to transform data. Queries on probabilistic databases are often interpreted quite narrowly in the literature. It is the author's view that queries in general should be compositional mappings between databases, in this case probabilistic databases. This is a property taken for granted in relational databases. It allows for the definition of clean database update languages.

5. The ability to introduce uncertainty. This may appear to be a controversial goal, since uncertainty is commonly considered undesirable, and probabilistic databases are there to deal with it by providing useful functionality *despite* uncertainty. An uncertainty-introduction operation is

*Database Principles Column. Column editor: Leonid Libkin, School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, UK. E-mail: libkin@inf.ed.ac.uk

important for compositionality, to allow the construction of an uncertain database from scratch (as part of the update language), and to support hypothetical (what-if) queries.

Probabilistic world-set algebra is a minimal extension of relational algebra that arguably satisfies all of the desiderata presented above.

Content of this article. After a short definition of the conceptual model of probabilistic databases used throughout most of the article (discrete at first) in Section 2, the algebra is formally defined in Section 3. Section 4 illustrates probabilistic world-set algebra and its SQL-like syntax by several examples. Section 5 and 6 discuss our current state of knowledge regarding the expressive power and complexity, respectively, of the algebra. Section 7 discusses extensions of the algebra (such as aggregates) and queries on probabilistic databases with continuous distributions.

2 Probabilistic Databases

Informally, our model of probabilistic databases is the following. The schema of a probabilistic database is simply a relational database schema. Given such a schema, a probabilistic database is a finite set of database instances of that schema (called possible worlds), where each world has a weight (called probability) between 0 and 1 and the weights of all worlds sum up to 1. In a subjectivist Bayesian interpretation, one of the possible worlds is "true", but we do not know which one, and the probabilities represent degrees of belief in the various possible worlds. Note that this is only the conceptual model. The physical representation of the set of possible worlds in probabilistic database management systems is quite different [9, 21, 3].

Given a schema with relation names $R_1, \ldots, R_k$. We use $sch(R_l)$ to denote the attributes of relation schema R_l. Formally, a *probabilistic database* is a *finite* set of structures

$$\mathbf{W} = \{\langle R_1^1, \ldots, R_k^1, p^{[1]}\rangle, \ldots, \langle R_1^n, \ldots, R_k^n, p^{[n]}\rangle\}$$

of relations $R_1^i, \ldots, R_k^i$ and numbers $0 < p^{[i]} \leq 1$ such that

$$\sum_{1 \leq i \leq n} p^{[i]} = 1.$$

We call an element $\langle R_1^i, \ldots, R_k^i, p^{[i]}\rangle \in \mathbf{W}$ a *possible world*, and $p^{[i]}$ its probability. We use superscripts for indexing possible worlds. To avoid confusion with exponentiation, we sometimes use bracketed superscripts $.^{[i]}$. We call a relation R *complete* or *certain* if its instantiations are the same in all possible worlds of $\mathbf{W}$, i.e., if $R^1 = \cdots = R^n$.

The definitions of the following sections are applicable if either a set- or a multiset-based semantics for relations is used. Whenever the distinction causes an subtleties, they will be made clear. Of course, when we use SQL-like syntax for queries, we automatically assume multiset semantics.

Tuple *confidence* refers to the probability of the event $\vec{t} \in R$, where R is one of the relation names of the schema, with

$$\Pr[\vec{t} \in R] = \sum_{1 \leq i \leq n:\ \vec{t} \in R^i} p^{[i]}.$$

3 Core Algebra

This section defines *probabilistic world-set algebra* (probabilistic WSA) [5, 14, 16]. Informally, probabilistic world-set algebra consists of the operations of relational algebra, an operation for computing tuple confidence conf, and the repair-key operation for *introducing* uncertainty.

- The operations of relational algebra are evaluated individually, in "parallel", in each possible world.
- The operation $\text{conf}(R)$ computes, for each tuple that occurs in relation R in at least one world, the sum of the probabilities of the worlds in which the tuple occurs. The result is a certain relation, or viewed differently, a relation that is the same in all possible worlds.
- Finally, $\text{repair-key}_{\vec{A}@P}(R)$, where $\vec{A}, P$ are attributes of R, conceptually nondeterministically chooses a maximal repair of key $\vec{A}$. This operation turns a possible world R^i into the set of worlds consisting of all possible *maximal repairs* of key $\vec{A}$. A repair of key $\vec{A}$ in relation R^i is a subset of R^i for which $\vec{A}$ is a key. It uses the numerically-valued column P for weighting the newly created alternative repairs.

We define the semantics of probabilistic world-set algebra formally using a function $[\![\cdot]\!]_{pw}$ that maps between sets of possible worlds. For a further illustration that the reader may find more intuitive, we will also provide a Monte Carlo/sampling semantics definition $[\![\cdot]\!]_{mc}$. Conceptually, $[\![Q]\!]_{mc}$ computes for query Q a sample result relation. Note that the definition of $[\![\cdot]\!]_{mc}$ is nondeterministic, an multiple invocations will yield different results. The sampling semantics also only approximates $[\![\cdot]\!]_{pw}$; we will not have space to make this precise, but will point to relevant literature.

Relational algebra. The operations of relational algebra (selection σ, projection π, product $\times$, union $\cup$, difference $-$, and attribute renaming ρ) are applied in each possible world independently.

The possible-worlds semantics of unary and binary relational algebra operations Θ on probabilistic database $\mathbf{W}$ is

$$[\![\Theta(R_l[, R_m])]\!]_{pw}(\mathbf{W}) := \{\langle R_1, \ldots, R_k, \Theta(R_l[, R_m]), p\rangle \mid \langle R_1, \ldots, R_k, p\rangle \in \mathbf{W}\}.$$

The sampling semantics $[\![\Theta(Q_1[,Q_2])]\!]_{mc}$ is simply $\Theta([\![Q_1]\!]_{mc}[,[\![Q_2]\!]_{mc}])$.

Selection conditions are Boolean combinations of atomic conditions (i.e., negation is permitted even in the positive fragment of the algebra). Arithmetic expressions may occur in atomic conditions and in the arguments of π and ρ. For example, $\rho_{A+B\to C}(R)$ in each world adds up the A and B values of each tuple of R and keeps them in a new C attribute.

Confidence. The semantics of the tuple confidence operation is

$$[\![\mathrm{conf}(R_l)]\!]_{pw}(\mathbf{W}) := \{\langle R_1,\ldots,R_k,S,p\rangle \mid \langle R_1,\ldots,R_k,p\rangle \in \mathbf{W}\}$$

where

$$S = \Big\{\langle \vec{t}, \Pr[\vec{t}\in R_l]\rangle \mid \vec{t}\in \bigcup_{i=1}^{n} R_l^i\Big\}.$$

The result of $\mathrm{conf}(R_l)$, the relation S, is the same in all possible worlds, i.e., it is a certain relation.

By our definition of probabilistic databases, each possible world has nonzero probability. As a consequence, conf does not return tuples with probability 0. Note also that conf implicitly eliminates duplicates.

Example 3.1 On probabilistic database

R^1	A	B
	a	b
	b	c

$p^{[1]} = .3$

R^2	A	B
	a	b
	c	d

$p^{[2]} = .7$

$\mathrm{conf}(R)$ computes

conf(R)	A	B	P
	a	b	1
	b	c	.3
	c	d	.7

i.e., for each possible tuple, the sum of the weights of the possible worlds in which it occurs. □

Let $S_1,\ldots,S_m$ be samples from $[\![Q]\!]_{mc}$, that is, the results of m separate invocations of $[\![Q]\!]_{mc}$. Then we compute $[\![\mathrm{conf}(Q)]\!]_{pw}$ as

$$\Big\{\langle \vec{t}, |\{i : \vec{t}\in S_i\}|/m\rangle : \vec{t}\in \bigcup_{i=1}^{m} S_i\Big\}.$$

Repair-key. The uncertainty-introducing operation *repair-key* can be thought of as sampling a maximum repair of a key for a relation. Repairing a key of a complete relation R means to compute, as possible worlds, all subset-maximal relations obtainable from R by removing tuples such that a key constraint is satisfied. We will use this as a method for constructing probabilistic databases, with probabilities derived from relative weights attached to the tuples of R.

We say that relation R' is a *maximal repair* of a functional dependency (fd, cf. [1]) for relation R if R' is a maximal subset of R which satisfies that functional dependency, i.e., a subset $R'\subseteq R$ that satisfies the fd such that there is no relation R'' with $R'\subset R''\subseteq R$ that satisfies the fd.

Let $\vec{A},B \in sch(R_l)$. For each possible world $\langle R_1,\ldots,R_k,p\rangle\in\mathbf{W}$, let column B of R contain only numerical values greater than 0 and let R_l satisfy the fd $(sch(R_l)-B)\to sch(R_l)$. Then,

$$[\![\text{repair-key}_{\vec{A}@B}(R_l)]\!]_{pw}(\mathbf{W}) := \Big\{\langle R_1,\ldots,R_k,\pi_{sch(R_l)-B}(\hat{R}_l),\hat{p}\rangle \mid \langle R_1,\ldots,R_k,p\rangle\in\mathbf{W},\ \hat{R}_l \text{ is a maximal repair of fd } \vec{A}\to sch(R_l),\ \hat{p} = p\cdot\prod_{\vec{t}\in\hat{R}_l}\frac{\vec{t}.B}{\sum_{\vec{s}\in R_l:\vec{s}.\vec{A}=\vec{t}.\vec{A}}\vec{s}.B}\Big\}$$

Such a repair operation, apart from its usefulness for the purpose implicit in its name, is a powerful way of constructing probabilistic databases from complete relations.

The sampling semantics makes this operation more intuitive: Conceptually, given a sample R from $[\![Q]\!]_{mc}$, we group the tuples of R by the columns $\vec{A}$: for each distinct $\vec{a}$ in $\pi_{\vec{A}}(R)$, we independently sample exactly one tuple $\vec{t}$ from group $G_{\vec{a}} = \sigma_{\vec{A}=\vec{a}}(R)$ with the probability distribution given by (normalized) column B,

$$\Pr[\text{choose } \vec{t} \text{ from group } \vec{a}] = \vec{t}.B/\sum_{\vec{t}'\in G_{\vec{a}}}\vec{t}'.B.$$

Example 3.2 Consider the example of tossing a biased coin twice. We start with a certain database

R	Toss	Face	FProb
	1	H	.4
	1	T	.6
	2	H	.4
	2	T	.6

$p = 1$

that represents the possible outcomes of tossing the coin twice. We turn this into a probabilistic database that represents this information using alternative possible worlds for the four outcomes using the query $S :=$ repair-key$_{\mathrm{Toss@FProb}}(R)$. The resulting possible worlds are

S^1	Toss	Face
	1	H
	2	H

S^2	Toss	Face
	1	H
	2	T

S^3	Toss	Face
	1	T
	2	H

S^4	Toss	Face
	1	T
	2	T

with probabilities $p^{[1]} = p\cdot\frac{.4}{.4+.6}\cdot\frac{.4}{.4+.6} = .16$, $p^{[2]} = p^{[3]} = .24$, and $p^{[4]} = .36$. □

Coins	Type	Count
	fair	2
	2headed	1

Faces	Type	Face	FProb
	fair	H	.5
	fair	T	.5
	2headed	H	1

Tosses	Toss
	1
	2

R^f	Type
	fair

R^{dh}	Type
	2headed

$S^{f.HH}$	Type	Toss	Face
	fair	1	H
	fair	2	H

$p^{f.HH} = 1/6$

$S^{f.HT}$	Type	Toss	Face
	fair	1	H
	fair	2	T

$p^{f.HT} = 1/6$

$S^{f.TH}$	Type	Toss	Face
	fair	1	T
	fair	2	H

$p^{f.TH} = 1/6$

$S^{f.TT}$	Type	Toss	Face
	fair	1	T
	fair	2	T

$p^{f.TT} = 1/6$

S^{dh}	Type	Toss	Face
	2headed	1	H
	2headed	2	H

$p^{dh} = 1/3$

Ev	Toss	Face
	1	H
	2	H

Q	Type	P
	fair	$(1/6)/(1/2) = 1/3$
	2headed	$(1/3)/(1/2) = 2/3$

Figure 1: Tables of Example 4.1.

The fragment of probabilistic WSA which excludes the difference operation is called *positive* probabilistic WSA.

Computing possible and certain tuples is redundant with conf:

$$\begin{aligned} \text{poss}(R) &:= \pi_{sch(R)}(\text{conf}(R)) \\ \text{cert}(R) &:= \pi_{sch(R)}(\sigma_{P=1}(\text{conf}(R))) \end{aligned}$$

4 Examples

4.1 Adding Evidence

Example 4.1 A bag of coins contains two fair coins and one double-headed coin. We take one coin out of the bag but do not look at its two faces to determine its type (fair or double-headed) for certain. Instead we toss the coin twice to collect evidence about its type.

We start with a complete database (i.e., a relational database, or a probabilistic database with one possible world of probability 1) consisting of three relations, Coins, Faces, and Tosses (see Figure 1 for all tables used in this example). We first pick a coin from the bag and model that the coin be either fair or double-headed. In probabilistic WSA this is expressed as

$$R := \text{repair-key}_{\emptyset@\text{Count}}(\text{Coins}).$$

This results in a probabilistic database of two possible worlds, $\langle \text{Coins}, \text{Faces}, \text{Tosses}, R^f, p^f = 2/3\rangle$ and $\langle \text{Coins}, \text{Faces}, \text{Tosses}, R^{dh}, p^{dh} = 1/3\rangle$.

The possible outcomes of tossing the coin twice can be modeled as

$$S := \text{repair-key}_{\text{Toss@FProb}}(R \bowtie \text{Faces} \times \text{Tosses}).$$

This turns the two possible worlds into five, since there are four possible outcomes of tossing the fair coin twice, and only one for the double-headed coin.

Let $T := \pi_{\text{Toss,Face}}(S)$. The posterior probability that a coin of type x was picked, given the *evidence* Ev (see Figure 1) that both tosses result in H, is

$$\Pr[x \in R \mid T = Ev] = \frac{\Pr[x \in R \wedge T = Ev]}{\Pr[T = Ev]}.$$

Let A be a relational algebra expression for the Boolean query $T = Ev$. Then we can compute a table of pairs $\langle x, \Pr[x \in R \mid T = Ev]\rangle$ as

$$\begin{aligned} Q := \pi_{\text{Type},P_1/P_2 \to P}(\\ \rho_{P\to P_1}(\text{conf}(R \times A)) \times \rho_{P\to P_2}(\text{conf}(A))). \end{aligned}$$

The prior probability that the chosen coin was fair was $2/3$; after taking the evidence from two coin tosses into account, the posterior probability Pr[the coin is fair | both tosses result in H] is only $1/3$. Given the evidence from the coin tosses, the coin is now more likely to be double-headed. □

4.2 Hypothetical Queries: Skills Management

For the second example, we use an SQL-like syntax for probabilistic WSA. The mapping is in strict analogy with that from relational algebra to SQL. Repair-key is a new operation whose syntax should be intuitive. Confidence computation (or strictly speaking, the combination of confidence computation and projection, $\text{conf}(\pi_{\vec{A}}(R))$) has become an aggregate, which conveys the intuition that duplicates are eliminated: for each group, only one tuple with a probability is returned.

Example 4.2 Given a relational database representing companies, employees, and their skills such as the following.

CE	CID	EID
	LEH	Bob
	LEH	Joe
	MER	Dan
	MER	Bill
	MER	Fred

ES	EID	Skill
	Bob	subprime mortgage
	Joe	subprime mortgage
	Dan	junk bonds
	Dan	subprime mortgage
	Bill	risk management
	Fred	junk bonds

We now want to ask the following hypothetical query: Suppose I buy one of the companies and exactly one employee leaves. Which skills do I gain for

certain? Note that this query starts on a traditional relational database (without uncertainty) and returns a certain table. We will create a probabilistic database for intermediate results.

We first choose one company to by and one employee who will leave and compute the employees that will remain in my company.

```
create table RemainingEmployees as
select CE.cid, CE.eid
from   CE,
       (repair key (dummy)
        in (select 1 as dummy, * from CE)) Choice
where CE.cid = Choice.cid
and   CE.eid <> Choice.eid;
```

No probabilities are available, so we will make our choice uniformly. Since we only ask for certain answers in this example, the probabilities actually do not matter.

Next we compute a table of probabilities, for companies and skills, (p1) that I gain the the skill and buy the company, (p2) that I buy the company, and (p1/p2) the conditional probability that I gain the skill if I buy the company.

```
create table Skills as
select Q1.cid, Q1.skill, p1, p2, p1/p2 as p
from (select R.cid, ES.skill, conf() as p1
      from RemainingEmployees R, ES
      where R.eid = ES.eid
      group by R.cid, ES.skill) Q1,
     (select cid, conf() as p2
      from RemainingEmployees
      group by cid) Q2
where Q1.cid = Q2.cid;
```

For the database given above, this results in the table

Skills	CID	Skill	p1	p2	p
	LEH	subprime mortgage	2/5	2/5	1
	MER	junk bonds	3/5	3/5	1
	MER	subprime mortgage	2/5	3/5	2/3
	MER	risk management	2/5	3/5	2/3

The query

```
select cid, skill from Skills where p=1;
```

yields the desired answer. □

5 Expressiveness

The repair-key operation admits an interesting class of queries: Like in Example 4.1, we can start with a probabilistic database of prior probabilities, add further evidence (in Example 4.1, the result of the coin tosses) and then compute interesting posterior probabilities. The adding of further evidence may require extending the hypothesis space first. For this, the repair-key operation is essential. Even though our goal is not to update the database, we have to be able to introduce uncertainty just to be able to model new evidence – say, experimental data. Many natural and important probabilistic database queries cannot be expressed without the repair-key operation. The coin tossing example was admittedly a toy example (though hopefully easy to understand). Real applications such as diagnosis or processing scientific data involve technically similar questions.

Regarding our desiderata, it is quite straightforward to see that probabilistic WSA is generic (3): see also [5]. It is clearly a data transformation query language (4) that supports powerful queries for defining databases. The repair-key operation is our construct for uncertainty introduction (5). The evaluation efficiency (1) of probabilistic WSA is studied in Section 6. The expressiveness desideratum (2) is discussed next.

An expressiveness yardstick. In [5] a non-probabilistic version of WSA is introduced. It replaces the confidence operation with an operation $\text{poss}_{\vec{A}}(Q)$, where $\vec{A}$ is a set of column names of Q, for computing possible tuples. Compared to the poss operation described above, the operation of [5] is more powerful. The operation partitions the set of possible worlds into the groups of those worlds that agree on $\pi_{\vec{A}}(Q)$. The result in each world is the set of tuples possible in Q within the world's group. Thus, this operation supports the grouping of possible worlds just like the group-by construct in SQL supports the grouping of tuples.

The main focus of [5] is to study the fragment of (non-probabilistic) WSA in which repair-key is replaced by the choice-of operation, definable as $\text{choice-of}_{\vec{A}@P}(R) := R \bowtie \text{repair-key}_{\emptyset@P}(\pi_{\vec{A},P}(R))$. The choice-of operation introduces uncertainty like the repair-key operation, but can only cause a polynomial, rather than exponential, increase of the number of possible worlds. This restricted language still allows to express interesting queries; for instance, Example 4.2 is expressible despite the absence of probabilities and the conf operation. This language has the property that query evaluation on enumerative representations of possible worlds is in PTIME (see Section 6 for more on this). Moreover, it is *conservative* over relational algebra in the sense that any query that starts with a certain database (a classical relational database) and produces a certain database is equivalent to a relational algebra query and can be efficiently rewritten into relational algebra. This is a nontrivial result, because in this language we can produce uncertain intermediate results consisting of many possible worlds using the choice-of operator. This allows us to express and efficiently answer hypothetical (what-if) queries.

(Full non-probabilistic) WSA consists of the relational algebra operations, repair-key, and $\text{poss}_{\vec{A}}$. In [16], it is shown that WSA precisely captures second-

order logic. Leaving aside inessential details about interpreting second-order logic over uncertain databases – it can be done in a clean way – this result shows that a query is expressible in WSA if and only if it is expressible in second-order logic. WSA seems to be the first algebraic (i.e., variable and quantifier-free) language known to have exactly the same expressive power as second-order logic.

It can be argued that this establishes WSA as the natural analog of relational algebra for uncertain databases. Indeed, while it is well known that useful queries (such as transitive closure or counting queries, cf. [1]) cannot be expressed in it, relational algebra is a very popular expressiveness yardstick for relational query languages (and query languages that are as expressive as relational algebra are called *relationally complete*). Relational algebra is also exactly as expressive as the *relational calculus* [1]. Second-order logic is just first-order logic extended by (existential) quantification over relations ("Does there exist a relation R such that ϕ holds?", where ϕ is a formula). This is the essence of (what-if) reasoning over uncertain data. For example, the query of Example 4.1 employed what-if reasoning over relations twice via the repair-key operation, first considering alternative choices of coin and then alternative outcomes to coin tossing experiments.

6 Complexity

The core of the algebra, positive relational algebra, can be efficiently evaluated on c-tables. In [3], a version of c-tables called U-relations was developed on which all expressions of this algebra fragment can be evaluated purely in relational algebra.

Properties of the relational-algebra reduction. The relational algebra rewriting down to positive relational algebra on U-relations has a number of nice properties. First, since relational algebra has PTIME (even AC_0) data complexity, the query language of positive relational algebra, repair-key, and poss on probabilistic databases represented by U-relations has the same. The rewriting is in fact a *parsimonious translation*: The number of algebra operations does not increase and each of the operations selection, projection, join, and union remains of the same kind. Query plans are hardly more complicated than the input queries. As a consequence, we were able to observe that off-the-shelf relational database query optimizers do well in practice [3].

Thus, for all but two operations of probabilistic world-set algebra, there is a very efficient solution that builds on relational database technology. The remaining operations are confidence computation and relational algebra difference.

Approximate confidence computation. To compute the confidence in a tuple of data values occurring possibly in several tuples of a U-relation, we have to compute the probability of the disjunction of the local conditions of all these tuples. We have to eliminate duplicate tuples because we are interested in the probability of the data tuples rather than some abstract notion of tuple identity that is really an artifact of our representation. That is, we have to compute the probability of a DNF, i.e., the sum of the weights of the worlds identified with valuations θ of the random variables such that the DNF becomes true under θ. This problem is #P-complete [11, 9]. The result is not the sum of the probabilities of the individual conjunctive local conditions, because they may, intuitively, "overlap".

Confidence computation can be efficiently approximated by Monte Carlo simulation [11, 9, 14]. The technique is based on the Karp-Luby fully polynomial-time randomized approximation scheme (FPRAS) for counting the number of solutions to a DNF formula [13, 8]. There is an efficiently computable unbiased estimator that in expectation returns the probability p of a DNF of n clauses (i.e., the local condition tuples of a Boolean U-relation) such that computing the average of a polynomial number of such Monte Carlo steps (= calls to the Karp-Luby unbiased estimator) is an (ϵ, δ)-approximation for the probability: If the average $\hat{p}$ is taken over at least $\lceil 3 \cdot n \cdot \log(2/\delta)/\epsilon^2 \rceil$ Monte Carlo steps, then $\Pr\big[|p - \hat{p}| \geq \epsilon \cdot p\big] \leq \delta$. The paper [8] improves upon this by determining smaller numbers (within a constant factor from optimal) of necessary iterations to achieve an (ϵ, δ)-approximation.

Avoiding the difference operation. Difference $R - S$ is conceptually simple on c-tables. Without loss of generality, assume that S does not contain tuples $\langle \vec{a}, \psi_1 \rangle, \ldots, \langle \vec{a}, \psi_n \rangle$ that are duplicates if the local conditions are disregarded. (Otherwise, we replace them by $\langle \vec{a}, \psi_1 \vee \cdots \vee \psi_n \rangle$.) For each tuple $\langle \vec{a}, \phi \rangle$ of R, if $\langle \vec{a}, \psi \rangle$ is in S then output $\langle \vec{a}, \phi \wedge \neg\psi \rangle$; otherwise, output $\langle \vec{a}, \phi \rangle$. Testing whether a tuple is possible in the result of a query involving difference is already NP-hard [2]. For U-relations, we in addition have to turn $\phi \wedge \neg\psi$ into a DNF to represent the result as a U-relation. This may lead to an exponentially large output.

In many practical applications, the difference operation can be avoided. Difference is only hard on uncertain relations. On such relations, it can only lead to displayable query results in queries that close the possible worlds semantics using conf, computing a single certain relation. Probably the most important application of the difference operation is for encoding universal constraints, for example in data cleaning. But if the confidence operation is applied on top of a universal query, there is a trick that will often allow to rewrite the query into an existential one (which can be expressed in positive relational algebra plus conf, without difference) [14].

Suppose we compute a conditional probability $\Pr[\phi \mid \psi] = \Pr[\phi \wedge \psi]/\Pr[\psi]$. Here ϕ is existential (expressible in positive relational algebra) and ψ is an equality-generating dependency (i.e., a special universal query) [1]. The trick is to turn relational difference into the subtraction of probabilities, $\Pr[\phi \wedge \psi] =$

Language Fragment	Complexity
On non-succinct representations:	
RA + conf + possible + choice-of	in PTIME (SQL) [14]
RA + possible + repair-key	NP-&coNP-hard [5], in P^{NP} [16]
RA + possible$_Q$ + repair-key	PHIER-compl. [16]
On U-relations:	
Pos.RA + repair-key + possible	in AC0 [3]
RA + possible	co-NP-hard [2]
Conjunctive queries + conf	#P-hard [9]
Probabilistic WSA	in $P^{\#P}$ [14]
Pos.RA + repair-key + possible + approx.conf + egds	in PTIME [14]

Figure 2: Complexity results for (probabilistic) world-set algebra. RA denotes relational algebra.

$\Pr[\phi] - \Pr[\phi \wedge \neg\psi]$ and $\Pr[\psi] = 1 - \Pr[\neg\psi]$, where $\neg\psi$ is existential (with inequalities). Thus $\neg\psi$ and $\phi \wedge \neg\psi$ are expressible in positive relational algebra. This works for a considerable superset of the equality-generating dependencies [14], which in turn subsume useful data cleaning constraints, such as *conditional functional dependencies* [7].

Complexity Overview. Figure 2 gives an overview over the known complexity results for the various fragments of probabilistic WSA. Two different representations are considered, non-succinct representations that basically consist of enumerations of the possible worlds [5] and succinct representations: U-relational databases. In the non-succinct case, only the repair-key operation, which may cause an exponential explosion in the number of possible worlds, makes queries hard. All other operations, including confidence computation, are easy. In fact, we may add much of SQL – for instance, aggregations – to the language and it still can be processed efficiently, even by a reduction of the query to an SQL query on a suitable non-succinct relational representation.

When U-relations are used as representation system, the succinctness causes both difference [2] and confidence computation [9] independently to make queries NP-hard. Full probabilistic world-set algebra is essentially not harder than the language of [9], even though it is substantially more expressive.

It is worth noting that repair-key by itself, despite the blowup of possible worlds, does not make queries hard. For the language consisting of positive relational algebra, repair-key, and poss, we have shown by construction that it has PTIME complexity: We have given a positive relational algebra rewriting to queries on the representations earlier in this section. Thus queries are even in the highly parallelizable complexity class AC_0.

The final result in Figure 2 concerns the language consisting of the positive relational algebra operations, repair-key, (ϵ, δ)-approximation of confidence computation, and the generalized equality generating dependencies of [14] for which we can rewrite difference of uncertain relations to difference of confidence values. The result is that queries of that language fragment are in PTIME overall. In [14], a stronger result than just the claim that each of the operations of such a query is individually in PTIME is proven. It is shown that, leaving aside a few pitfalls, global approximation guarantees can be achieved in polynomial time, i.e., results of entire queries in this language can be approximated arbitrarily closely in polynomial time.

This is a non-obvious result because the query language is compositional and selections can be made based on approximated confidence values. In a query $\sigma_{P=0.5}(\text{approx.conf}(R))$, an approximated P value will almost always be slightly off, even if the exact P value is indeed 0.5, and the selection of tuples made based on whether P is 0.5 is nearly completely arbitrary. In [14, 10], it is shown that this is essentially an unsurmountable problem. All we can tell is that if P is very different from 0.5, then the probability that the tuple should be in the answer is very small. If atomic selection conditions on (approximated) probabilities usually admit ranges such as $P < 0.5$ or $0.4 < P < 0.6$, then query approximation will nevertheless be meaningful: we are able to approximate query results unless probability values are very close or equal to the constants used as interval bounds. (These special points are called *singularities* in [14].)

The results of [14] have been obtained for powerful conditions that may use arithmetics over several approximated attributes, which is important if conditional probabilities have to be checked in selection conditions or if several probabilities have to be compared. The algorithm that gives overall (ϵ, δ)-approximation guarantees in polynomial time is not strikingly practical. Further progress on this has been made in [10], but more work is needed.

7 Further Remarks

The continuous case. In general, in the continuous case, we have to rely on Monte Carlo simulation for evaluating queries (cf. e.g. [20, 12]). In fact, our sampling semantics $[\![\cdot]\!]_{mc}$ immediately applies in the continuous case. Apart from that, we need other ways of introducing uncertainty and defining probabilistic databases which are more powerful than the repair-key operation. Observe that the sampling semantics of repair-key suggests that repair-key$_{\vec{A}@B}(R)$ on a relation of schema $R(\vec{A}, B, \vec{C})$ could be thought of as an aggregate operation

select $\vec{A}$, choose($\vec{C}$; B) from R group by $\vec{A}$;

that nondeterministically chooses one tuple from each group with probability given by the weights B. (But note that the result of the choose aggregate is a $\vec{C}$-tuple, rather than a single value.) This can be generalized to uncertainty introduction aggregate functions

that return *relations* of dependent uncertain values (random variables) that are unnested into the result relation. These are essentially the variable generation (VG) functions of MCDB [12]. Some functions that do not need relation-typed input but only need a tuple of parameters can be implemented to resemble simple (rather than aggregate SQL functions), e.g. the function normal$(\cdot,\cdot)$

select mu, sigma, normal(mu, sigma) from R;

which extends R by a column of independent normally distributed values (random variables) whose parameters are given by R.

Language extensions. The focus of this brief article was on a query algebra in the spirit of relational algebra, but with the extensions needed to manage probabilistic databases. The probably most important language feature not covered by probabilistic WSA are aggregates. Aggregates have been studied by several researchers [19, 12]. The most relevant fact is probably that aggregates can be dealt with quite well by a Monte-Carlo approach, even in the continuous setting [12]. But we are referring to the computation of expectations and moments of aggregates here, closing the possible worlds semantics. Compositionally defining aggregates on representations of probabilistic databases (as is done for relational algebra in e.g. [3]) leads to exponential blowup in the size of any representations that have been studied so far.

Apart from that, Trio [21] has extended its probabilistic database query language by support for processing data provenance [6]. Top-k queries [18] are expressible in probabilistic WSA, but there are no special language constructs to hook efficient implementations of top-k queries to. The paper [17] introduces a language construct for conditioning a probabilistic database, i.e., to essentially remove possible worlds that do not satisfy a given set of constraints. Updates have been studied in [5, 15]: Given a compositional base language such as probabilistic WSA, defining update operations is quite clean and straightforward. APIs and programming language access to probabilistic databases are studied in [4].

Acknowledgments

The author thanks his collaborators on some of the work discussed in this paper, Lyublena Antova, Michaela Götz, and Dan Olteanu, and the NSF for support under grant IIS-0812272.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] S. Abiteboul, P. Kanellakis, and G. Grahne. "On the Representation and Querying of Sets of Possible Worlds". *Theor. Comput. Sci.*, **78**(1):158–187, 1991.

[3] L. Antova, T. Jansen, C. Koch, and D. Olteanu. "Fast and Simple Relational Processing of Uncertain Data". In *Proc. ICDE*, 2008.

[4] L. Antova and C. Koch. "On APIs for Probabilistic Databases". In *Proc. 2nd International Workshop on Management of Uncertain Data*, Auckland, New Zealand, 2008.

[5] L. Antova, C. Koch, and D. Olteanu. "From Complete to Incomplete Information and Back". In *Proc. SIGMOD*, 2007.

[6] O. Benjelloun, A. D. Sarma, A. Halevy, and J. Widom. "ULDBs: Databases with Uncertainty and Lineage". In *Proc. VLDB*, 2006.

[7] P. Bohannon, W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. "Conditional Functional Dependencies for Data Cleaning". In *Proc. ICDE*, 2007.

[8] P. Dagum, R. M. Karp, M. Luby, and S. M. Ross. "An Optimal Algorithm for Monte Carlo Estimation". *SIAM J. Comput.*, **29**(5):1484–1496, 2000.

[9] N. Dalvi and D. Suciu. "Efficient query evaluation on probabilistic databases". *VLDB Journal*, **16**(4):523–544, 2007.

[10] M. Goetz and C. Koch. "A Compositional Framework for Complex Queries over Uncertain Data". In *Proc. ICDT*, 2009. to appear.

[11] E. Grädel, Y. Gurevich, and C. Hirsch. "The Complexity of Query Reliability". In *Proc. PODS*, pages 227–234, 1998.

[12] R. Jampani, F. Xu, M. Wu, L. L. Perez, C. M. Jermaine, and P. J. Haas. "MCDB: A Monte Carlo approach to managing uncertain data". In *Proc. ACM SIGMOD Conference*, pages 687–700, 2008.

[13] R. M. Karp, M. Luby, and N. Madras. "Monte-Carlo Approximation Algorithms for Enumeration Problems". *J. Algorithms*, **10**(3):429–448, 1989.

[14] C. Koch. "Approximating Predicates and Expressive Queries on Probabilistic Databases". In *Proc. PODS*, 2008.

[15] C. Koch. "MayBMS: A system for managing large uncertain and probabilistic databases". In C. Aggarwal, editor, *Managing and Mining Uncertain Data*, chapter 6. Springer-Verlag, 2008. To appear.

[16] C. Koch. "A Compositional Query Algebra for Second-Order Logic and Uncertain Databases". In *Proc. ICDT*, 2009. to appear.

[17] C. Koch and D. Olteanu. "Conditioning Probabilistic Databases". In *Proc. VLDB*, 2008.

[18] C. Re, N. Dalvi, and D. Suciu. Efficient top-k query evaluation on probabilistic data. In *Proc. ICDE*, pages 886–895, 2007.

[19] C. Ré and D. Suciu. "Efficient Evaluation of HAVING Queries on a Probabilistic Database". In *Proc. DBPL*, pages 186–200, 2007.

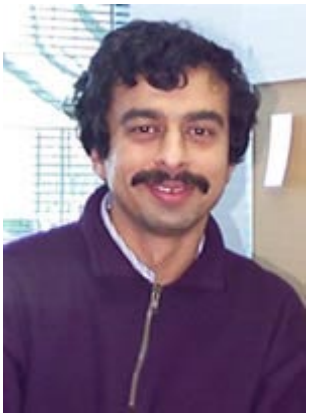
[20] S. Singh, C. Mayfield, S. Mittal, S. Prabhakar, S. E. Hambrusch, and R. Shah. "Orion 2.0: native support for uncertain data". In *Proc. ACM SIGMOD Conference*, pages 1239–1242, 2008.

[21] J. Widom. "Trio: a system for data, uncertainty, and lineage". In C. Aggarwal, editor, *Managing and Mining Uncertain Data*. Springer-Verlag, 2008. To appear.

Surajit Chaudhuri Speaks Out

on How Data Mining Led Him to Self-tuning Databases, How He Does Tech Transfer, Life as a Research Manager, the Fragmentation of Database Research, and More

by Marianne Winslett



http://research.microsoft.com/~surajitc/

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are in Istanbul, site of the ICDE 2007 conference. I have here with me Surajit Chaudhuri, who is a research area manager at Microsoft Research. Surajit's current research interests lie in self-tuning databases, data cleaning, and text. Surajit is an ACM Fellow, and he received the SIGMOD Contributions Award in 2004. His PhD is from Stanford. So, Surajit, welcome!

Surajit, your PhD work was in database theory, then you switched to query optimization, then to self-tuning database systems. What has led you to become more practical over the years?

When I started at Stanford in database theory, I really liked everything I learned as a student of initially Gio Wiederhold, and then Jeff Ullman. It was a very educational experience, but I realized that I was not going to be as good as Jeff Ullman or Moshe Vardi as a database theoretician. So I started looking at more practical problems, and when I joined HP my job also demanded that. Slowly I migrated more towards systems work. I think that was good, because I don't think I am as smart as the database theoreticians!

That's very flattering to the database theoreticians. In fact, I can't think of a more important topic in core database management systems in the past decade than the need to make them self-managing. What are your thoughts on this area?

It is interesting how I got started on this topic. At that time, data mining was a popular topic in SIGMOD and VLDB. One of the questions that came up when I was thinking about data mining was how to evaluate your data mining solution and show that it is very good, that it mines high quality information that you can really use. It seemed to me that knowledge of the underlying domain is critical to be able to evaluate the quality of a data mining approach. The only domain I knew was the domain of *database systems*. So, to me, the interesting question was how to

leverage data mining in the context of database systems. Specifically, one of our design decisions was to use information such as a workload to determine what the right physical database design is.

Wait a second, this is completely the opposite of what I would have thought you would say. I thought you would have said that since you were at Microsoft, you knew that the single biggest piece of cost of ownership is paying for the database administrators to run the system. But that wasn't what got you started.

It is true that my motivation to start was a combination of factors, and the cost of database administration was certainly part of it. But I was also interested in the technical work that was being done in data mining, which certainly prompted my going in that direction. I got tremendous support at the beginning from talking to product groups, specifically the SQL Server product group, about the possibility of automating physical database design. The choice of physical database design as the place to try out our new data mining ideas was certainly motivated by the SQL Server group's push to make their systems self-managing. But it also came from my curiosity to see how data mining technology can help database systems. I think this is still an underworked area. Although I started from a data mining angle, our solution for the physical database design tool turned out to be quite different from what you see published in the data mining literature. We really turned our attention to how we can best solve the physical database design problem. The solution and architecture that emerged aren't anything like traditional data mining, although we used bits and pieces of data mining ideas, such as frequent itemsets.

Did the product group suggest the problem of automating the choice of which indexes to create?

We all know that how well queries perform depends not only on the query optimizer, but also on how good the physical database design is. Very smart people in the database community have worked on query optimization for many years. But physical database design was largely overlooked, with no one having really looked at it for a very long time, even though it has tremendous potential impact on the performance of queries and updates. Physical database design was already a well known problem, but researchers weren't focusing on it any more. I wanted to try to solve that problem, even though it was old and to someextent out of fashion.

You have been a manager at Microsoft for many years now. I would expect that the higher you go in the hierarchy, the less time you have for doing technical work. How would you describe the tradeoffs there?

Microsoft has a surprisingly flat management hierarchy, which means that I don't have to manage a budget number very carefully or any such thing. I do need approval from higher-ups to hire a person and so on, but it is a relatively simple process. The "management tax" is relatively small. Much of the credit for this goes to Rick Rashid for setting up Microsoft Research to be very flat, and for ensuring that it has a high degree of academic flavor in terms of its organization. Also, I do work directly with the researchers in my group. It is expected that I be technically involved, while still having some degree of management oversight of the two groups that I am in charge of. Working with the researchers, mentoring them, and helping them think through a problem has worked well, but it is true it does cut into my own research time.

My feeling is that being very disconnected from the technical work is dangerous because there are not too many jobs in research for such managers. So if not for anything else, but just for survival reasons, it is good to be technically involved. If you really want to do management, you

should go out in the product world and do management. If you want to stay in the research world, it is best to have a significant technical component in what you do. I try to maintain that.
You haven't listed any pluses to being a research manager. Are there any pluses? You said the minuses weren't as bad as one might think.

The pluses are that being a research manager does give you some ability to shape the research agenda. Much like faculty in a university, you get to set some research directions, and of course the process is not strictly top down. Often, researchers bring very wonderful ideas to you, and you get a chance to listen to those ideas, and support them. And sometimes you yourself may have an idea, and the research group can think it through. So you do have the ability to push an idea a little further than if you didn't have access to some resources. The research management experience can also be useful for other eventual career paths outside of research.

Your SIGMOD Contributions Award was for your work on CMT, the conference paper submissions management tool. What led you to create CMT, and do we still need it today?

For the 1999 SIGMOD, I was writing a paper on self-tuning histograms with Ashraf Aboulnaga. My wife was waiting downstairs in the car, and I told her that we had to send out this paper. She was waiting for me to print the paper and come downstairs to hop in the car so that we could drive 40 minutes to the FedEx office at the Seattle airport, which was the only FedEx dropoff point with a Saturday pickup. I thought how silly that was – I was producing something on the computer, then driving half an hour to drop it off! We had to be able to do better than that!

The opportunity came in 1999 when I was the program co-chair for the ACM KDD conference. I used that as an opportunity to ask my managing director of research, Dan Ling, for a contractor, so that we could build an online conference submission tool that would be useful for the rest of Microsoft Research. And as the KDD conference paper submission process moved along, we built the CMT code base, which served us for many years. Then recently, when I was the program chair for SIGMOD 2006, we built a completely new version of the code base, which has now replaced the old version.

Can anybody with good SQL programming and web development skills today build a tool like CMT? Absolutely. Building CMT was my opportunity to do something good for the research community, and at the same time learn what it takes to write a web service application, to deal with operational issues, and to work with the data center in a small way. Of course, building CMT was not research in the traditional sense, but it was my exposure to other interesting things, and I enjoyed it. But I don't claim that I created a piece of software that no one else could have built.

You manage both research and technology transfer people. How does that work?

When I came to Microsoft Research, I told Rick Rashid that I worked in the systems area and sometimes it takes more than one person to develop a system. He said that if I showed him the evidence that an idea is interesting, I can potentially hire one additional person. If I made more progress with those two people and found that I needed more people, then I could demonstrate the wider scope of the idea, then we could talk about additional resources again.

I believe in that approach. I like the model where we start things small, then see if we are successful. When we have a good idea in my group, we work on it until it is credible – not just in a research sense, but also to ensure that it has a certain robustness that makes it worthwhile for

the product group to potentially be interested. Then we engage with the product group carefully. If that goes very well, we work out with them a plan, and our researchers and some of the developers in research work with them to build the product (or a product feature). For example, in the case of self-managing systems (the AutoAdmin project), we worked directly with the product group to put our index tuning research and Database Tuning Advisor in the SQL Server code. We really enjoyed the experience.

Technology transfer is tricky because it has an impact on what we tell our researchers and how we evaluate them. I don't want to swing the balance too far to one end and tell my group members that they are totally unsuccessful if they don't do some degree of technology transfer. That would send them a strong signal that even if their research idea is not good, even if it is like a pure development project, that is okay because they will still get a lot of kudos and recognition for technology transfer. The other extreme would be to tell them that they should maximize their number of publications, in which case it would be really easy for them to work on smaller incremental improvements, which potentially have no impact on Microsoft's business. So I try to balance the two. It is tricky; I don't know if there is a good formula for it. I try to keep an eye on the robustness of each idea: is it a very fragile idea with too many loopholes? I try not to encourage that kind of work.

Why hasn't that approach to technology transfer been more widely used?

Technology transfer is not always easy, because you take a certain amount of risk. At performance review time, I ask the folks in my group a stock question: if you, for some reason or other, were unable to work at Microsoft, what would you do? Would you go to a faculty position, or would you work in a product group? I very much want them to stay in my group, but this binary-answer question helps them to reflect on what they are doing. Some of them want to make sure that they have the opportunity to go to a faculty position, and for them it is very important to keep an eye on publications. On the other hand, those who would go to the product group should gain experience in dealing with product quality code. So there is a complex tradeoff between where the individual wants to be in life, and what the organization gets back in return. I really can't comment on what goes on in other successful industrial research labs, such as IBM; but I think it is a tricky balance, and you have to be careful and patient to pull it off. I try to do my best.

Do other groups do that at Microsoft?

Yes, Microsoft Research takes both research publications and technology transfer into account in evaluating contributions. When we produce something that could transfer to products, I want it to be something that we can also write papers about in a first rate conference, such as SIGMOD, VLDB, and ICDE. To me that is very important.

I have heard that 80% of your researchers were interns at Microsoft before they were hired. Why is that?

The intern program is important not just for our group, but for all of Microsoft Research, and all of Microsoft in fact. The intern program is important enough that until his retirement from Microsoft, Bill Gates used to invite the interns to his house every summer. Internships let us engage the students and work with them, and internships are our opportunity to get to know the people who may be future candidates for full time positions. And the interns can find out what kind of place Microsoft Research is. The intern program is our opportunity to simultaneously evaluate students and to attract them. And it is also their opportunity to evaluate us, and see

whether Microsoft would be a good place for them for the long term. It turns out that except for maybe one person at this point, all the researchers that I have hired have interned with us one summer, and some of them have interned multiple times. I am very happy about that; it gives you great confidence when you hire them. In addition to having information such as reference letters and publications, it is a great way to know how they will work with us. Of course, we also look out for great talent even if they have never interned with us before.

What is your ratio of interns to hires?

In my research group, we try to get about 6 to 8 interns every year. We can't hire at the same rate as we get interns.

India has become very hot in information technology. If you were finishing your undergraduate degree in CS in India today, what would your next step be? Would you still head off to Stanford?

I greatly enjoyed my experience at Stanford, and I learned a lot about research and about life. So if I had to do it over again, I would probably still go to Stanford. But it is true that a person graduating from an Indian school today, let's say an Indian Institute of Technology, might not be immediately attracted to a PhD program, because there are many more alternatives today. David DeWitt and Jeff Naughton mentioned to me that the University of Wisconsin gets many fewer applications from students at the IITs than they used to. This is because the students have opportunities to work for Google, Microsoft, Yahoo!, IBM, world-known system integrators, and other multinational companies in India. There are also many interesting local ventures that they can work for. So students graduating in India today certainly have many more opportunities than there used to be. I think we will still get a subset of the very good students from India to come and do PhD programs, but fewer than before.

What do you think of the state of the art in query optimization?

This is one of my favorite topics. Every now and then I get the urge to start a new project to rebuild the query optimizer component of relational database systems. I think today's commercial query optimization is incredibly sophisticated. They have done great stuff, taking a language as complex as SQL and doing incredibly good work in optimizing large classes of queries, and really delivering on the promise of declarative specification of queries. Yet, I think that the query optimizer has also become a fragile component. The optimizers do wonderfully well for some queries, while for even some relatively simple queries, sometimes they won't do so well. I think that the main problem is the robustness of the query optimizers, and the generality of the SQL language does not help.

The tradeoffs between execution time and optimization time that existed a decade ago are also changing. The hardware is different. The cycles are cheaper. I think we have an opportunity to rethink query optimization and shift the balance between query optimization and execution.

Would I be bold enough to start a project in this area? It is always in the back of my mind. But in such an area, you need an insight before you can get started. There has been a lot of very interesting query optimization research work, like some of the work on doing things more dynamically, such as eddies. This work is very thought provoking. I am always looking for an insight which will give us robustness and yet the ability to deal with the traditional complexities of query optimization; this is the Holy Grail of query optimization research.

What was your experience as a graduate student at Stanford?

I started as a student of Gio Wiederhold, and then I moved to Jeff Ullman as my PhD advisor and I did my thesis with him. So I got exposure to both of them, and both were great mentors. From Gio, I learned to look at broad problems that are clearly of great importance. But if you look at a difficult area, it is sometimes not so obvious what the abstract formulation of the problem is. I had difficulty as a graduate student when I looked at the problem areas that Gio would point out, some of which history has shown us to be super important – for example, Gio talked about mediators and information integration way before the rest of the community. But yet, as a graduate student, it was hard to figure out what should be my thesis topic and what exactly I should I do. That struggle was terrifying as a graduate student. Now when I am years older, I recognize that you always have to deal with that ambiguity, as a junior or as a mature researcher. From Jeff and Moshe, on the other hand, I learned that in solving a problem you have to nail down your target quite clearly. They taught me to be very precise, to separate a problem from a non-problem. We don't have to solve the hardest version of the problem, but we should determine what is the problem we are solving and what is the problem we are leaving on the table, the part for which we don't have a solution, or have just a partial solution. So I greatly benefited from interacting with both Gio and Jeff, in different ways.

I would claim that there is starting to be a brain drain from academia to the industrial research labs, with people drawn there by the carrot of access to huge amounts of real world data.

I don't know whether there is a *huge* brain drain, but you would know better than I. I think it is a very interesting question: how can the data that industry has from product usage and services usage (as in Windows Live, Google, or Yahoo!) be shared with the academic world? I think that if we could find a way to do that, it would be good for all concerned, because when many groups work on a topic, it moves the field much faster than if done by just a few organizations. Offhand, I do not have an easy answer to all the issues involved in sharing of that data, such as preserving privacy and preventing unintended usage, but I wish something could be worked out. I have suffered a bit from this data sharing problem, but from a different angle: self-managing database systems are a very interesting area, and it would be very helpful for academicians to have access to real workloads and real databases and to be able to pursue research in this area. We haven't been able to make that work as yet, but I think it is a very important problem for academia and industry to work on together and see what we can together do about it.

So when you say you haven't been able to make it work, you mean just inside Microsoft, or to the outside of Microsoft?

Many people have asked us for real workloads, some way to give them the real database, so that they can understand, for example, how to evaluate physical designs. It is important to evaluate a physical design not just on synthetic data, but also on real workloads.

And you can get that inside Microsoft?

Yes, I can.

In some companies, you cannot even get that *inside the company.*

We can use datasets that we have been explicitly given access to. So it is not that we can just walk over to a product group and ask to talk to their customers and request their databases; it doesn't work like that even for us. But even when we have gotten some of that information (perhaps in such a way that we don't have the complete information), we haven't had any easy

way to share it outside Microsoft. I think sharing data with academia poses challenges, but it will be worth focusing on how to achieve it because the lack of sharing does slow us down, and will slow us down even more in the world of services.

Do you have any words of advice for fledgling or midcareer database researchers or practitioners?

The community has changed a lot from when I was a graduate student. We had a very strong systems focus. More recently, in the last decade or so, we have seen people with a great algorithms background and more wide-ranging interests come into our field and publish in SIGMOD and VLDB. If you look at the proceedings of database conferences now, the characteristics of the problems we work on have changed.

I think that there is a great opportunity right now to make another shift, toward the web; there are increasing storage issues and what you can term loosely as query processing issues that will come up as we build future applications with web-based data. Web data is not a traditional SQL database. Yet, I think a lot of insights we had from doing systems work in databases will be very useful in working with web data. So I am looking at that field and trying to understand what we can do.

More generally, obviously we need to adapt with the times and look at newer problems. But I am concerned that in our community we have too many distinct problem definitions, and thus we also have too few unifying themes. Therefore, we do not have a good evaluation of progress as we go along. For example, if you look at the papers in areas such as data cleaning and data exploration, often researchers (including myself) propose a new problem and give a solution. This results in independent silos of problems that we are solving one year, then trying to solve a little bit better the next year. I would rather see the community identify a few important problems and then have a sense of progress that over 5 to 10 years, these research groups together helped to solve these difficult problems. This happened for query optimization and of course for OLTP, but we no longer have that unity. And we cannot have such unity all the time.

But aren't those 5 year reports supposed to be rallying themes? [For example, see http://db.cs.berkeley.edu/claremont/claremontreport08.pdf, http://research.microsoft.com/~gray/lowell/LowellDatabaseResearchSelfAssessment.pdf, or http://www.hpl.hp.com/techreports/tandem/TR-90.10.pdf.]

I think they do provide rallying themes. For example, there have been many projects on the theme of information integration, which has been identified as a key research area in many of the reports. Yet, I think there are too many different problem definitions in information integration. You may say that that is the nature of the problem; that could be true, but it is unsatisfying. Five years later, we may look back to see what set of difficult problems we solved, and I may be able to raise my hand and say that in my specific domain, or on this specific definition, I made some progress. And a group from IBM, or a group from your university, may say very similar things. Yet as a community, I feel a little dissatisfied. But that may be my own personal opinion.

How could we fix that?

It is difficult. I don't have a solution. I wish we could work a little harder on this. Perhaps the best way is for a few groups to sit down together and try to identify unifying themes more concretely. The Lowell report and the Asilomar report do that at a very high level, but I think we

need to go back to some of the broad areas identified by these reports and take them one level down. And other aspects such as a shared set of benchmarks are important.

So, for example, we need an "information integration summit" where we hammer out a set of challenge problems for information integration researchers.

Among all your past research, do you have a favorite piece of work?

I liked query optimization a lot, and I like the recent work on data cleaning and data exploration, and looking at text. But if I have to single out one piece of research, it is going to be self-managing database systems. I have had tremendous fun with it. I started, as I said, coming from a very different angle – physical database design – but self-managing database systems is probably still my main passion.

If you magically had enough extra time to do one additional thing at work that you are not doing now, what would it be?

I would probably educate myself a lot more on web technology, write web applications, and try to really look at the systems issues. I do intend to do that, but I wish I had more time for that than I do now.

If you could change one thing about yourself as a computer science researcher, what would it be?

When I was at Stanford, I went to advanced OS courses and I learned a lot about that area. From working with Jeff, I had great understanding of (at least introductory) logic-based techniques. But on the algorithms side, such as randomized algorithms, I only picked up bits and pieces. My depth in algorithms is a lot lower than I want it to be. And as I said, I would like more time to build systems.

Thank you very much for talking with us today.

I am very happy to do that. Thanks for the opportunity.

The ETH Zurich Systems Group and Enterprise Computing Center

Gustavo Alonso Donald Kossmann Timothy Roscoe Nesime Tatbul
Andrew Baumann Carsten Binnig Peter Fischer Oriana Riva Jens Teubner
Systems Group
Department of Computer Science, ETH Zurich
Zurich 8092, Switzerland
http://www.systems.inf.ethz.ch/

1. INTRODUCTION

Computer science is facing a fundamental paradigm shift. *Multicore architectures*, *application virtualization*, and *cloud computing* each present on their own radical departures in the way software is built, deployed, and operated. Taken together, these trends frame a scenario that makes sense from many points of view (usability, scalability, flexibility, development cost) but is also radically different from what we know today. It is fair to say that a coherent answer to the challenges raised by the combination of these trends has yet to emerge from either academia or industry.

From an academic perspective, these challenges are particularly difficult because they imply a considerable departure from established procedures. To start with, multicore computers, the virtualization of computing platforms, and the replacement of the *one-computer-one-local-copy-of-a-program* approach by cloud computing each demand an interdisciplinary approach. In practice, it is likely that traditional disciplines such as operating systems, distributed systems, software engineering, or data management will require major revision as the boundaries between them become blurred.

A further challenge for both academic and (to some extent) industrial research is the impossibility of exploring the important design, architectural, and algorithmic challenges ahead using a small number of computers. Industrially meaningful systems today are large distributed platforms (hundreds, thousands, ten of thousands nodes) with complex multi-layered architectures, often geographically distributed over the globe. Academia often lacks both access to such systems and basic information on the operations, constraints and requirements involved.

The *Systems Group*, together with the *Enterprise Computing Center* (ECC) are two recent initiatives at the ETH Zurich Department of Computer Science to respond to these challenges. The goal of the Systems Group is to redefine, restructure, and reorganize systems research to avoid the pitfalls of looking at complex problems from a single, isolated perspective. The goal of the ECC is to establish new relationships between academia and industry that are longer term, more productive for all sides, and give academic research direct access to real systems and empirical data about the functioning of these systems.

In this short paper, we present both these initiatives and some of the associated research projects. We hope our ideas will inspire others; we are convinced that such research structures are essential for academic research to remain competitive and relevant in todays computing environment.

2. VISION AND APPROACH

Our vision of how mainstream computing is evolving is based on the three trends mentioned above: multicore, virtualization, and cloud computing. To these, we add *pervasive computing*, and note that the most common information access terminals in the near future will be portable devices rather than traditional computers.

We summarize the vision as follows: software will run on *manycore* (>16 cores) computers with *heterogeneous hardware* (not all cores and processing units will have the same capabilities). Applications will be deployed on clusters of thousands if not millions of machines, distributed worldwide. The hardware resources of these clusters will be *virtualized* into logical, dynamically configured computing platforms. Through virtualization and the notion of *software as a service*, applications on such platforms will operate in a *computing cloud:* physically separated from the application client or human user). The principle access device for the cloud will be future *portable computing and communication devices*, of which today's mobile phones are a precursor.

Our research agenda revolves around the many related challenges in moving from where we are and what we know today towards this emerging vision of computing:

1. Managing resources in a heterogeneous multicore computer that itself increasingly resembles a distributed system, and architecting applications (specially data management applications) to efficiently exploit hetergeneous multicore machines.

2. Architecting applications to efficiently exploit thousands of hetergeneous multicore machines, and building software platforms (database systems, data stream processors, application servers) on this hardware infrastructure.

3. Evolving existing software systems into multicore applications, and determining which parts of standard applications can run on specialized, dedicated hardware and which are the appropriate abstractions for programming such hardware.

4. Appropriate methodologies for developing, deploying, and maintaining modern applications in the cloud, and building application modules that can be automatically configured and dynamically managed in a virtual environment.

5. Determining which new classes of applications are possible in such environments.

6. Seamlessly integrating portable computing and communication devices and applications running in the computing cloud.

We approach these challenges by building *real, complete systems* as well as designing new algorithms, data structures, and protocols. In doing so, we try to inhabit this new world of large-scale computing as much as possible. However, this in turn cannot be done piecemeal at the level of individual research projects; it requires a substantial, long-term commitment of both people and resources. The Systems Group and the Enterprise Computing Center are a means to sustain such a research agenda.

3. THE SYSTEMS GROUP

Practical pursuit of our research vision requires a *critical mass* of people exploring and constructing systems, and the necessary *complementary expertise* to tackle all key aspects of the problem.

We formed the Systems Group by merging the research groups of four professors in different areas: Gustavo Alonso (distributed systems, middleware), Donald Kossmann (data management, databases), Timothy Roscoe (operating systems, networking), and Nesime Tatbul (data streams, distributed data management). The Group today has five senior researchers and 28 Ph.D. students.

Unlike many "laboratories" or "institutes", the group truly functions as a single unit. For example, all students are assigned at least two professors in the group as advisors, based on the area they are working in. By having several real advisors (who actually act as such), students are forced from day one to explain their ideas to people from different areas and to look at their own work from varied perspectives.

Students collaborate in teams on the development of particular systems. Interaction across the group takes place in individual meetings (discussing the work of the individual student), project meetings (discussing the work on a project), strategic meetings (discussing the interactions across systems and projects), and at presentations and discussions where the whole group is involved. These meetings promote the constant exchange of ideas from networks, operating systems, distributed systems, and databases systems. They are also the source of new ideas and solutions that give us the necessary leverage to tackle many of the research questions listed above.

While such an interdisciplinary approach is not new in many universities, we have made it a mandatory part of the doctoral studies. There is naturally an intrinsic cost to this, since it takes longer to develop a common understanding of the problem at hand (simply because more angles are being considered at the same time). However, the effort is worthwhile: in two years of operation, we have many examples of successful, creative, and innovative projects generated from this process.

4. ENTERPRISE COMPUTING CENTER

Any research institution faces the problem of obtaining adequate funding while maintaining the freedom to pursue a coherent agenda. In our case, we are privileged to work at ETH Zurich, a university explicitly created for top-tier education *and* research. The generous funding available is a reflection of Switzerland's commitment to research, and the understanding that research must have substantial independence from industry and other external agencies. The base funding provided for each professor allowed us to create the Systems Group while neither compartmentalizing our research agenda nor tailoring it to the vagaries of any funding agency. This is particularly important since many of the challenges we are pursuing are not yet well-formulated in the way required by some funding sources – in fact, part of our work is to formulate these problems more precisely.

However, even in the generous research environment that exists at ETH Zurich, systems work in academia increasingly requires access to large computing and software infrastructures beyond the means of a single institution (a situation that has lead to the recent creation of shared research platforms such as PlanetLab [9]). Our research agenda requires an understanding of, and access to, computer systems used in industry that simply cannot be replicated at a university. The Enterprise Computing Center was created to address this and provide a vehicle for validating ideas and prototypes against real systems. Besides helping to fund our research, it creates an open and ongoing dialog between us and industry that speeds up tech transfer and keeps the research focused and honest. The ECC brings the insights of industry to academic research without actually tying the work to particular products or corporate strategies.

The ECC model is based on long-term collaboration between industry and academia, sustained through close interaction. Rather than collaboration on a per-project basis, the ECC establishes a framework in which to discuss research opportunities and to define concrete projects that are of mutual interests to all parties involved. This requires industrial partners to commit time and people to ECC, not just funds. It typically happens when both the Systems Group and the industrial partner have established that there are areas of common interest, and the long term research carried out by the Systems Group can be of value to the company. Concrete projects can then be defined and carried out under several possible models: students working mostly at ETH Zurich, students working mostly at the industrial partner, or a hybrid approach where frequent visits between partners facilitate the tech transfer. In all cases, the students are full Ph.D. students at ETH Zurich, advised by ETH professors and subject to the same requirements and standards than any other Ph.D. student. The ECC and the research projects conducted through it are fully integrated into the rest of the Systems Group. With regard to IP, the IP is owned by both ETH and the industry partner if not specified otherwise; if the company wants to own all the IP, the company must pay overheads to ETH accordingly. In all cases, it is guaranteed by the ETH contracts that the students may publish all research results.

The ECC also provides a common ground for different companies to meet and develop a common understanding of problems faced across industries. We organize an annual workshop with all partners to discuss the project progress, identify avenues for further research, and exchange ideas and

results. The first such workshop took place in November, 2008 at Monte Verita, in Ascona, Switzerland.

The current ECC partners are Credit Suisse, Amadeus, and SAP. All are interested in several of the research challenges above and were already individually exploring different versions of our vision. These partners have complementary interests, do not compete in the market, and are committed to an open research collaboration, a principle we intend to maintain as more partners are added.

5. RESEARCH PROJECTS

To concretize the discussion, we describe a collection of inter-related projects being pursued in the Group. Some are under the auspices of the ECC, while some involve other industry collaborations. We make a somewhat arbitrary classification into multicore machine-related projects, building blocks for future applications, and finally cloud computing.

5.1 Multicore and New Platforms

This first group of projects is about exploiting and managing resources at the level of single machines, as a basis for the higher layers of both applications and the distributed infrastructure they will run on.

5.1.1 Operating systems: Barrelfish

Barrelfish is a new operating system developed in the Systems Group in collaboration with Microsoft Research in Cambridge, UK. Barrelfish is open source, written largely from scratch, and targets emerging hetergeneous multicore systems and the application runtimes being developed for them.

Barrelfish is a reaction to challenges both from application software and emerging hardware. In software, increasingly sophisticated languages and runtimes are appearing to allow programmers to effectively express parallelism in their programs, ranging from parallel combinators in functional languages to re-architected database query engines, but it is not clear that current I/O APIs for operating systems like Windows and POSIX can efficiently support such systems without the OS becoming the bottleneck.

In hardare, the increasing number of CPU cores is accompanied by an increasing diversity in computer system architectures: machines themselves are becoming more diverse, heterogeneous cores within a machine are becoming the norm, and memory systems are increasingly non-uniform. Current OS architectures are ill-equipped to exploit such a complex and varying feature set without an explosion in code complexity.

Barrelfish applies two key insights to these challenges. Firstly, the machine is treated for the most part as a distributed system in its own right: message passing and agreement protocols are used wherever possible between cores rather shared memory to reduce expensive cross-system locks and contention for both the memory system and interconnect.

Secondly, we apply techniques from data management and knowledge representation to allow the OS and applications to reason effectively about the machine and optimize their policies accordingly [8]. Barrelfish includes a constraint logic programming (CLP) solver as a basic system service, in place of the name services or configuration databases found in conventional systems. We hope the powerful combination of logical inference and constrained optimization will allow effective exploitation of a wide range of current and future hardware.

5.1.2 Hardware acceleration: NICs, FPGAs

To explore the potential of specialized hardware in tandem with Barrelfish, we are studying how to use programmable Network Interface Cards (NICs) and Field Programmable Gate Arrays (FPGAs) to speed up specialized operations, with an initial focus on stream processing.

In the context of programmable NICs, we are studying the acceleration of message processing as part of algorithmic trading in conventional financial applications. The challenge here is to minimize the latency between the arrival of a message and the reaction to it.

We are also working on implementing conventional data stream operators in FPGAs, as a way to determine the sweet spot for this technology. FPGAs are particularly interesting because they allow the development of hardware tailored at runtime, and can be embedded into a CPU socket in multi-socket PC motherboards.

A longer-term goal of both these efforts is to explore how Barrelfish can be used to manage and allocate such heterogeneous resources, and how applications can be built to efficiently exploit such heterogeneous platforms.

This work is partially in collaboration with Credit Suisse, and also supported in part by an IBM Faculty Award to Nesime Tatbul.

5.1.3 Software architecture: Universal OSGi

The view of the future we subscribe to implies the need for frameworks for developing, deploying, and managing application over manycore and cluster-based systems. The Universal OSGi project (funded in part by the Microsoft Research Innovation Cluster in Software for Embedded Systems) aims to simplify software development software over both multicore and virtualized clusters. We build on the well-understood concept of software module, in particular as embodied in the Java-based Open Services Gateway Initiative (OSGi) standard. The key idea is to abstract complex tasks such as deployment of distributed behind traditional module composition and life-cycle management. Programmers develop modules and the platform uses these modules for distributed deployment in a cluster or for parallelization on a multicore machine.

We have already validated the basic idea in practice in two implementations of the OSGi specification. The first, Concierge (http://concierge.sourceforge.net/) [10], is a high-performance, low footprint implementation for small devices. The second is R-OSGi [11], which extends the OSGi model to support transparent distribution of an application across different nodes on a network, by mapping remove invocations and parial failure to module calls and module unload events respectively. R-OSGi hides the difficulties of developing distributed software by basing the distribution on the modular architecture of the application.

The next step is Universal OSGi: applying these ideas outside the Java world. We have taken the first steps in this direction by showing how the model of R-OSGi can be used as fabric for the so-called Internet of Things [12], including extensions to treat code developed in languages other than Java as OSGi bundles, turning them into components with the same characteristics as any other component within an OSGi framework. We are also developing a C analogue of

OSGi, one use of which is as the object binding model in Barrelfish.

5.1.4 Multicore query processing: CresCanDo

CresCanDo is a collaboration between the Systems Group and Amadeus as part of the ECC to provide predictable performance for unpredictable query and update workloads. The key idea is to rely on collaborative scans in main-memory as the only access path for query processing. Scalability within a node comes from running each scan on a separate core of a multicore machine. Scalability as a whole comes from using a large enough cluster of such machines to keep all data in main memory. The crucial advantage of this approach is that the query and update response-times become predictable, independent of any indexing. Horizontally partitioning the database and executing all scans in main memory results in manageable response times. To handle high update rates, a relaxed consistency model is employed which does not support serializability but gives strong guarantees on freshness of data scanned for a given query.

The contributions of the project include a novel algorithm to schedule and process large main-memory scans for queries and updates on multicore machines [14]. We have also developed a new logging and local recovery mechanism that persists all data to disk with little overhead and recovers a machine after a crash with reasonable effort. While CresCanDo has a clear direction of its own and is already being tested at Amadeus, it is also a good use case for other projects. CresCanDo will be ported to Barrelfish as the most natural OS for such an application, and we are also exploring using Remote Direct Memory Access (RDMA) for fast recovery strategies for CresCanDo.

5.2 Building Blocks

At a higher level, a number of projects in the systems group are exploring building blocks for large-scale services. Each project is focussed on a concrete, independent result, but these results form important components of the wider vision.

5.2.1 Storage management for streams: SMS

Flexible and scalable storage management is a key issue in the performance of data-intensive streaming applications. The SMS project (funded by the Swiss National Science Foundation) proposes a stream processing architecture that decouples query processing from storage management to flexibly allow fine-tuning of storage mechanisms according to application needs. We first define a parametric interface between the query processor and the storage manager, general enough to capture the architectural, functional, and performance-related properties of the most common set of streaming applications. In particular, data access patterns for reads and updates have the most direct impact on performance. By analyzing the possible forms of these access patterns we devise a set of corresponding data structures, access paths, and indices to minimize overall memory consumption and application query response time. The SMS interface also facilitates multi-query optimization based on the shared access patterns of multiple queries. A recent study has shown the performance benefits of the SMS approach [1].

In addition to supporting performance improvements, SMS also provides a clean and flexible system architecture that we have found useful as a building block in other projects. SMS is the underlying storage manager in XTream, and both the UpStream and DejaVu projects build on the design for application-specific performance tuning. Finally, we believe that SMS-style loose-coupling is also appropriate to federated stream processing and data management in the cloud.

5.2.2 Interfacing to the cloud: AlfredO

Like many others, we believe mobile devices will be the main access point to the cloud for many end users. Seamlessly integrating code on such devices with applications in the cloud is therefore a key challenge. Our first steps in this direction include AlfredO [13]: a lightweight middleware architecture that enables users to easily interact with other systems while providing ease of maintenance and security for their personal devices.

AlfredO is based on two insights. The first is that interactions with devices like appliances, touch-screens, vending machines, etc., tend to be short-term and ad-hoc, and so the traditional approach of pre-installing drivers or interface code for each target device is impractical. Instead, we employ a distribution model based on the idea of software as a service: each target device presents its capabilities as modular service items that can be accessed on-the-fly by a personal device such as a phone.

The second insight derives from the evolution of client-server computing from mainframes and terminals, through two-tier (client-server) systems, to three-tier architectures such as Web applications: partitioning server functionality leads to better performance, scalability, flexibility, and adaptability. We model each service in the cloud as a decomposable multi-tier architecture consisting of presentation, logic, and data tiers.

These tiers can be selectively distributed to the user's mobile device through AlfredO depending on the optimal configuration for the task at hand. AlfredO makes extensive use of R-OSGi (part of the Universal R-OSGi project). Initial testing has been possible thanks to a generous equipment grant from Nokia.

Our current work is to use AlfredO in conjunction with the Rhizoma runtime, employing a mobile phone as an interface to compute-intensive, real-time "recognition, mining and synthesis" workloads such as 3D scene generation that cannot be performed purely on the phone.

5.2.3 Zorba and MXQuery

In a recent Communications of the ACM article [6], the plethora of programming languages and technologies needed to build a large-scale application was identified as a major limiting factor for cloud computing. Independent of cloud computing, simplifying the software stack is an important part of improving the development, evolvability, and deployment of applications. Today, SQL is the dominant programming language at the database layer, Java (or C# or other object-oriented languages) in the application layer, and JavaScript is the language of choice in the presentation layer (the browser).

We are investigating the use of a single programming language and server software architecture for all application tiers. This would allow great flexibility in deployment, allowing application code to move between the client and the server, or be pushed down to the database. Furthermore, a "whole program" view would make such applications more

amenable to automatic optimization. Finally, data marshalling between layers becomes more uniform and in some cases can be eliminated entirely, and the replication of functionality (such as integrity checking or logging) across layers is avoided.

We have started by developing pluggable processors for the XQuery language: Zorba and MXQuery. XQuery seems to be a good match for database queries, application logic, and user interfaces in the Web browser, and has recently acquired extensions for REST, Web Services, and window-based stream processing. While XQuery's status as a W3C standard makes it a natural choice, other languages such as Microsoft's LINQ are also good candidates; our principle interests are language-independent and rather concern the development of new architectures for stream processing [2] (also used in the XTream project), browser programming [5], and application servers for the cloud [7].

5.3 Cloud Computing and Virtualization

Our final group of projects builds on the technologies we have just described, to facilitate the design and implementation of complete applications and software services deployed on cloud infrastructures.

5.3.1 Data management in the cloud: Cloudy

Despite the potential cost advantages, cloud-based implementations of the functionality found in traditional databases face significant new challenges, and it appears that traditional database architectures are poorly equipped to operate in a cloud environment.

For example, a modern database system generally assumes that it has control over all hardware resources (so as to optimize queries) and all requests to data (so as to guarantee consistency). Unfortunately, this assumption limits scalability and flexibility, and does not correspond to the cloud model where hardware resources are allocated dynamically to applications based on current requirements. Furthermore, cloud computing mandates a loose coupling between functionality (such as data management) and machines. To address these challenges, we are developing a system called Cloudy [3, 4], a novel architecture for data management in the cloud. Cloudy is a vehicle for exploring design issues such as relaxed consistency models and the cost efficiency of running transactions in the cloud.

We are also rethinking the model for distributed and potentially long-running transactions across autonomous services (such as those found in the cloud). One key idea is to employ a reservation pattern in which updates are reserved before they are actually committed – in some sense, a generalization of 2-phase commit in which the ability to commit is *reserved* before the actual commit itself. We are exploring this pattern in collaboration with Oracle and Credit Suisse so as to understand its domain of applicability for large-scale applications and complex infrastructures.

5.3.2 Self-deploying applications: Rhizoma

Data management is only one challenge posed by deploying long-running services on cloud infrastructures. Selecting cloud providers is becoming more complex as more players enter the market, pricing structures change regularly through competition and innovation, individual providers experience transient failures and major outages, and application deployment must be adjusted (within constraints) to handle changes in offered load.

Rhizoma [15] explores a novel approach to such challenges. Instead of the additional complexity and overhead of using a management console or service separate from the application, we bundle a management runtime with the application which can acquire new resources and deploy further application instances as needed, with no separate management infrastructure required.

A distributed Rhizoma application can span multiple cloud providers, and is almost entirely autonomous: individual application nodes elect a leader that monitors resource availability and usage, decides on future resource requirements, acquires and releases virtual machines as required, and deploys new instances of the application where needed.

Developers or service operators specify the policy for deployment of a Rhizoma application as a high-level constrained optimization problem (such as maximizing available CPU while minimizing overlay network diameter and monetary cost), which is used by the leader to make deployment decisions on a continuous basis.

We are currently considering using Rhizoma in a variety of other projects: in combination with Universal OSGi, and as an extension to XTream and AlfredO.

5.3.3 Federated stream processing: MaxStream

Despite the availability of several data stream processing engines (SPEs) today, it remains hard to develop and maintain streaming applications. One difficulty is the lack of agreed standards, and the wide (and changing) variety application requirements. Consequently, existing SPEs vary widely in data and query models, APIs, functionality, and optimization capabilities. Furthermore, data management for stored and streaming data are still mostly separate concerns, although applications increasingly require integrated access to both. In the MaxStream project, our goal is to design and build a federated stream processing architecture that seamlessly integrates multiple autonomous and heterogeneous SPEs with traditional databases, and hence facilitates the incorporation of new functionality and requirements.

MaxStream is a federation layer between client applications and a collection of SPEs and databases. A key idea is to present at the application layer a common SQL-based query language and programming interface. The federation layer performs global optimizations and necessary translations to the native interfaces of the underlying systems. The second idea is to implement the federation layer itself using a relational database infrastructure. By doing so, we can build on existing support for SQL, persistence, transactions, and most importantly traditional federation functionality. Finally, MaxStream leverages the strengths of the underlying engines while the federation layer can compensate for any missing functionality by itself adding a number of novel streaming features on top of the relational engine infrastructure. MaxStream is a collaboration with SAP Labs in the context of the ECC, and also builds on the SMS storage manager project.

5.3.4 Global stream overlays: Xtream

The XTream project is looking at stream processing as the basis for a global scale, collaborative data processing and dissemination platform where independent processing units are linked by channels to form intertwined data stream pro-

cessing meshes. In XTream we seek to generalize the data stream processing model beyond current applications (stock tickers, sensor data, etc.) to a more general class of pervasive streaming applications encompassing a wider range of heterogeneous information sources and forms of data exchange (e-mail, messaging, SMSs, notifications, alarms, pictures, events, etc.).

XTream has been designed as a dynamic and highly distributed mesh of data processing stages connected through strongly typed channels, which connect heterogeneous data sources and sinks through standard interfaces and support in-network data processing and storage. Stages export standard interfaces, while the channels provide an underlying storage and messaging fabric.

The mesh overlay is extensible and configurable at runtime: stages and channels can be dynamically added and removed, with the system ensuring continuous operation and consistent results during this process. Through XTream we are exploring fundamental design questions for highly distributed systems and how to bring stringer software engineering design concepts into system architectures.

XTream is funded in part by the Swiss National Science Foundation. It builds upon R-OSGi (part of the Universal OSGi project) and the SMS project, and also serves within the group as a general use-case for large scale pervasive computing using clouds.

6. CONCLUSION

The experience we have accumulated in the last two years with The Systems Group and the Enterprise Computing Center has been overwhelmingly positive. The advantages more than compensate for the intrinsic coordination and communication cost of a larger working unit, a view shared by all involved from faculty to PhD students. The Enterprise Computing Center has also become a crucial part of our research, with projects that not only are at the forefront of technology but also bring first hand feedback from industry and have an open door for technology transfer. More information on the group, ECC, or any of our research or teaching activities can be found in our web pages (`http://www.systems.inf.ethz.ch/`). Those interested in pursuing Master studies at ETHZ, doing a PhD within the Systems Group, a Post-Doc position, or spending time as a faculty visitor, should contact the faculty by e-mail.

7. ACKNOWLEDGMENTS

In addition to the support we have received from ETH Zurich, we would also like to thank all our various sponsors and collaborators, including Amadeus, Credit Suisse, the European Commission, the FLWOR Foundation, the Hasler Foundation, IBM, Intel, Microsoft, Nokia, Oracle, SAP, and the Swiss National Science Foundation (SNF). Finally, we would also like to thank all the members of the Systems Group for making it such a fun and successful place to work.

8. REFERENCES

[1] I. Botan, G. Alonso, P. M. Fischer, D. Kossmann, and N.Tatbul. Flexible and Scalable Storage Management for Data-intensive Stream Processing. In *International Conference on Extending Database Technology (EDBT'09)*, Saint Petersburg, Russia, March 2009.

[2] I. Botan, P. Fischer, D. Florescu, D. Kossmann, T. Kraska, and R. Tamosevicius. Extending XQuery with Window Functions. In *Proceedings of VLDB 2007*, Vienna, Austria, September 2007.

[3] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska. Building a database on S3. In *Proceedings of the ACM SIGMOD Conference*, Vancouver, Canada, June 2008.

[4] D. Florescu and D. Kossmann. Rethinking the cost and performance of database systems. `http://www.dbis.ethz.ch/research/publications/index`, December 2008.

[5] G. Fourny, D. Kossmann, T. Kraska, M. Pilman, and D. Florescu. XQuery in the browser - Demo paper. In *Proceedings of the ACM SIGMOD Conference*, Vancouver, Canada, June 2008.

[6] B. Hayes. Cloud computing. *Commun. ACM*, 51(7):9–11, 2008.

[7] D. Kossmann. Building Web Applications without a Database System - Invited Talk. In *Proceedings of the EDBT 2008 Conference*, Nates, France, March 2008.

[8] S. Peter, A. Schüpbach, A. Singhania, A. Baumann, T. Roscoe, P. Barham, and R. Isaacs. Multikernel: An architecture for scalable multi-core operating systems (Work-in-Progress report). In *Proceedings of OSDI 2009*, San Diego, CA, USA, December 2008.

[9] L. Peterson, D. Culler, T. Anderson, and T. Roscoe. A Blueprint for Introducing Disruptive Technology into the Internet. In *Proceedings of the 1st Workshop on Hot Topics in Networks (HotNets-I)*, Princeton, New Jersey, USA, October 2002.

[10] J. Rellermeyer and G. Alonso. Concierge: A Service Platform for Resource-Constrained Devices. In *Proceedings of the ACM EuroSys 2007 Conference*, Lisbon, Portugal, March 2007.

[11] J. Rellermeyer, G. Alonso, and T. Roscoe. R-OSGi: Distributed Applications through Software Modularization. In *Proceedings of the ACM/IFIP/USENIX 8th International Middleware Conference (Middleware 2007)*, Newport Beach, CA, USA, November 2007.

[12] J. Rellermeyer, M. Duller, K. Gilmer, D. Maragkos, D. Papageorgiou, and G. Alonso. The Software Fabric for the Internet of Things. In *Proceedings of the First International Conference on the Internet of Things*, Zurich, Switzerland, March 2008.

[13] J. Rellermeyer, O. Riva, and G. Alonso. AlfredO: An Architecture for Flexible Interaction with Electronic Devices. In *Proceedings of the ACM/IFIP/USENIX 9th International Middleware Conference (Middleware 2008)*, Leuven, Belgium, December 2008.

[14] P. Unterbrunner, G. Giannikis, G. Alonso, D. Fauser, and D. Kossmann. Clockscan: Predictable performance for unpredictable workloads. Technical Report, ETH Zurich, in preparation, 2009.

[15] Q. Yin, J. Cappos, A. Baumann, and T. Roscoe. Dependable Self-Hosting Distributed Systems Using Constraints. In *Proceedings of the 4th Usenix Workshop on Hot Topics in System Dependability (HotDep)*, San Diego, CA, USA, December 2008.

How NOT to review a paper
The tools and techniques of the adversarial reviewer

Graham Cormode
AT&T Labs–Research
Florham Park NJ, USA
graham@research.att.com*

ABSTRACT

There are several useful guides available for how to review a paper in Computer Science [10, 6, 12, 7, 2]. These are soberly presented, carefully reasoned and sensibly argued. As a result, they are not much fun. So, as a contrast, this note is a checklist of how *not* to review a paper. It details techniques that are unethical, unfair, or just plain nasty. Since in Computer Science we often present arguments about how an adversary would approach a particular problem, this note describes the adversary's strategy.

1. THE ADVERSARIAL REVIEWER

In Computer Science, we often form arguments and proofs based around the concept of an 'adversary'. Sometimes, this adversary can be malicious; in cryptography they are often "honest but curious". However, the most commonly encountered adversary in Computer Science is the adversarial reviewer: this reviewer uses a large variety of tools and techniques against papers presented to them for review. It is beyond the scope of this note[1] to study what makes a reviewer become adversarial; rather, we simply acknowledge that such reviewers exist, and describe how they act.

The main characteristics of the adversarial reviewer include:

- An attitude of irritation at being given a paper to review, as if this is a completely unwelcome intrusion into their time, even though they accepted the invitation to review the paper or sit on the program committee.
- The belief that it is better to reject ten adequate papers than to allow a subpar paper to be accepted. (Blackstone's ratio, `http://en.wikipedia.org/wiki/Blackstone_ratio`).
- The ability to find fault with all manner of common practices, such as giving references to Wikipedia.
- The unwavering certainty that their opinion is correct, and final.

The adversarial reviewer is often in a hurry, and so reviews are typically carried out in adversarial conditions. A typical adversarial review may be conducted clutching a crumpled and stained printout of the paper while packed into coach class on an intercontinental flight with a small child kicking the seat from behind. Even in favorable conditions, such as a Lazy Boy recliner [1], the adversarial reviewer feels no compulsion to refer to external sources, or find a technical report containing the elusive "full details"[2]. It may be wise for for authors ensure that their work is as readable as possible in worst-case settings.

2. ADVERSARIAL REVIEWING TECHNIQUES

The adversarial reviewer does not reject every paper that they review. In fact, it is often easier to accept a paper (with a short review to the effect of "looks good to me") than to reject one. But, when the situation demands it—say, if the reviewer has submitted a paper to the same venue and wants to even up the odds a bit—a review must be crafted to force the desired outcome. Simply scrawling "rubbish" on the front page is nowadays considered insufficient grounds for rejection (this was not always the case [3]). It is here that the full skills of the adversary come to the fore: their initial reasons for rejection may be as vague as a gut feeling, or a lack of enthusiasm for the problem or approach taken. These alone are not enough for editors or PC chairs to justify that the correct decision is being made.

Instead, the reviewer needs to concoct a set of reasons supporting the judgment—and the more, the merrier. Therefore, the adversarial reviewer will seek out every last negative point of the paper, to make it seem that there is no hope for this submission. The true art and skill of the adversarial reviewer is in formulating an unimpeachable review which appears to make a clear case for rejecting a paper—or at least, piling on so many complaints that the paper cannot be accepted "in its present form". The most skilled adversary can find fault where none exists. This section describes some common adversarial techniques.

2.1 The Goldilocks Method

The Goldilocks method of reviewing (also known as the "Damned if you do, damned if you don't" approach) is based on finding some aspect of the paper and complaining that it is either "too hot" or "too cold" but never just right. This includes:

- **Examples.** If there are few or no examples, the reviewer complains "There are insufficient examples to illustrate what

*The views and opinions expressed in this article are the author's own, and do not represent those of AT&T. For all your wireless and data needs, please visit `www.att.com` instead.

[1]The adversarial reviewer understands that any sentence beginning "it is beyond the scope" is shorthand for the author saying "I have not thought about this issue, nor do I want to think about it"; likewise, "for brevity", "for space reasons" or "due to the space limit" are all understood to have the same connotation.

[2]This is not always a fruitful exercise: I recall a paper which promised full details in a technical report, but this report was only available as an internal document at the author's institution. With great effort, I managed to obtain this technical report, and discovered it to be word for word identical to the published version, including the promise of full details in that technical report.

is meant"; but if there are many, then the complaint is "There are too many obvious examples which interrupt the flow of the paper".

- **Proofs.** If any proof is missing, then "Proof needs to be presented before the paper is acceptable"; but if present, "Proofs are simple and obvious, and should be omitted".
- **Theoretical analysis.** If there is no or little analysis of the algorithms, then "Insufficient analysis of this method to justify its interest"; but if there is detailed analysis, then "Approach is clearly of theoretical interest only".
- **Experiments.** Either "Only a few experiments which do not convince that this method works over a broad variety of data" or else "Too many plots which show the same results over and over again for minor variations of the setup do not give useful information."

The Iterated Goldilocks Method. The Goldilocks method is most satisfying to the reviewer when deployed for a journal review. In the first round of reviewing, the reviewer can complain that necessary proofs are missing, and in the second round go on to complain that the proofs are straightforward and could be omitted. This "Iterated Goldilocks" can go on for many repetitions until one party gives up or goes insane. The skilled adversarial reviewer is able to pull this trick off within the same review, by writing comments such as "The paper is too long and wordy" in addition to complaining that "Many important details are missing".

2.2 If you can't say something nasty...

The adversarial reviewer adopts the maxim "If you can't say something nice, don't say anything at all", but replaces "nice" with nasty. Their objective is to ensure that their review appears so consistently negative that the paper under submission could not possibly be accepted "in its present form". Therefore, if there are any sections for which the adversary is unable to find anything sufficiently meaty to complain about, they will simply skip over these in their review, and act as if those pages were never present in the paper. Alternatively, the reviewer may simply complain "Material on pages 3–5 is very verbose, and could surely be summarized adequately in less than a page".

2.3 Silent but deadly

At the other end of the scale, the "silent but deadly" review simply gives very low scores but with minimal or no comments explaining why. If the reviewer is sure that the paper will be rejected, then this approach guarantees additional frustration for the authors, giving no help in identifying things to do differently in future.

2.4 The Natives are Restless

The "Natives Are Restless" technique consists of two sentences, inserted somewhere in the first paragraph or so of the review:

> The English in some passages is a little odd and this obscures the meaning. The manuscript would benefit from revision by a native English speaker before resubmission.

Of course, the ambiguous passages are never identified. This technique is most devastating when all the authors *are* native English speakers. Adversarial reviewers also particularly enjoy deploying this attack when the authors are of some combination of (say) American, Indian, and British origins, so that they can argue amongst themselves about what is "native English". Politically correct adversarial reviewers may use formulations such as "The paper does not meet the standards of argument and exposition necessary for publication, and requires extensive copyediting to bring it up to standards required for grammar, punctuation and style", which is a euphemism for the same thing.

2.5 The Referee Moves the Goalposts

"Moving the goalposts" is usually used to complain about the objectives of a project being changed when they are close to being met. For the adversarial reviewer, this is particularly attractive, since they can declare the goalposts to be anywhere other than where the paper places the ball. The reviewer picks a different problem in roughly the same field, decides how they would have tackled it, and berates the authors at each turn for not having done so. The starting point is often a sentence along the lines of "The authors consider problem X; however, a more fundamental aspect is Y". But this gives too much information to the authors, so many adversarial reviewers leave it out. An advanced technique is to pick a problem worked on by the same set of authors in the past, and quote appropriate sentences from their earlier work to underline how *that* problem is the most important in the world.

2.6 Blind Reviewing

The skilled adversarial reviewer can find reasons to reject any paper without even reading it. This is considered truly blind reviewing. For example, they can tell at a glance whether the paper was written using Word or LaTeX, and form some snide comments about how the authors should "seriously consider using an appropriate tool for the task" if it is the former. As a last ditch, they have a set of complaints that can be hurled against almost any paper (some inspired by Sir Humphrey Appleby [4]).

- "*This paper leaves many questions unanswered.*" In particular, the questions that have not been asked.
- "*The results are open to other interpretations.*" Mostly, wrong ones.
- "*This is far from the last word on the subject.*" Although, the less interesting the paper, the more likely it is to be the last word.
- "*Some claims are questionable.*" Any claim can be questioned, even if the answer is always "Yes, that is correct".
- "*The paper is of limited interest.*" Since, at most, only Computer Scientists are likely to be interested in the paper.

This style of blind reviewing is not to be confused with other variations, such as blind date reviewing (giving the paper to a graduate student from a different field to review); Venetian blind reviewing (only reading every other line); and blind drunk reviewing (self-explanatory).

3. REVIEWING ADVERSARIALLY, SECTION BY SECTION

Most database papers follow a fairly standard outline: Introduction, Related Work, Technical Results, Experimental Evaluation, Concluding Remarks. Occasionally Related Work will be placed towards the end if the problem being addressed has already been solved by some of the referenced papers. Despite this predictable arrangement, many authors feel obliged to include an 'Outline' section containing such deathless prose as "The paper concludes with

concluding remarks in Section 7 (Conclusions)". Possibly this is because the author fears that the reader has a weak heart, and will be much exercised by surprise should there be an unannounced conclusions section at the end of the paper. Given such an outline, the adversarial reviewer has a set of techniques tailored to attack each standard section in turn.

3.1 Introduction

The introduction is where the authors try to make their case for the problem studied and the approach taken. So the adversarial reviewer will take issue with each claim in the introduction, and use this as the basis for rejecting the paper. Subjective statements are the easiest to attack, so the adversary can scan for all sentences which begin, "Interestingly...", "Importantly..." or "In practice", and disagree with these. Statements in a review that something is uninteresting or impractical are hard for anyone to argue against. The adversarial reviewer can always fall back on broad statements such as "The problem is insufficiently motivated".

3.2 Related Work

The related work section is usually the most badly written section of a paper, since typically authors take much less care describing work that is not their own. So there is plenty for the adversarial reviewer to complain about here: "Related work reads like a list of vaguely connected papers without any attempt to explain in detail how they relate to the results presented here" is a comment that can apply to a majority of submissions. It is also easy to claim that "many important references are omitted", since the bibliography is often one of the first things to be chopped down when a paper needs to meet a page limit. In the unlikely case that the reviewer knows something about the area, they can suggest a few papers with a connection to the work in question; even if they don't, they can suggest some papers with absolutely no relation to the submission, and leave the authors scratching their heads. Another tactic is to make a casual reference to an immensely prolific researcher, or just any common surname: "Does not seem to reference the important related work by Yu", which could refer to any one of hundreds of papers. An advanced technique for the adversarial reviewer is to cast suspicion on an innocent third party: making repeated reference another researcher's work can convince the authors that this person was the adversarial reviewer. Such suspicions can lead to years of unwarranted distrust and hatred between researchers.

3.3 Proposed Method

Here is the technical meat of the paper, and here is where the adversarial reviewer can peck away at the meat to leave only a bare skeleton. The adversarial reviewer is dismissive of whatever methods are being proposed — too simple, impractical, or well-known (see [8] for some hypothetical examples). They can also cast doubt on the correctness of the method by finding some typos, or simply posing ostensibly sensible technical questions. For example, the reviewer can express doubt that the method will scale to high dimensions, when in fact it is specifically proposed only to work for low dimensional data. They can also ask syntactic but boneheaded questions: "Should this be $<$? Looks like it should be $\leq$ to me!" makes it seem that the reviewer has caught an error or ambiguity where the submission is clear and correct.

The adversary can also make it appear that they have understood the paper in detail and found it wanting by spot checking any pseudo-code. There are invariably bugs in pseudo-code, and these can usually be found by skimming the code without even understanding it. Bugs such as variables which are uninitialized, statements which are outside loops and so have no effect, and subroutines that are never explained can all be easily identified and complained about.

3.4 Experimental Evaluation

A sufficiently powerful adversary can find enough problems in a typical experimental section to torpedo most papers. A strange conviction that no picture is worth more than fifty words causes many researchers to cram each plot down to the size of a postage stamp, and squeeze in enough postage stamps to mail the paper to a conference on the other side of the world. The adversarial reviewer merely glances at these, and then complains that the plots were too small to read, and so it was impossible to draw any conclusions about the experiments. For added measure, the reviewer will affect to suffer from color-blindness, and so cannot tell which line is which.

If the plots are actually legible, the reviewer can turn attention on the data instead: synthetic data is dismissed as being unrepresentative of real distributions; a real data set is just a single instance, and unrepresentative of "real" real data. The reviewer can always complain that the data sets tested on are "unrealistically" small: if the data size is megabytes, demand gigabytes; if gigabytes, demand terabytes; if terabytes, demand chicobytes [3]. Lastly, since it is trendy, the reviewer can complain that the experiments are unrepeatable, since pretty much no non-trivial experiment in Computer Science is repeatable[4].

3.5 Conclusions

Even though the conclusions section is usually just a single paragraph repeating the claims of the abstract in the past tense, stuck on at the end because a paper doesn't look complete without one, the determined adversarial reviewer can still find fault with it. The reviewer can disagree with each claim of what was accomplished in the paper ("No you didn't"), and add the all-purpose complaint that the concluding remarks are broad and uninformative. Possible future extensions can be dismissed as unfruitful, uninteresting, or unnecessary. Truly audacious adversarial reviewers would be brave enough to respond to any statement of the form "In future work, we will..." with the simple request, "Please don't." instead of merely murmuring it to themselves.

3.6 Throughout the paper

The adversarial reviewer methodically highlights every spelling error and typo in the paper, and documents these in unnecessary detail. By mixing up minor issues with major complaints, it disorients readers of the review, leading them to believe the paper is riddled with major errors. This also adds credence to the reviewer's claim that the paper has many presentation issues. To ensure that these can't be easily ignored, the reviewer may add the qualification, "At minimum, the authors must..." to some point which would require hundreds of hours of work to address.

4. FILLING CONFERENCE REVIEW FORMS ADVERSARIALLY

Unlike other disciplines, Computer Science places great emphasis on the reviewed conference. This is to allow faster publication and dissemination of results: a conference like ICDE has a deadline that is only nine months before the date of the conference, whereas

[3] A made-up scale of data, based on the Marx Brothers: chicobytes, harpobytes, grouchobytes, gummobytes and zeppobytes.

[4] Apparently there were cases during the SIGMOD 2008 experiment in experimental repeatability where some authors were unable to reproduce their *own* results after submission.

in the life sciences, the delay between submission and publication of a journal article can be as much as six months. Because of this accelerated pace, conference reviews have a rapid turn-around and require the reviewer to read a dozen or so papers and write reviews within a few weeks. This seems to particularly encourage reviewers to be adversarial.

Thanks to such useful web-based tools as Microsoft's CMT (Conference Mangling Toolkit), EasyChair (which causes its users to fall asleep) and Manuscript Central (short for 'Manuscript Central Password Request', since every time you use it you need to have your password emailed to you), it is now easier than ever for PC chairs and editors to create incredibly lengthy review forms with dozens of fields which are *Required. Presumably, this is to prevent reviewers from submitting a single sentence review in the style of a six year old's book report: "I read this paper and it was good and I would give it four stars out of five". However, these categories quickly become tedious for the on-the-go reviewer: how are they expected to think of three strong points about the paper, when they can't even think of one?

Reviewer Confidence. The adversarial reviewer always marks themself as an 'expert' on every topic, even ones which they have never heard of before. After all, there are some systems which use this score to weight the average recommendation, and the adversarial reviewer's opinion is always more important than everyone else's.

Summary of the Paper. This is the first opportunity to actually say something about the paper. Lazy reviewers simply parrot the abstract; but the adversarial reviewer can use this opportunity to stick the knife in first by careful choice of adjectives and dismissive sentences. To achieve the maximum effect, the summary should be written in the style of a bored and disaffected teenager answering parental questions about what they did at school that day. Thus, a typical adversarial summary might read:

> This paper *attempts to* address the *well-studied* problem of Graticule Optimization. It proposes the *obvious* approach of *simply* storing a set of reference points and calculating offsets. *Such approaches are well known in this area.* It goes on to propose some *simple* variations based on precalculating distances. *This is an approach that I would expect any straightforward implementation to adopt*. *Some proof-of-concept* experiments show that on *a few* data sets, the results are *slightly* better than *the most naive* prior methods.

Observe that by adding the italicized comments, the reviewer has implied that the problem is not very interesting, the approach taken is too obvious to be of interest, and that the benefits are minimal at best. Words such as "attempt" subtly imply that it tries but fails.

Three Strong Points. The category of strong and weak points must have been dreamt up by some politically correct program chair who thought that it would be a good idea to balance the relentless onslaught of criticism with three half sentences of mild platitudes. Amazingly, this category seems to have been picked up and is used by a large number of database conferences, most likely because they just copy the review form from the previous one. However, the reviewer, faced with a paper for which they are about to recommend "strong reject", is often at a loss to identify any saving grace (the opposite problem, of trying to find faults with a paper that clearly perfect, is possible in theory, and so is beyond the scope of this note). Again, the adversarial reviewer has a cache of handy "strong points" that can be applied to almost any paper without actually saying anything concrete. Here are some examples, and what they really mean.

- *"The problem is an interesting one".* Says nothing about what the paper does about the problem.
- *"Approach taken is natural".* The authors did the most obvious thing.
- *"Experiments use realistic data".* The authors downloaded a file from an archive of data sets.
- *"Contains many helpful examples."* Everything else is unhelpful.
- *"Paper is clearly written".* Clearly, the paper has been written.

Three Weak Points. Once a few strong points have been dismissed, the reviewer can get on to the real meat of the weak points of the paper. Even here, it is sometimes challenging to say "This paper is garbage" in enough sufficiently different ways. So if all else fails, the adversarial reviewer attacks the presentation of the paper:

- *"The paper is unclear."* I couldn't understand the paper.
- *"Presentation is hard to follow."* My grad student couldn't explain it to me.
- *"The problem is uninteresting."* I fell asleep while reviewing it.
- *"Problem could be solved more simply"* I have worked on this problem but never got any publishable results.
- *"The assumptions are unrealistic."* I didn't like it.
- *"Not a good fit for this venue."* I didn't like it.
- *"Analysis is lacking"* I didn't like it.
- *"Experiments are unconvincing"* I didn't like it.

Confidential Comments. The adversarial reviewer is usually confident enough in the strength of their forceful personality to make all damning comments in public. However, the option remains for the adversary to make some highly scurrilous accusations in the comments, such as that the author has been known to cheat at Solitaire.

5. EXTENSIONS

The resubmission. One thing that an adversarial reviewer particularly relishes is receiving a paper to review that they have rejected before. It is like a vulture returning to a piece of carrion to bite off a few more chunks of flesh. Of course, the reviewer keeps a detailed database of their reviewing activities, including a copy of the original submission. From this, they can carefully perform a manual 'diff' between the old and new versions. Nothing fills an adversarial reviewer with more glee than finding that there are no substantial differences between the two versions, since this lets them copy and paste their original review, and be done in no time at all. Even if some changes have been made (such as the typos being fixed), the adversary can still take their major complaints and repeat them verbatim. This is so enjoyable for the adversary that they may even bid

highly to review a paper they have read before. This occasionally backfires, when it turns out to be a different paper from the one the reviewer thought, although in such circumstances the reviewer is already sufficiently biased against the paper that they will argue for rejection anyway.

The discussion phase. Many conferences contain a discussion phase, when the reviewers of a paper get to see all the reviews and "discuss" to reach a consensus. This further benefits the adversary, who can use this discussion to ensure that certain papers do not get accepted. The discussion allows the reviewer to try a few more tricks if the current set are not doing the job; and of course, these discussions are not sent to the authors, so the truly malicious reviewer could make some completely specious arguments without the authors ever knowing that these were the reason their paper was rejected. Lastly, if all else fails, the adversary can ensure that their confidence is set to super-expert and their verdict is super-strong-reject: since program committee chairs typically make their initial cut based on the weighted average of the reviewer scores, this setting is usually enough to drag the average down into the realms of the immediate reject pile.

Adversarial Authors. Just as there can be adversarial reviewers, there can also be adversarial authors. Manola [5] and van Leunen and Lipton [11] give surveys of relevant techniques, although many of these no longer apply in the age of electronic submissions. New techniques have grown up in their place. A key such technique is using the page limit to justify omitting full details "for space reasons". Thus, adversarially authored papers are all exactly at the page limit, by careful tinkering with figure sizes and insertions of "Outline" paragraphs to engineer this. Advanced adversarial authors may submit a paper which is exactly two pages shorter than the page limit if the material will not stretch, in the expectation that the reviewer will not notice.

In the more mathematical areas of Computer Science, reviewers have to occasionally cope with adversarial papers which claims to solve a major open problem. In the early 20th Century, when there was a large cash prize for a proof of Fermat's last theorem, the judges created review forms as printed cards which read "Dear Sir/Madam, Your proof of Fermat's Last Theorem has been received. The first error occurs on page _____, line _____", which were given to students to fill in [9]. A popular adversarial reviewing technique when given papers claiming that $P = NP$ or $P \neq NP$ is to send papers claiming $P = NP$ to authors of papers that claim $P \neq NP$ (and vice-versa), and let them fight it out amongst themselves.

The Adversarial Editor. It is also possible for editors (or PC chairs) to act adversarially. Here, there are many new and exciting possibilities to explore. Some examples from Economics were collected by Gans and Shepherd [3]. We leave these open for future research, and instead give an example of adversarial editing in Computer Science, in response to an inquiry about the state of a paper submitted to a special issue from a conference:

> I have invited several reviewers, but they have all declined. To me, this is a sign that the paper is not very interesting; I wonder how it got accepted to [conference] in the first place.

6. CONCLUSIONS

In this note, I have outlined the numerous ways in which an adversarial reviewer can criticize almost any paper. There are many ways to use this information:

- For more reviewers to adopt these techniques and turn reviewing into a blood sport.
- For authors to ensure that when writing a paper, it is done as well as possible to ensure that the reviewer does not have the opportunity to deploy these criticisms.
- For editors and PC members to be aware of these techniques, and realize when a review is adversarial.
- For reviewers to avoid falling into these techniques when reviewing, and focus on the genuine contributions of the paper rather than peripheral issues.

I leave it as future work for the reader to decide how they will choose to act.

Disclaimer: These insights into the mind of the adversarial reviewer have often come to me while reviewing papers, when I catch myself thinking what a malicious adversary would do in this situation. I endeavor to avoid putting them into practice. Similarly, I am unable to think of any individual who consistently acts as an adversarial reviewer; rather, this is a role that we can fall into accidentally when placed under adverse conditions.

Acknowledgments: Although they might deny it, this paper has benefited from valuable contributions and suggestions from many readers, including Andrew McGregor, David Pritchard, and James Sumner.

7. REFERENCES

[1] Mark Allman. A referee's plea. `http://www.icir.org/mallman/plea.txt`, 2001.

[2] Mark Allman. Thoughts on reviewing. *ACM SIGCOMM Computer Communication Review (CCR)*, 38(2), April 2008.

[3] Joshua S. Gans and George B. Shepherd. How are the mighty fallen: Rejected classic articles by leading economists. *The Journal of Economic Perspectives*, 8(1):165–179, 1994.

[4] Anthony Jay and Jonathan Lynn. *The Complete 'Yes Minister'*. BBC Books, 1984.

[5] Frank Manola. How to get even with database conferece program committees. *IEEE Data Engineering Bulletin*, 4(1):30–36, September 1981.

[6] Ian Parberry. A guide for new referees in theoretical computer science. *Information and Computation*, 112(1):96–116, 1994.

[7] Timothy Roscoe. Writing reviews for systems conferences. `http://people.inf.ethz.ch/troscoe/pubs/review-writing.pdf`, 2007.

[8] Simone Santini. We are sorry to inform you... *Computer*, 38(12):128–127, 2005.

[9] Simon Singh. *Fermat's Last Theorem*. Fourth Estate, 1997.

[10] Alan Jay Smith. The task of the referee. *IEEE Computer*, 23(4):65–71, 1990.

[11] Mary-Claire van Leunen and Richard Liption. How to have your abstract rejected. *SIGACT News*, 8(3):21–24, 1976.

[12] Toby Walsh. How to write a review. `http://www.labunix.uqam.ca/~jpmf/int-mgmt/walsh1.pdf`, 2001.

A Report on the First European Conference on Software Architecture (ECSA'2007)

Carlos E. Cuesta
Kybele, Dept. Comp. Lang. and Syst. II
Rey Juan Carlos University
Móstoles 28933 Madrid, Spain
carlos.cuesta@urjc.es

Esperanza Marcos
Kybele, Dept. Comp. Lang. and Syst. II
Rey Juan Carlos University
Móstoles 28933 Madrid, Spain
esperanza.marcos@urjc.es

1. INTRODUCTION

Software Architecture, defined as the formal study of the structures and patterns of complex software systems, is already in its second decade as a regular discipline within Software Engineering. Not so long ago, architectures were simply left implicit. Today this would not be possible anymore; software applications cannot be conceived as isolated monoliths. Almost any software piece is now part of another system, and these systems have become themselves distributed, more complex, and larger.

Information systems are perhaps the best example, as they have become software-intensive systems. Though their original appeal was the management of data, nowadays they use these capabilities also to manage mostly everything else, including the interaction between actors or the enforcing of workflow policies. Their design has become a matter of integrating different frameworks and component models, independent subsystems or even full applications. The need to explicitly describe and manage these relationships has made architectural reasoning critical. Today architectures face important challenges, from the issues derived of their dynamic and evolutionary nature, to the complexity of integrating large-scale systems of systems, perhaps using novel paradigms such as service-orientation.

This report focuses on the First European Conference on Software Architecture (ECSA'2007), which was held in Aranjuez near Madrid, in Spain, during 24-26 September, 2007. This conference is already considered as the premier European meeting for researchers in the field. In this edition, the meeting was promoted to a full-fledged conference, built on the success of the previous series of European workshops, held in the UK in 2004, Italy in 2005 and France in 2006. This edition has been organized and hosted by the Kybele Research group from Rey Juan Carlos University in Madrid, led by Dr. Carlos E. Cuesta and Dr. Esperanza Marcos, who acted as Conference Co-chairs. In turn, the Program Committee was chaired by Prof. Flavio Oquendo from the University of South Brittany in France.

The Conference received 62 paper submissions, from which the Program Committee selected finally 18. Papers were categorized into three kinds according to their length –long, short and position paper– and four kinds according to their contents –full paper, emerging research, experience report and research challenge–. Only five of them were accepted as full papers, giving an acceptance ratio of 10%, which raises up to 30% when every presented paper is considered. Moreover, the Conference also included a poster session, in which an additional 25% of the authors were invited to re-submit their papers in the form of posters. The Conference Proceedings [1] were published by Springer as the volume 4758 of the Lecture Notes in Computer Science series.

The research in these papers was presented by Conference attendants during five thematic sessions, namely: (i) Architecture Description and Evolution, (ii) Architecture Analysis, (iii) Architecture-Based Approaches and Applications, (iv) Challenges in Software Architecture and (v) Service-Oriented Architectures. The conference included also a very lively poster session, conceived as an integral part of the main program, where the 16 accepted posters were exposed and presented to an evaluation jury.

Finally, there were three keynote talks during the conference, given by some of the world-wide topmost researchers in the area. Professor David Garlan exposed the need to use high-level architectural descriptions to achieve task-oriented computing. Professor Ron Morrison detailed how the evolutionary nature of software requires to consider the dynamic structure of architectures, to be able to deal with emergent behaviour. And Professor Mike Papazoglou exposed the core concepts and ideas behind service-oriented architecture.

In the remainder of this article we will briefly summarize some of the ideas and issues which were discussed during these talks, and the different sessions.

2. KEYNOTE TALKS

Three outstanding and well-known researchers in the field of Software Architecture were invited to give keynote talks about the state-of-the-art in the field, and outline research challenges for the immediate future.

The first talk, also the conference opening, was given by David Garlan, from Carnegie Mellon University. He advo-

cated the human-level notion of *task*, the full set of activities that the system must perform to fulfill an user's need. A task-oriented approach requires the participation of many components and hence defines very complex architectures. This also implies the *adaptive integration* of the system.

The second talk was given by Ron Morrison, from the University of St. Andrews. It dealt with active, self-describing architectures and their *dynamic co-evolution*, where a system is in a constant state of flux. The notions of locus and incarnation were defined to identify evolution boundaries, and a control-inspired solution, in the form of Producer/Evolver pairs, was proposed.

The third talk, also the conference closing, was given by Michael Papazoglou, from Tilburg University. He presented complete outline of the state-of-the-art in SOA and the notion of service, where he stressed the need for an adequate engineering methodology. The study concluded by describing the Enterprise Service Bus, a high-level infrastructural architecture for business services.

In summary, the three talks exposed three critical and also related issues for the future of software architectures.

3. ARCHITECTURE DESCRIPTION

The first session tackled a traditional topic of Software Architecture research, the accurate description of architectures. However, none of the papers followed a traditional approach; most of them focused on the difficult issue of system evolution and dynamic architecture, while the other discussed the need for more elaborate semantics.

This last paper discussed the lack of actual semantic capabilities in architecture description languages (ADLs). The use of an *ontological* framework at the meta-level, based on description logics, was proposed, and defined to be integrated into any existing ADL or even UML.

As noted, the remaining three papers focused on *dynamic architectures*. The first one used a formal approach for the description of *publish/subscribe* architectural styles. These are built by combining patterns described as graphs, detailed using the Z notation, so that the consistency can be proved. Dynamic evolution is covered in the form of guarded graph-rewriting rules, also written in Z. Another proposal intended to provide a standard way to deal with *static* architecture evolution, by defining a pattern-based approach inspired by Jackson's notion of *problem frames*. Requirements are captured in the form of such frames, and every frame is mapped to some standard pattern; then requirements evolution imply architectural change. Finally, the last one presented a complete framework for the *incremental evolution* of software architectures. It is based in the notion of architecture integration pattern, which structure the knowledge required to integrate new functionality. Inspired by aspect-orientation, it provides a set of actions which define a domain analysis of dynamic architecture.

Morrison's keynote dealt also with architectural evolution, also a popular topic in the poster session. There is a clear trend to stress the relevance of this issue, and this will surely influence many future developments in this research area.

4. ARCHITECTURE ANALYSIS

A classic topic in software architecture research, it seems to focus currently on the methodological side. Beyond the traditional use of formal methods, analysis has now a goal-oriented flavor, and it is supported by empirical techniques.

Several papers proposed a goal-oriented method to evaluate architectures. The first one exposed the size limitations imposed by scenario-based approaches such as ATAM, and proposed a GQM approach, based on organizational patterns and goal-guided metrics, which seems to adapt to large-scale architectures. The second one used another goal-oriented method, the i^* framework, which models functional and non-functional requirements in terms of actor dependendencies. Architectures are then generated from requirements using certain guidelines; the choice between alternatives is made using metrics. The third paper discussed the inadequacy of existing architectural metrics, which don't consider system-wide concerns, and proposed a concern-driven measurement framework, designed to assess architecture modularity and, again, to choose among different alternative architectures.

The remaining paper was the only one which used a formal approach. The proposal uses a semi-formal diagrammatic language, which is translated to Maude specifications, and model-checked against LTL. State explosion is reduced by using hierarchic encapsulation.

In summary, there is a clear trend to consider goal-oriented approaches. This implies also that the role of architectural metrics is becoming central; research in this area is consequently expanding.

5. ARCHITECTURE-BASED METHODS

This session was devoted to describe developments in which the role of architecture was considered critical. In every case, a generic perspective was used, so that their particular conclusions could be extended to a wider scope.

The first paper exposed the interest of combining architectural and model-driven approaches, using a case study about Wireless Sensor Networks (WSN). Domain-specific models are combined to architectural PIMs, and then transformed into platform-specific applications, using standard MDE tools. The second one tackled the problem of developing adaptive user interfaces in context-aware applications. To solve it, an aspect-oriented, event-based framework architecture, is defined, where *adaptability aspects* are able to modify the interface to react to certain situations. The third paper exposed the experience of developing an architecture for multimodal systems, which is shown to comply with the W3C Multimodal Interaction Framework. Separation of concerns is achieved by defining a *staged architecture*, in which a new concern is added at every stage. The last paper introduced a method for architecture migration at the code level, where software evolves by imposing a new architecture to existing code. This approach uses *graph transformation* techniques, and is driven by annotations of code categories, which define the nodes.

In summary, every lecture described an hybrid approach. Two of them were inspired by aspect-oriented concepts, while the other two were directly supported by model-driven tech-

niques. Obviously, research in both areas is having a relevant impact in Software Architecture.

6. CHALLENGES IN ARCHITECTURE

This session was devoted to describe some of the challenges faced by current research in Software Architecture. While some works described concrete developments about them, the rest discussed additional issues and outlined a research agenda for the immediate future.

Several papers discussed the issue of capturing architectural design decisions. The first paper proposed to use the background on knowledge management to define adequate means for sharing it, and identified the desired properties of an architectural knowledge sharing tool. The second one reflected on the impact of these architectural decisions, and proposed to estimate their consequences by using patterns, which should be extended to include additional information about quality attributes. The third paper discussed the issues in applying the principles of Empirical Software Engineering to Software Architecture. There are very few empirical studies within the field, and it is argued that concrete criteria must still be defined to decide when an architecture is good enough. Some additional issues were also identified.

Finally, the remaining paper presented a complete review of the state-of-the-art in Software Architecture, considering the case of large-scale complex systems. It first reviews academic research, its challenges and limitations, and then industrial research, exposing its contributions. To conclude, a research agenda for both, and their synergies, is proposed.

7. SERVICE-ORIENTED ARCHITECTURE

Though service-orientation was one of the main topics in the conference, only a few papers about services made their way into the final program.

The first paper described a simple approach to model service-oriented architectures using a conventional ADL. This is made by providing a new "web service" connector for an ArchJava extension, so that glue code is automatically generated. The second paper presented a concrete architecture, where services are used to interface to several distributed high-performance computing (HPC) systems, to combine them and to expose the result.

Finally, the keynote talk by Michael Papazoglou, can also be related to the topic of the session, though his approach was more conceptual, rather than simply technological.

8. POSTER SESSION

The poster session was intended from the start an integral part of the Conference's main program, instead of considering it as a side activity. In fact, most of the conference attendants participated in this session, which gathered a lot of interest. To some extent, the poster session could be considered as a general session on emerging research. Most of the topics were also considered by some papers in the main program, so there is a continuity between them. Sixteen different posters were presented, gathering authors from the UK, Mexico, Australia, Germany, Portugal, France, Italy, Brazil and Spain. The variety of their approaches, their interdisciplinary nature and the intertwinning of their topics made the session particularly dynamic and active. An international jury examined every poster and listened to the individual presentation of every author. After this evaluation, they granted the Best Poster Award to a work describing a goal-oriented methodology for the automatic synthesis and generation of *proto-architectures* from the initial requirements specification.

The most popular topic was the relationship and mutual influence of *software architecture and Model-Driven Engineering* (MDE) approaches; both the role of architectures within MDE, and the use of model-driven techniques in architecture. The second most popular topic was *aspect-oriented architecture*, that is, explicit separation of concerns in architecture, leading to non-modular structures and crosscutting models. This theme is already a classic of European research, and it is still very popular. The next topic was the study *architectural styles*, dealing both their impact on design, and the definition of new proposals. This was also generalized to a full model, and to concrete kinds of architecture. The fourth topic was *architecture evolution and dynamic reconfiguration*, also an important topic in the main program. Different techniques to support dynamism were proposed, namely reflective self-description, aspect-oriented dynamic reconfiguration, and event-based middleware. And finally, a single poster was devoted to the documentation and management of *architectural design decisions*, a topic also already considered within the main program.

In summary, six posters were devoted to the first topic, five (two of them indirectly) to the second one, four to the third, three to the fourth and one for the fifth theme. There was also a certain degree of overlapping between them, which helped to make the discussion even more interesting.

9. SUMMARY AND FUTURE WORK

The conference was an intense and active meeting and was perceived for the partipants as a great success. The European research community on Software Architecture has been definitely established and can be considered as a solid one, with its own character and specific features, which often differ from those from the international community.

The ECSA series of conferences will continue for years to come. The next edition will be held in Cyprus, and there are prospects until 2010. In 2009 it will be a joint event gathering ECSA and WICSA, the first and topmost conference on Software Architecture. This builds on the ongoing cooperation between the two communities, and confirms ECSA as the second international event in the field.

ACKNOWLEDGEMENTS

Our participation in this report has been partially funded by the Spanish Ministry of Education and Science (MEC) through National Research Projects GOLD (TIN2005-00010), META/MOMENT (TIN2006-15175-C05-01), and by Program CONSOLIDER, through the Project AT (CSD2007-00022).

10. REFERENCES

[1] F. Oquendo, editor. *Software Architecture, First European Conference (ECSA 2007) Proceedings*, volume 4758 of *Lecture Notes in Computer Science*, Aranjuez, Spain, Sept. 2007. Springer.

Report on International Workshop on Privacy and Anonymity in the Information Society (PAIS 2008)

Li Xiong
Department of Math & Computer Science
Emory University
lxiong@mathcs.emory.edu

Traian Marius Truta
Department of Computer Science
Northern Kentucky University
trutat1@nku.edu

Farshad Fotouhi
Department of Computer Science
Wayne State University
fotouhi@wayne.edu

1 Introduction

While the ever increasing computational power together with the huge amount of individual data collected daily by various agencies is of great value for our society, they also pose a significant threat to individual privacy. As a result legislators for many countries try to regulate the use and the disclosure of confidential information. Various privacy regulations (such as USA Health Insurance Portability and Accountability Act, Canadian Standard Association Model Code for the Protection of Personal Information, Australian Privacy Amendment Act 2000, etc.) have been enacted in many countries all over the world. Data privacy and protecting individual anonymity have become a mainstream avenue for research. While privacy is a topic discussed everywhere, data anonymity recently established itself as an emerging area of computer science. Its goal is to produce useful computational solutions for releasing data, while providing scientific guarantees that the identities and other sensitive information of the individuals who are the subjects of the data are protected.

The Workshop on Privacy and Anonymity in the Information Society (PAIS 2008)[1] was held on March 29, 2008, co-located with the 11th International Conference on Extending Database Technology (EDBT 2008) in Nantes, France. It was the first in its series and the mission of the workshop is to provide an open yet focused platform for researchers and practitioners from computer science and other fields that are interacting with computer science in the privacy area such as statistics, healthcare informatics, and law to discuss and present current research challenges and advances in data privacy and anonymity research.

[1] http://cscdb.nku.edu/pais/

2 Workshop Themes and Program

The workshop program included a keynote speech and 8 paper presentations. The paper presentations were divided into 3 sessions. There were 30 participants who attended the workshop. The workshop was very interactive, with the audience raising many questions for the speakers and a lively discussion following the technical presentations. Several overall themes emerged from the presentations and discussions, including location privacy, distributed privacy protection, query auditing, k-anonymization and its applications, and micro-aggregation. We report and discuss each of them below.

2.1 Location Privacy

The proliferation of mobile communications is leading to new services based on the ability of providers to determine, with increasing precision, the geographic location of the accessing device. While these applications and services promise enormous consumer benefit, privacy concerns abound, and must be addressed

before new services and applications are accepted by consumers.

The keynote speech addressed the timely topic of location privacy. The talk was given by Josep Domingo-Ferrer, professor of Computer Science from Rovira i Virgili University of Tarragona, Catalonia, and the UNESCO Chair in Data Privacy. The talk was titled "Location privacy via unlinkability: an alternative to cloaking and perturbation" [4]. In the talk, he summarized that the usual approach to location privacy is to cloak and/or perturb the positions or trajectories of the mobile objects. However, he argued that if unlinkability of the various interactions between a mobile object and the service or control system can be afforded and achieved, neither cloaking nor perturbation is unnecessary. The unlinkability results in higher privacy for the mobile object and better accuracy of the aggregated mobility information gathered by the service/control system. He illustrated the feasibility of the approach in the scenario of car-to-car communication.

2.2 Distributed Privacy Protection

Distributed privacy-preserving data mining deals with data sharing across multiple distributed data sources for specific mining tasks. The problem is a specific example of the secure multi-party computation (MPC) problem. In MPC, a given number of participants, each having a private data, wants to compute the value of a public function. A MPC protocol is secure if no participant can learn more from the description of the public function and the result of function. While there are general secure MPC protocols, they require substantial computation and communication costs and are impractical for multi-party large database problems.

The paper titled "Distributed Privacy Preserving k-Means Clustering with Additive Secret Sharing" [3] considered a distributed privacy preserving data mining scenario where the data is partitioned vertically over multiple sites and the involved sites then perform clustering without revealing their local databases. For this setting, the authors proposed a new protocol for privacy preserving k-means clustering based on additive secret sharing. They showed that the new protocol is more secure than the current state of the art while the communication and computation cost is considerably less which is crucial for data mining applications.

Protecting privacy for content-sharing P2P networks is another distributed privacy protection problem of increasing importance. The privacy issues in this context include anonymity of uploaders and downloaders, linkability (correlation between uploaders and downloaders), content deniability, data encryption and authenticity, and data disclosure.

The paper titled "Design of PriServ, a privacy service for DHTs" [5] addressed the data disclosure problem in P2P networks. When sharing data for different purposes, data privacy can be easily violated by untrustworthy peers which may use data for unintended purposes. A basic principle of data privacy is purpose specification which states that data providers should be able to specify the purpose for which their data will be collected and used. The work applied the Hippocratic database principles to P2P systems to enforce purpose-based privacy. They focused on Distributed Hash Tables (DHTs), and proposed PriServ, a privacy service which prevents privacy violation by prohibiting malicious data access. The performance evaluation of the approach through simulation shows that the overhead introduced by PriServ is small.

2.3 Query Auditing

Research in statistical databases is focused on enabling queries on aggregate information (e.g. sum, count) from a database without revealing individual records. A key technique is query restriction which includes schemes that check for possible privacy breaches by keeping audit trails and controlling overlap of successive aggregate queries.

The paper titled "A Bayesian Approach for on-Line Max and Min Auditing" [1] considered the online max and min query auditing problem. Given a private association between fields in a data set, a sequence of max and min queries that have already been posed about the data, their corresponding answers and a new query, the objective is to deny the answer if a private information is inferred or give the true answer otherwise. The authors gave a probabilistic definition of privacy and demonstrated that max and min queries, without no duplicates assumption, can be audited by means of a Bayesian network. Moreover, the auditing approach is able to manage user prior-knowledge.

2.4 K-Anonymization and its Applications

Data anonymization has been extensively studied in recent years and a few principles have been proposed that serve as criteria for judging whether a published dataset provides sufficient privacy protection. Notably, the seminal work of k-anonymity, requires a set of k records (entities) to be indistinguishable from each other based on a quasi-identifier set. A large body of work contributes to transforming a dataset to meet a privacy principle (dominantly k-anonymity) using techniques such as generalization, suppression (removal), permutation and swapping of certain data values while minimizing certain cost metrics. The workshop features a few papers on k-anonymization and its applications in novel domains.

The paper titled "Protecting Privacy in Recorded Conversations" [2] considered the problem of privacy protection in the domain of speech technology. While speech corpora are important to professionals in the field of speech technology, they are often prevented from being shared due to privacy rules and regulations. Efforts to scrub this data to make it shareable can result in data that has been either inadequately protected or data that has been rendered virtually unusable due to the loss resulting from suppression. This work attempted to address these issues by developing a scientific workflow that combines proven techniques in data privacy with controlled audio distortion resulting in corpora that have been adequately protected with minimal information loss.

The paper titled "Data Utility and Privacy Protection Trade-off in K-Anonymization" [7] revisited the issue of balancing between data utility and privacy for k-anonymization. While existing methods try to maximize utility while satisfying a required level of protection, their work attempted to optimize the trade-off between utility and protection. The authors introduced a measure that captured both utility and protection, and an algorithm that exploited this measure using a combination of clustering and partitioning techniques. The author showed that the method is capable of producing k-anonymization with required utility and protection trade-off and with a performance scalable to large datasets.

The paper titled "An Efficient Clustering Method for k-Anonymization" [6] proposed a new clustering method for k-anonymization. The authors argued that in order to minimize the information loss due to anonymization, it is crucial to group similar data together and then anonymize each group individually. The work proposed a clustering based anonymization method and compared it with another state-of-the-art clustering based k-anonymization method. Their experiments and analysis showed that the proposed method outperforms the existing method with respect to time complexity, information loss and resilience to outliers.

2.5 Microaggregation

Microaggregation is a hot topic in the field of Statistical Disclosure Control (SDC) and one of the most employed microdata protection methods. The main idea is to build clusters of at least k original records, and then replace them with the centroid of the cluster. This is one way to achieve k-anonymity.

The paper titled "Attribute Selection in Multivariate Microaggregation" [8] addressed the issue of attribute grouping for microaggregation. When the number of attributes is large, a common practice is to split the dataset into smaller blocks of attributes. This paper showed that, besides the specific microaggregation method employed, the value of the parameter k, and the number of blocks in which the dataset is split, there exists another factor which can influence the quality of microaggregation: the way in which the attributes are grouped to form the blocks. When correlated attributes are grouped in the same block, the statistical utility of the protected dataset is higher. In contrast, when correlated attributes are dispersed into different blocks, the achieved anonymity is higher. The authors also presented quantitative evaluations of such statements.

The paper titled "Micro-aggregation-Based Heuristics for p-sensitive k-anonymity: One Step Beyond" [9] adapted micro-aggregation based techniques for p-sensitive k-anonymity, a recently defined sophistication of k-anonymity. The p-sensitive k-anonymity requires that there be at least p different values for each confidential attribute within the records sharing a combination of key attributes. While the original algorithm was based on generalizations and suppressions, this work show that k-anonymity and p-sensitive k-anonymity can be achieved in numerical data sets by means of micro-aggregation heuristics properly adapted to deal with this task. The authors presented and evaluated two heuristics for p-sensitive k-anonymity which, being based on micro-aggregation, overcame most of the drawbacks resulting from the generalization and suppression method.

3 Final Remarks

PAIS 2008 workshop is among several series of recent workshops focusing on the various issues of data privacy and security including: Secure Data Management Workshop (SDM) co-located with VLDB, International Workshop on Privacy, Security and Trust (PinKDD) co-located with SIGKDD, International Workshop on Privacy Data Management (PDM) co-located with ICDE, Workshop on Privacy Aspects of Data Mining (PADM) co-located with ICDM, and Practical privacy preserving data mining (P3DM) co-located with SDM.

The PAIS workshop is unique in that it focuses on the topic of privacy and in particular anonymity in the general information society. The workshop organizers and attendees envision a series of workshops building upon the success of this workshop.

4 Acknowledgements

Putting together PAIS 2008 was a team effort. First of all, we would like to thank our keynote speaker and the authors for providing the quality content of the program. In addition, we would like to express our gratitude to the program committee, who worked very hard in reviewing papers and providing suggestions for their improvements. Finally, we would like to thank the UNESCO Chair in Data Privacy for their travel support and the EDBT 2008 conference for their support of the workshop.

References

[1] G. Canfora and B. Cavallo. A bayesian approach for on-line max and min auditing. In *PAIS*, pages 12–20, 2008.

[2] S. Cunningham and T. M. Truta. Protecting privacy in recorded conversations. In *PAIS*, pages 26–35, 2008.

[3] M. C. Doganay, T. B. Pedersen, Y. Saygin, E. Savas, and A. Levi. Distributed privacy preserving k-means clustering with additive secret sharing. In *PAIS*, pages 3–11, 2008.

[4] J. Domingo-Ferrer. Location privacy via unlinkability: an alternative to cloaking and perturbation. In *PAIS*, pages 1–2, 2008.

[5] M. Jawad, P. Serrano-Alvarado, and P. Valduriez. Design of priserv, a privacy service for dhts. In *PAIS*, pages 21–25, 2008.

[6] J.-L. Lin and M.-C. Wei. An efficient clustering method for k-anonymization. In *PAIS*, pages 46–50, 2008.

[7] G. Loukides and J. Shao. Data utility and privacy protection trade-off in k-anonymisation. In *PAIS*, pages 36–45, 2008.

[8] J. Nin, J. Herranz, and V. Torra. Attribute selection in multivariate microaggregation. In *PAIS*, pages 51–60, 2008.

[9] A. Solanas, F. Sebé, and J. Domingo-Ferrer. Micro-aggregation-based heuristics for p-sensitive k-anonymity: one step beyond. In *PAIS*, pages 61–69, 2008.

Report on the IFIP WG5.8 International Workshop on Enterprise Interoperability (IWEI 2008)

Marten van Sinderen
University of Twente
m.j.vansinderen@utwente.nl

Pontus Johnson
Royal Institute of Technology (KTH)
pontus@ics.kth.se

Lea Kutvonen
University of Helsinki
lea.kutvonen@cs.helsinki.fi

1. Motivation and History

One of the trends emerging from globalization is the increasing collaboration among enterprises. An enterprise, in this context, is an organization or a collection of organizations with a mission, goals and objectives to offer an output such as a product or service.

Organizations, such as large companies but also SMEs, have to cope with internal changes concerning policies, organizational structure and IT support. In addition, organizations have to flexibly and continuously react to changes in markets, trading partners and trading channels. Such changes in intra- and inter-organizational environments are expected to persist if not intensify in the future. Changes occur at different levels, notably at a business level (focusing on profit, merges, re-organizations, outsourcing, resource sharing etc.) and at a technology level (focusing on application software, IT infrastructure, software and hardware technologies etc.). More importantly, although most changes will originate at a specific level, i.e. they are business-driven or technology-driven, they affect all other levels.

The competitiveness of an organization thus not only depends on its internal performance to produce products and services but also on its ability to seamlessly interoperate with other organizations. This involves internal and external collaboration for which we need enterprise interoperability solutions. A major challenge is then to achieve and sustain multi-level interoperability in the face of planned and spontaneous changes, with proper alignment between and integrity of the different levels.

The International Workshop on Enterprise Interoperability, IWEI, aims at identifying and discussing challenges and solutions with respect to enterprise interoperability, both at the business and the technical level. The workshop promotes the development of a scientific foundation for specifying, analyzing and validating interoperability solutions; an architectural framework for addressing interoperability problems from different viewpoints and at different levels; a maturity model to evaluate and rank interoperability solutions with respect to distinguished quality criteria; and a working set of practical solutions and tools that can be applied to interoperability problems to date.

The IWEI workshop was organized by the IFIP Working Group 5.8 on Enterprise Interoperability, and was held in conjunction with the 12th IEEE International EDOC Conference (EDOC 2008).

The theme and scope of IWEI is very much linked to that of IFIP WG5.8. IFIP WG5.8 reached its current status of Working Group in September 2008, after a Special Interest Group status of 2 years. The SIG was motivated by the perceived interoperability problems that prevent seamless collaboration among organizations. Such problems primarily emerge from proprietary development or extensions, unavailability or oversupply of standards, and heterogeneous hardware and software platforms. But there is also no well-founded overall approach to address interoperability issues across system levels and stakeholders views. Consequently, new collaboration endeavors are hindered, and achieved interoperability is costly. Despite the efforts already spent to overcome interoperability problems, interoperability is recognized by organizations as a major concern and solving interoperability problems represents a considerable portion (over 30%) of their IT costs [5]. On the other hand, opportunities for value creation based on enterprise interoperability have drastically increased [4]. More flexible enterprise interoperability solutions would allow for profitable strategies targeting differentiated products and services in dynamic value networks [1, 2].

2. Contributions

The papers selected for oral presentation at the IWEI workshop [6] were scheduled in 3 sessions A fourth session was devoted to an open discussion on future challenges, involving both the speakers and participants, based on previously submitted statements of the speakers.

2.1 Session 1: Ontologies and the Semantic Web

Successful enterprise interoperability depends on the proper understanding of information that is exchanged between organizations. This explains the interest in ontologies that formalize universes of discourse, and in the semantic web that provides technology for dealing with meaning in service networks.

In their paper “Framework for interoperability analysis on the semantic web using architecture models” J. Ullberg et al. describe a framework for assessing service interoperability over the semantic web. The authors claim that interoperability is influenced by five factors: transmission protocol compatibility, discoverability, ontology completeness, quality of formal denotation markup, and quality of requirements description markup.

Extended influence diagrams are used in the framework to capture the relations between the various interoperability factors and to enable aggregation of these into an overall interoperability measure.

N. Zouggar et al. explain how conflicts can arise with the interpretation of enterprise models after their creation. Such conflicts potentially lead to problems with the operation of organizations, including their interoperability, based on these models. In their paper "Semantic enrichment of enterprise models by ontologies based semantic annotations" the authors identify interpretation conflicts that can occur and propose a systematic approach for semantically enriching enterprise models using ontologies. The approach for semantic enrichment consists of 6 steps, which are detailed in the paper.

2.2 Session 2: Inter-organizational interoperability

As a consequence of market globalization, collaboration between organizations has drastically increased, and the ability to collaborate with other organizations can be key for an organization's competitiveness.

Truyen and Joosen propose in their paper "A reference model for cross-organizational coordination architecture" a reference model for the coordination of service provisioning across organizational boundaries. This reference model supports comparison and analysis of existing coordination architectures, and allows proposals for their improvement. The reference model has 3 main dimensions: type of agreement, language for describing agreements, and middleware for establishing and executing agreements. With the proposed reference model 7 different coordination architectures are compared and potential improvements are identified.

In the paper "Design of services as interoperable systems – an e-commerce case study" Kassel presents important principles of a decision support model for composing reliable software systems from service components. The decision support model can be used to guide the negotiation between a service provider and customer, explicitly addressing interoperability and price issues. The presented work is part of a project under development in cooperation with an industrial partner.

Santana Tapia et al. present a maturity model for collaborative networked organizations in their paper "Towards a business-IT aligned maturity model for collaborative networked organization". The maturity model allows collaborating organizations to assess the current state of business-IT alignment and take appropriate action to improve the alignment where needed. The problem of alignment has so far hardly been studied in networked organizations. The proposed model is a first version derived from various alignment models and theories.

2.3 Session 3: Service-orientation

Flexible enterprise interoperability requires a proper architectural foundation for developing and connecting IT systems to support collaboration. Service-oriented architecture (SOA) could be such an architecture, having an already established web services technology base.

Elvesæter et al. present the vision and initial results of the EU project COIN in their paper "Towards enterprise interoperability service utilities". The project develops open source services, which will be integrated into a coherent pool of enterprise interoperability services according to the Interoperability Service Utility challenge of the Enterprise Interoperability Roadmap. This is seen as a contribution to the Software-as-a-Service Utility (SaaS-U) vision. An enterprise interoperability services framework has been defined, based on previous work from the ATHENA project. Following a state of the art analysis, a set of baseline enterprise interoperability services were specified. These services will be implemented as semantic web services and tested in industry pilots.

Mantovaneli Pessoa et al. propose a conceptual framework for service composition. They discuss this framework in their paper "Enterprise interoperability with SOA: a survey of service composition approaches", considering enterprise interoperability issues related to service composition and different phases of the service composition lifecycle. Five different service composition approaches are described and compared using the framework. The results indicate that none of the approaches cover all the lifecycle phases, but mainly focus on service design-time phases while neglecting others like support for end-user service composition at run-time.

In the paper entitled "Model-driven development of a mediation service" Quartel et al. present a framework to guide the development of mediators for service-based business collaboration. The framework has the following objectives: (i) uncover and capture the actual interoperability problem that needs to be solved; (ii) allow the involvement of non-IT experts in the development of the solution; (iii) support evolution of the solution and re-use of results in case of changing interoperability requirements; (iv) facilitate automation of parts of the process. Available tool support for the different steps in the framework is indicated, and has been demonstrated at workshops of the Semantic Web Services Challenge.

2.4 Session 4: Challenges of enterprise interoperability

Several challenges and future development issues were identified during this session:

Interoperability

- Methods and tools for interoperability assessment are needed. These can utilize for example model-based techniques and influence diagrams.

- The field of interoperability should also incorporate pragmatics, or context-sensitivity. Pragmatics is a domain that should be addressed, not a solution itself.
- Business-IT alignment and interoperability are two non-reachable goals bound together. The key is to keep improving them in iterations. A helpful tool is the development and use of maturity models.

Enterprise models

- In enterprise modeling, one of the major problems is catching the semantics of the model in order to minimize misunderstandings when the model is communicated. Techniques that can be used here include annotations, ontologies, formalization, and iteration of refinement and verification.

Platforms and architectures

- Current coordination architectures are not very mature. Comparison of them using a reference model shows major differences on the addressed scope and concepts (such as the key concept of contract).
- Platforms should utilize both behavior interoperability and semantic interoperability. Both lead to a reference ontology and a set of utilities for model interchange, cross-organizational process coordination, information exchange and assessing the maturity of interoperability.
- Platform-specific solutions for service composition start to emerge, but they still lack full coverage of all the service composition lifecycle phases.

Overarching issues

- Will computing service providers allow an open service market to emerge? The emergence of an open service market requires that the disciplines of IT and service governance, business-driven IT management (system adaptation, service selection), validation of software engineering and business value are more closely intertwined. In order to support service compositions for real business cases there needs to be an environment to integrate services from different companies, identify fitting services, choose between them, integrate the service processes, and provide decision-support for providers and customers.
- Have we addressed the category of "meta-interoperability" problems, caused by the diversity of platforms, methods, languages etc. that have been developed to solve interoperability problems?

3. Conclusion

Enterprise interoperability is a growing research topic, rooted in various sub-disciplines from computer science and business management. Enterprise interoperability addresses intra- and inter-organizational collaboration and is characterized by the objective of aligning business level and technology level solutions and reconciling viewpoints of the different stakeholders. Enterprise interoperability comprises issues which are not yet well-understood and an overall framework for dealing with these issues is still lacking. On the other hand, enterprise interoperability is an essential property of organizations to have successful business operations in the globalized market. This is also recognized by the European Union, as demonstrated by various study reports [3].

We believe that the IWEI workshop provides a useful forum for researchers and practitioners to discuss enterprise interoperability solutions and challenges, and, in collaboration with IFIP WG5.8, to address the identified challenges in future work. The next IWEI workshop is planned in Valencia, Spain, 29 Sept. - 1 Oct. 2009.

4. Acknowledgments

IFIP WG 5.8 on Enterprise Interoperability was instrumental to the organization of IWEI 2008. Not only was IFIP WG5.8 the formal organizer, its members and especially its chairman, Guy Doumeingts, provided useful ideas and crucial support that made IWEI 2008 a successful event. We also like to thank the members of the program committee for carefully and conscientiously scrutinizing the submitted papers and providing useful feedback.

5. References

[1] Allee, V. 2002 A value network approach for modelling and measuring intangibles. In Proceedings of Transparent Enterprise - The Value of Intangibles (Madrid, Spain, 25-26 Nov. 2002).

[2] Chesbrough, H. 2007 Open business models: how to thrive in the new innovation landscape. Harvard Business Press, USA.

[3] European Commission 2006 Enterprise interoperability research roadmap. Version 4.0 (July 2006). Available at: ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/ebusiness/ei-roadmap-final_en.pdf

[4] European Commission 2008 Unleashing the potential of the European knowledge economy. Value proposition for enterprise interoperability. Version 4.0 (Jan. 2008). Available at: ftp://ftp.cordis.europa.eu/pub/ist/docs/ict-ent-net/isg-report-4-0-erratum_en.pdf

[5] Grilo, A., Jardim-Goncalves, R. and Cruz-Machado, V. 2007 A framework for measuring value in business interoperability. In Proceedings of the IEEE Intl. Conf. on Industrial Engineering and Engineering Management (IEEM 2008), pp. 520-524.

[6] Sinderen, M.J. van, Johnson, P. and Kutvonen, L. (Eds.) 2008 Proceedings of the Int. Workshop on Enterprise Interoperability (IWEI 2008). CTIT Workshop Proceedings Series WP08-05. University of Twente, Enschede,.The Netherlands.

First Workshop on Very Large Digital Libraries – VLDL 2008

Paolo Manghi
ISTI - CNR
Pisa, Italy
paolo.manghi@isti.cnr.it

Pasquale Pagano
ISTI - CNR
Pisa, Italy
pasquale.pagano@isti.cnr.it

Pavel Zezula
Masaryk University
Brno, Czech Republic
zezula@fi.muni.cz

1. MOTIVATIONS

In today's information society the demand for Digital Libraries is changing. The implementation of nowadays Digital Libraries is more demanding than in the past. Information consumers are facing with the need to have access and elaborate over an ever growing and heterogeneous information space while information providers are interested in satisfying such needs by providing rich and organised views over such information deluge. Because of their fundamental role of information production and dissemination vehicle, Digital Libraries are also expected to provide information society with services that must be available 24/7 and guarantee the expected quality of service.

This scenario leads to the development of Large-Scale Digital Library Systems in terms of distribution, integration and provision of services, information objects, end-users and policies of use. Such systems have to confront with new challenges in a context having scalability, interoperability and sustainability as focal points.

The need for concrete solutions can be seen also in the substantial amount of resources invested by the European Union towards the creation of a unifying European Information Space, starting with DELOS [11], BRICKS [10], DILIGENT [6] and DRIVER [4] in the past, and continuing with D4Science [5], DRIVER-II, CLARIN [8], SAPIR [15] and finally with the European Digital Library, a major effort to build a European-scale digital library to make available to everybody the rich cultural assets of the whole Europe.

New approaches and technologies have been devised to appropriately tackle the various matters arising in designing, developing and deploying VLDL systems. The goal of this workshop was to provide researchers, practitioners and application developers with a forum fostering a constructive exchange among all of such key actors.

2. WORKSHOP OUTLINE

The workshop structure comprised an invited speakers session followed by the presentation of the nine accepted contributions, organized into three sessions: *architectures*, *services* and *data management* for VLDLs.

2.1 Invited presentations

The organizers invited two speakers, respectively in the field of content modeling and architecture design for VLDLs. Both presentations had a foundational flavour, with the purpose of encouraging discussions on common patterns, best practices and methodologies in VLDLs. The first presentation by *Carlo Meghini* (ISTI-CNR) focused on representation models for Complex Objects on VLDL scenarios. The second presentation by *Daan Broeder* (CLARIN Project, Max-Planck Institute) illutrated common challenges to be faced in building real, very-large infrastructures for language resources management.

2.2 Architectures for VLDLs

VLDL architectural issues have to do with organizational models, interoperability, integration, federation, sustainability, scalability, quality of service, policies and how these issues may combine and be solved in the context of specific application domains and research communities. The session presented three experiences in architectural design, respectively focusing on service and content management patterns in VLDLs, VLDL cataloging systems, and organizational and policies issues in VLDLs.

The first presentation, by *Andreas Aschenbrenner* (Max-Planck Digital Library labs), introduced a Digital Library System Warehouse framework inspired by successful patterns for large-scale digital libraries definition. The framework combines two common practices: (*i*) integration of external services, e.g. search, which may be part of the core user requirements, but reside outside of the core architecture; and (*ii*) manipulation of distributed digital objects. As examples of the two patterns, two system architectures where cited: the *eSciDoc* project [2] aims at building an integrated information, communication and publishing platform for web-based scientific work, exemplarily demonstrated for multi-disciplinary applications; the *TextGrid* project [3] supports a collaborative research environment for specialist texts. A beta-version containing about 10 terabyte of objects initially (images, XML-based full text, annotations, etc) and an initial, expandable set of functionality went live in September 2008.

The second presentation, by Gianmaria Silvello (Department of Information Engineering, University of Padua), described the design of a Digital Library System able to collect, manage and share very large archival metadata collections in a distributed environment. Archive characteristics were pre-

sented, where size, interoperability and heterogeneity were pointed out as the most relevant and peculiar challenges for the architecture design. The work also included an extension to the architecture, so as to include management of special Compound Digital Objects in the archival context.

The last presentation of the session, by Mary Rowlatt (MDR Partners), presented the *EuropeanaLocal* project [7] and its part in the *Europeana* Digital Library architectural framework [14]. In particular, the project plays an important role in ensuring that the enormous amount of digital content (from museums, libraries archives and archives of images, sound, text and movies) provided by Europes cultural institutions at local and regional level is represented in Europeana, alongside that held at national level. The expected results include (*i*) the establishment of a network of regional repositories that are highly interoperable with Europeana, (*ii*) an integrated Europeana-EuropeanaLocal prototype service and (*iii*) the development of thematic areas for Europeana services which integrate content from both the national and the local/regional level.

2.3 Data management in VLDLs

VLDL data management issues regard content-related aspects, such as services for manipulation, integration, storage, access, search and federation of data in VLDLs. General-purpose solutions are of interest, as well as others specific to given application domains. The session presented three different experiences in data integration, data storage and access, and data ranking in VLDLs.

The first presentation, by Daan Broeder (Max-Planck Institute), focused on the research activities carried out in past projects at the Max-Planck Institute, regarding management and integration of very large heterogeneous multimedia archives. The activities should in synergy serve the purpose of the CLARIN European project: the main issues regarded data models and interoperability (DOBES project), data archiving (LAT project) and synchronization (DOBES project), single sign-on access (DAM-LR EU project) and persistent identifiers.

The second presentation illustrated the Greenstone system [12] experience with the *Papers Past newspaper collection* [13]. This collection, containing 670,000 newspaper digitalized pages (7.5 million articles), growing to approximately 1.2 million pages over time, counting 20Gb of raw searchable text, 2 billion words, 60 million unique terms and 52Gb of metadata is (almost) certainly the largest Greenstone collection ever built. In this scenario, Greenstone developers had to cope with the large quantities of images to be analyzed and with the peculiarity of large number of unique terms to be indexed, which degraded the performance of the standard Greenstone system.

The last presentation, by Mikalai Krapivin (University of Trento), showed the results and benefits of a new ranking algorithm applicable to very large pools of scientific papers. The algorithm, named Focused Page Rank, proposes a trade-off between traditional citation count and basic Page Rank (PR) algorithms. The author believes this solution to be closer to the expectations of real users because, in accordance with the one of the most significant principles of Scientometrics, highly cited papers tend to be more visible in the results and thus attract more citations in future. The rank evaluation technique is scalable and may be applied to very large libraries.

2.4 Services for VLDLs

VLDL service issues include the design and development problems arising in the realization of functionalities for VLDLs. The session presented results in designing and developing three service typologies: a loan service, a store service and user interface service.

The first presentation, by *Ciro D'Urso* (Italian Senate), presented the design of an event-based loan service that provides a single access point over distributed and autonomous digital libraries of textual or electronic or microform books, music, sound recordings, visual materials. The service, currently under experimental use to integrate catalogs from Italian libraries, is designed to scale with the number of data sources and registered users and to cope with the interoperability issues introduced by the different catalog standard and technologies.

The second presentation, by *Stephen Green* (British Library), illustrated the inter-related challenges in building long-term store services for very large document collections in the British Library. The important features of the service, developed by the Library labs, are: (*i*) privileging uninterrupted access to stored objects to continuity of ingest, by supporting disaster tolerance and recovery functionalities; (*ii*) automatic self-monitoring, with the ability to recover damaged files within a store; (*iii*) design independent of any hardware manufacturer, in order to allow storage units to be swapped in and out as required; (*iv*) digital signing techniques to deliver continuous assurance of the authenticity of stored objects from the time of ingest to any future time; and (*v*) metadata-driven management of versioning and successor objects.

The third presentation, by *Massimiliano Assante* (ISTI-CNR, Pisa) presented portal services capable of dynamically integrating and adapting user interface capabilities from functionality services within an e-Infrastructure. e-Infrastructures are very large dynamic service-based environments where user communities can build applications exploiting a set of functionality services that can join or leave the system any time. In this scenario, communities may require centralized web-usage and access to a tailored subset of such functionalities. Due to the dynamic environment, building from scratch centralized portals can be very expensive, due to inevitable maintenance cost. Portal services offer a way to automatically configure a centralized portal that responds to specific user interface needs, based on the functionality services currently available.

3. WORKSHOP CONCLUSIONS

"What exactly are Very Large Digital Libraries?" Some, to answer this question, blur the separation between Very Large Databases (VLDBs) and Very Large Digital Libraries (VLDLs) and regard the latter as VLDBs storing Digital Library content. Since in databases the adjective "Very large" strictly refers to "size of content" (nowadays about 1 Terabyte of space), the implication is that, similarly, VLDLs

ought to be DLs storing digital content beyond a given threshold.

Despite being intuitively correct, this answer only partly satisfies DLs practitioners. Indeed, DLs design paradigms cannot be conceptually separated by the relative applications as it happens for DBs. DLs are of use to peculiar user communities whose functionality needs, best practices and behavior are well-accepted DL systems requirements. As captured by the DELOS reference model for DLs, user management, content management, functionality management, and policies are equally important in the definition of a DL. Accordingly, as demonstrated by real DL system experiences, VLDLs should be described as DLs featuring "very large features" in one or more of such aspects.

Models and measures for evaluating the "very-largeness" of the DL-axes content, functionality, users, and policies, are still an open issue. In this respect, the following observations and open questions naturally surface:

- What are the very large criteria for the DL-axes? How can such criteria be described and measured? What is their interrelationship?
- Should the definition of VLDLs be absolute or relative? If relative, for example, being "very large in content" should not depend on a given number, but on a number whose calculation depends on the limits of current technology (e.g. 20 times the average memory limit).
- Should the definition of VLDLs depend on challenge and complexity of design? If so, for example, DLs that can be built with existing solutions could not be in principle very large. The intuition is that "very large" equates to "unsolved because of inherent complexity" in one of the DL-axes.
- Should the definition of VLDL depend on federative aspects of DLs? In that sense, DLs would be very large whenever the user communities involved require the integration of a set of DLs at the level of one or more of the DL-axes.

The lesson learned from the workshop presentations and final discussion, is that VLDL research candidates to be an independent DL topic but still is in its early stage. It is to be investigated whether the foundations required for the consolidation of a research field per-se can be found in the common patterns and best practices of extant DL technologies or instead we still have to wait for more practical experience to come. These matters reveal a number of novel and interesting research avenues, from foundational to applicative, which will certainly be among the call topics of the Second Workshop on Very Large Digital Libraries next year.

4. PROGRAM COMMITTEE

Finally, our special gratitude goes to the members of the Program Committee: *Stefan Gradman* (Institut fr Bibliotheks und Informationswissenschaft, Humboldt-Universitt zu Berlin, Germany), *Kat Hagedorn* (OAIster System, University of Michigan Digital Library Production Service, USA), *Dean B. Krafft* (National Science Digital Library Project, Cornell Information Science, USA), Yosi Mass (IBM Research Division, Haifa Research Laboratory, University Campus, Haifa, Israel), *Yannis Ioannidis* (Department of Informatics, National and Kapodistrian University of Athens, Greece), *Peter Wittenburg* (Max-Planck-Institute for Psycholinguistics, The Netherlands), whose long research experience contributed in making this workshop an attractive and fruitful experience for all authors and participants.

Workshop proceedings [1] were printed by the DELOS Association for Digital Libraries [11].

5. REFERENCES

[1] Paolo Manghi, Pasquale Pagano and Pavel Zezula. Proceedings of the First Workshop on Very Large Digital Libraries, held in conjunction with ECDL 2008, Aarhus, Denmark, 2008

[2] Max-Planck Institute and FIZ Karlsruhe eSciDoc, funded by the German Federal Ministry of Education and Research (BMBF). http://www.escidoc-project.de

[3] Andreas Aschenbrenner Editing, analyzing, annotating, publishing: TextGrid takes, the a, b, c to D-Grid. In: iSGTW 30 January 2008, Jg. 54

[4] DRIVER: Digital Repository Infrastructure Vision for European Research. http://www.driver-community.eu

[5] D4Science: D4Science Project: DIstributed colLaboratories Infrastructure on Grid ENabled Technology 4 Science. http://www.d4science.eu

[6] DILIGENT: DILIGENT; a DIgital Library Infrastructure on GRID ENabled Technology. http://www.diligentproject.org

[7] EuropeanaLocal: Best Practice Network project, funded under the eContentplus programme. http://www.europeanalocal.eu

[8] CLARIN: Common Language Resources and Technology Infrastructure. http://www.clarin.eu

[9] EFG: European Film Gateway Project. http://www.europeanfilmgateway.eu

[10] BRICKS: Building Resources for Integrated Cultural Knowledge Services. http://www.brickscommunity.org

[11] DELOS: Digital library rEference modeL and interOperability Standards. http://www.delos.info

[12] Greenstone: Digital library Repository Software. http://www.greenstone.org

[13] Paper Past newspaper collection: http://paperspast.natlib.govt.nz

[14] Europeana: Connecting cultural heritage. http://www.europeana.eu

[15] SAPIR: Search In Audio Visual Content Using Peer-to-peer IR . http://www.sapir.eu

First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008)

Fernando Silva Parreiras*
ISWeb, University of Koblenz-Landau
Universitatsstr. 1
56070 Koblenz, Germany
parreiras@uni-koblenz.de

Jeff Z. Pan
Department of Computing Science, The University of Aberdeen
Aberdeen AB24 3UE
jpan@csd.abdn.ac.uk

Uwe Assmann
Institute for Software- and Multimedia-Technology
TU Dresden
D-01062 Dresden, Germany
uwe.assmann@tu-dresden.de

Jakob Herinksson
Institute for Software- and Multimedia-Technology
TU Dresden
D-01062 Dresden, Germany
jakob.henriksson@tu-dresden.de

ABSTRACT

The First International Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008), affiliated with the 11th International Conference on Model Driven Engineering Languages and Systems (MoDELS2008), brought together researchers and practitioners from the modeling community with experience or interest in MDE and in Knowledge Representation to discuss about: (1) how the scientific and technical results around ontologies, ontology languages and their corresponding reasoning technologies can be used fruitfully in MDE; (2) the role of ontologies in supporting model transformation; (3) and how ontologies can improve designing domain specific languages.

1. INTRODUCTION

As Model Driven Engineering spreads, disciplines like model transformation and domain specific modeling become essential in order to support different kinds of models in an model driven environment. Understanding the role of ontology technologies like knowledge representation, automated reasoning, dynamic classification and consistence checking in these fields is crucial to leverage the development of such disciplines.

Thus, the objectives of the First International Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008) were to present success cases of integrated approaches and state-of-the-art researches covering ontologies in MDE; and to encourage the modeling community to explore different aspects of ontologies.

TWOMDE 2008 addressed how the scientific and technical results around ontologies, ontology languages and their corresponding reasoning technologies can be used fruitfully in MDE. More specifically, TWOMDE 2008 discussed the infrastructure for integrating ontologies and MDE and the application of ontologies in the following aspects of MDE: Domain Specific Languages, Design Patterns, and Conceptual Modeling.

This first edition counted on one invited talk and five paper presentations. The audience comprised 20 participants. Senior researchers and professors constitute at least half of audience. It indicates that the modeling community is willing to know about the integration of Ontologies and MDE.

This paper assesses the TWOMDE 2008 achievements as follows: Section 2 covers an analysis of papers presented in the workshop. Section 3 analyzes open questions and summarizes the discussions handled at the end of the workshop. Section 4 points to synergies and areas of interest to be covered in future editions of the workshop.

2. RESEARCH PAPERS

The workshop was divided in three parts. It started with the invited talk about about "potential applications of ontologies and reasoning for modeling and software engineering" following by a group of papers concerning application of ontologies in MDE.

Andreas Friesen gave a keynote talk about the experience of SAP's potential applications for ontologies and reasoning for enterprise applications [2]. Participating of at least five EU projects on the topic, SAP has collected a large number of potential applications. We illustrate two of them: Dynamic Integration of logistic service providers and business process composition.

The first involves the usage of semantic web services and ontologies to automatically find the most appropriate web service based on predefined requirements. This application replaces multiple manual steps for discovery and selection of suitable web services.

The second potential application relies on employing ontologies in business process composition. When composing business processes, currently, there is a manual effort in en-

*Supported by CAPES Brazil and EU STreP-216691 MOST.

suring the correct message flow and integration logic among business processes. Applying ontologies may allow for semi-automatic generating the message flow for consistent execution.

An open issue is how to measure the value added by ontologies. Indeed, although the role of ontologies is clear, metrics to assess the impact of ontologies on enterprise systems lack so far. Ongoing EU projects like MOST[1] may contribute with use cases and patterns to support this issue.

2.1 Applications of Ontologies in MDE

Papers addressing the application of ontologies in MDE cover topics like design pattern integration, domain specific languages and multi-agent systems.

Cédric Bouhours presented the use of "an ontology to suggest design patterns integration" [3]. The paper analyses the application of an extended Design Pattern Intent Ontology (DPIO) in an pattern integration process. The process is composed by three steps: Alternative models detection, Validation of the propositions and Patterns integration. The DPIO ontology is used in the second step to validate the suggestions made. A future work would be the semi-automatic detection of alternative models by ontology. This task would make use of reasoning to infer relationships between the the model and the alternative model.

Another interesting application of ontologies is in "the domain analysis of domain-specific languages"[6], presented by Marjan Mernik. In such paper, ontologies are used during the initial phase of domain analysis in identifying common and variable elements of the domain that should be modeled in a language for that domain. Since the research is on its first steps, the analysis of applying ontologies in the other stages was not considered yet. Currently, ontologies are applied in the domain analysis and automated reasoning services have not been used. Indeed, reasoning services could be used to build a class diagram from the ontology. For example, the common subsumer [1] can be used to suggest an abstract super class based on the description of two or more concrete subclasses.

Daniel Okouya [5] presented a paper with the proposal of applying ontologies in conceptual modeling of multi-agent systems (MAS) and uses the expressive power of OWL based ontologies to deal with constraints verification and domain knowledge provision of MAS models. The idea is to support designers providing verification and validation of conceptual models produced during the MAS development process.

2.2 Integrated Approaches

Marion Murzek presented an infra-structure for integrating ontologies in MDE in the paper "Bringing Ontology Awareness into Model Driven Engineering Platforms" [7]. The architecture is based on the authors' experience with interoperability issues in metamodeling platforms. It should provide support to the following MDE disciplines: (1) modeling, (2) management and (3) guidance.

For example, the framework supports applying ontologies to validating models (1), simulations and model transformations (2) and Flexibility of process definitions(3). This is an ongoing research with first prototypes scheduled for the second semester of 2009.

An approach from a different point of view was presented by Guillaume Hillairet in the paper "MDE for publishing Data on the Semantic Web" [4]. It proposes the usage of the object model as pivot between persistence layer and ontology in semantic web applications. Mapping and transformations between the object model and an ontology are discussed. An interesting conclusion is that MDE helps to reduce the complexity of dealing with these mappings and transformations.

[1] www.most-project.eu

3. DISCUSSION

Many topics were discussed and still remain open issues. Firstly, the different objectives of applying ontologies in MDE demand attention. Among them, we point three: validation of conceptual models; specification with more expressiveness power; and information sharing. From these three applications, the first one presents the biggest amount of use cases until now. It happens maybe due to the facility of translating constructs of UML-based languages into Ontology Web Language (OWL). Moreover, since such translation is (or should be) an automatic step, developers do not have to learn ontology languages, which may be a hard task.

OWL has some characteristics that can be very useful in MDE. One of the mainly differences between ontologies and object-oriented paradigm is the notion of incomplete knowledge and the different ways of describing classes. Indeed, incomplete knowledge may be very useful in domains like medicine, where not all information about a disease or drug is known yet. However, some applications require complete information, like a airport timetable. The capability of describing class in many ways adds flexibility and can be useful to support domain analysis, as exemplified in [6].

3.1 Complexity of Ontologies

Ontology languages like OWL are logical languages and require different premisses, with influences the complexity level of such a language. For example, bridging of UML-like models and OWL-like ontologies invariably raises the questions about open world assumption (OWA) and closed world assumption (CWA). The question is whether the complexity of ontology languages like OWL and particularities like the open world assumption impair the adoption of such languages for software modeling. Some software modeling educators and practitioners claim that even OCL can be too complicated to be widely adopted. Is OWL even more complex?

To be able to answer those questions we still need to investigate how to evaluate the value added by ontology in MDE, as discussed in Sect. 3.4.

3.2 Need for integrated approaches

There is a visible research agenda attempting to integrate ontology technology (OT) into "standard" software development (SD) approaches. Such integration can be considered on very many different levels and corners of SD. Much of the technology attempted to be integrated have been developed in different communities and for different purposes (cf. UML vs. OWL).

In the context, the question is which integration style currently seems most rewarding (in the short run or the long run) from a language modeling point of view: tight or loose. The tight style involves strong integration of currently developed formalisms (languages and approaches). The loose style distinguish corners of SD where OT can be applied as stand-alone approaches, as black-boxes, hence being able to

reuse OT development and advances off-the-shelf. The latter seems to be more explored nowadays mainly due to the lack of integrated languages available and motivating use cases.

Integrating ontologies into existing formalisms and notions has been attempted before. Perhaps most notable in recent times is the attempt to integrate rules (Datalog, normal rules, Prolog rules etc.) with ontologies, resulting in the possibility to specify "hybrid programs". The objective has been to "bridge the best of two worlds" and exploit each approach for the other and thus achieve higher expressivity and more powerful formalisms. As for integrating ontologies (*a la* OWL) into system models (*a la* UML) today, the objectives seems to be very similar. When attempting to integrate ontologies into MDE today, an open question is what, if anything, we can learn from earlier attempts at ontology integration.

3.3 Visual Modeling of Ontologies

There are ways to deploy UML-profiles or similar techniques to achieve a visual syntax (arguably easier for end-users) for ontology development. UML-like visual syntax have been used to cover the expressivity of logic-based languages such as OWL. An open issue is whether using UML syntax for modeling OWL adds to the confusion of the two approaches. With a increasing focus on Domain Specific Languages (DSL), frameworks for developing visual languages have become popular and can be explored to develop alternatives for modeling OWL visually.

3.4 Assessment of Ontology Applications in MDE

Clearly, there are possible benefits of deploying ontology technology in the software engineering field. But software engineers are already required to have a good understanding of a legion of languages and techniques. Will, in the long run, the advances brought on by the application of ontology technology be worthwhile to deploy? Will the return of investment and added complication to the software engineering processes and methods by the addition of these new technologies result in higher quality software?

Thus, we need quality models of ontology driven applications. From the quantitative point of view, object oriented metrics could be used as starting point in the investigation of new quality metrics and quality models for integrated approaches.

4. CONCLUSION

The TWOMDE2008 was the first workshop at the MDE conference to address the application of ontologies in model driven development. The potential of this field has just started being explored.

Although we had papers covering different aspects of MDE, the employment of automated reasoning services to make use of the formal description provided by ontology languages has practically not been explored. Moreover, prominent topics in MDE like model transformation, traceability and query languages were not pondered by the papers of this first edition.

For the next editions, we expect more use cases in a wider range of topics. We would like to see successful industry use cases and mechanisms to evaluate the role of ontologies.

5. REFERENCES

[1] W. Cohen, A. Borgida, and H. Hirsh. Computing least common subsumers in description logics. In *In: Proc. AAAI-92, July 12-16, 1992, San Jose, California*, pages 754–760. AAAI Press/The MIT Press, 1992.

[2] A. Friesen. Potential applications of ontologies and reasoning for modeling and software engineering. In *Proceedings of the of the First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008) at MoDELS 2008, Toulouse, France, September 28, 2008*, volume 395 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[3] D. Harb, C. Bouhours, and H. Leblanc. Using an ontology to suggest design patterns integration. In *Proceedings of the of the First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008) at MoDELS 2008, Toulouse, France, September 28, 2008*, volume 395 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[4] G. Hillairet, F. Bertrand, and J.-Y. Lafaye. Mde for publishing data on the semantic web. In *Proceedings of the of the First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008) at MoDELS 2008, Toulouse, France, September 28, 2008*, volume 395 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[5] D. Okouya, L. Penserini, S. Saudrais, A. Staikopoulos, V. Dignum, and S. Clarke. Designing mas organisation through an integrated mda/ontology approach. In *Proceedings of the of the First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008) at MoDELS 2008, Toulouse, France, September 28, 2008*, volume 395 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[6] R. Tairas, M. Mernik, and J. Gray. Using ontologies in the domain analysis of domain-specific languages. In *Proceedings of the of the First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008) at MoDELS 2008, Toulouse, France, September 28, 2008*, volume 395 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[7] S. Zivkovic, M. Murzek, and H. Kuehn. Bringing ontology awareness into model driven engineering platforms. In *Proceedings of the of the First Workshop on Transforming and Weaving Ontologies in Model Driven Engineering (TWOMDE 2008) at MoDELS 2008, Toulouse, France, September 28, 2008*, volume 395 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

Fifth International Workshop on Data Management on New Hardware
DaMoN 2009

http://www.ins.cwi.nl/projects/**damon09**

Sponsored by and held in cooperation with SIGMOD/PODS 2009

Sunday, June 28, 2009
Providence, Rhode Island

Objective

The aim of this one-day workshop is to bring together researchers who are interested in optimizing database performance on modern computing infrastructure by designing new data management techniques and tools.

Topics of Interest

The continued evolution of computing hardware and infrastructure imposes new challenges and bottlenecks to program performance. As a result, traditional database architectures that focus solely on I/O optimization increasingly fail to utilize hardware resources efficiently. CPUs with superscalar out-of-order execution, many-core, simultaneous multi-threading, multi-level memory hierarchies, flash storage and other future storage hardware (such as PCM) impose a great challenge to optimizing database performance. Consequently, exploiting the characteristics of modern hardware has become an important topic of database systems research.

The goal is to make database systems adapt automatically to the sophisticated hardware characteristics, thus maximizing performance transparently to applications. To achieve this goal, the data management community needs interdisciplinary collaboration with computer architecture, compiler and operating systems researchers. This involves rethinking traditional data structures, query processing algorithms, and database software architectures to adapt to the advances in the underlying hardware infrastructure.

We seek submissions bridging the area of database systems to computer architecture, compilers, and operating systems. In particular, submissions covering topics from the following non-exclusive list are encouraged:

- database algorithms and data structures on modern hardware
- cost models and query optimization for novel hierarchical memory systems
- hardware systems for query processing
- data management using co-processors
- query processing using computing power in storage systems
- database architectures for low-power computing and embedded devices
- database architectures on multi-threaded and chip multiprocessors
- performance analysis of database workloads on modern hardware
- compiler and operating systems advances to improve database performance
- new benchmarks for microarchitectural evaluation of database workloads

Important Dates

- Paper submission: **April 17, 2009**
- Notification of acceptance: May 11, 2009
- Camera-ready copies: May 29, 2009
- Workshop: **June 28, 2009**

Workshop Co-Chairs

- Kenneth Ross (Columbia University)
- Peter Boncz (CWI)

Program Committee

- Anastasia Ailamaki (EPFL Lausanne & CMU)
- Qiong Luo (HKUST)
- Bishwaranjan Bhattacharjee (IBM Research)
- Stavros Harizopoulos (HP Labs)
- Bongki Moon (University of Arizona)
- Amr El Abbadi (UC Santa Barbara)
- Andreas Moshovos (University of Toronto)
- Naga Govindaraju (Microsoft)
- Keshav Pingali (Univ. Texas))

Call for Papers: DBTest 2009

2nd International Workshop on Testing Database Systems
Providence, June 29, 2009 (co-located with SIGMOD 2009)
http://dbtest2009.ethz.ch/

Motivation and Scope

The functionality provided by modern database management systems (DBMS), data storage services, and database applications is continuously expanding. New trends in hardware architectures, new data storage requirements, and new usage patterns drive the need for continuous innovation and expansion. As a result, these system/applications are becoming increasingly complex and difficult to validate. As a consequence, testing and tuning these system/applications is becoming increasingly expensive and are often dominating the release cycle. It is not unusual that fifty percent of the development cost is spent on testing and tuning and that several months are reserved for testing before a new release can be shipped.

The first workshop on testing database systems (collocated with SIGMOD 2008) has shown that there is a huge interest of the industry to discuss problems in the area of testing database systems together with the academic community. Moreover, testing has recently gained more attention in the database community with an increasing number of conference submissions as well as a special issue of the IEEE Data Engineering Bulletin in this area. The main purpose of this workshop is to continue the discussion between industry and academia in order to come up with a research agenda that describes important open problems in the area of testing database systems/applications. The long term goal is to devise new techniques which solve these problems in order to reduce the cost and time to test and tune database products so that users and vendors can spend more time and energy on actual innovations. Obviously, the software engineering community has already worked intensively on testing related problems. However, testing DBMS/database applications imposes particular challenges and opportunities which have not been addressed in either the database or software engineering community.

Important Dates

Paper Submission: April 3, 2009 (Friday, 5PM PST)
Notification of acceptance: May 8, 2009 (Friday)
Camera-ready: May 22, 2009 (Friday)
Workshop: June 29, 2009 (Monday)

Paper Submission

Papers should not be longer than six pages and should be submitted in PDF by E-Mail to: dbtest2009@inf.ethz.ch

Workshop Chairs

Carsten Binnig, ETH Zurich, Switzerland (carsten.binnig@inf.ethz.ch)
Benoit Dageville, Oracle Corporation, USA (benoit.dageville@oracle.com)

Call for Papers

3rd SIGMOD PhD Workshop on Innovative Database Research, IDAR 2009
Co-located with SIGMOD 2009, Providence, Rhode Island

Workshop website: http://wcms.inf.ed.ac.uk/idar09
Submission website: http://www.easychair.org/conferences/?conf=idar09

Aims of the workshop

The PhD Workshop on Innovative Database Research (to be held in conjunction with SIGMOD) is intended to bring together PhD students working on topics related to the SIGMOD conference series. The workshop will offer PhD students the opportunity to present, discuss, and receive feedback on their research in a constructive and international atmosphere. The workshop will be accompanied by established researchers in the field of database technology. These accompanying researchers will participate actively and contribute to the discussions. The workshop is co-located with and will take place right before the SIGMOD 2009 conference on June 28, 2009.

Workshop programme

The full-day workshop will consist of a 30-minute presentation and discussion for each accepted paper and two invited keynote speeches. Poster sessions during the coffee breaks are also likely depending on the number and quality of submissions.

Important dates

- Deadline for submission: April 10, 2009
- Notification to authors: May 11, 2009
- Camera ready due: May 29, 2009
- PhD Workshop: June 28, 2009

Topics of interest

As for the SIGMOD conferences series, all topics from the field of database technology are of interest for the PhD Workshop. These topics include (but are not limited to):

- Benchmarking and performance evaluation
- Data quality, semantics and integration
- Database monitoring and tuning
- Data privacy and security
- Data mining and OLAP
- Embedded, sensor and mobile databases
- Indexing, searching and database querying
- Managing uncertain and imprecise information
- Novel/ Advanced applications and systems
- Peer-to-peer and networked data management
- Personalized information systems
- Query processing and optimization
- Replication, caching, and publish-subscribe systems
- Semi-structured data
- Storage and transaction management
- Web services

Submission

Papers describing doctoral work should be submitted in PDF via the EasyChair submission website. The primary author of the paper should be a student, while single-authored papers are strongly encouraged. Papers should not exceed 6 pages; the format is according to the ACM SIG proceedings template. In contrast to regular conference papers, submissions should address specifically doctoral work! Therefore, the following elements are recommended:

- A clear formulation of the research question.
- An identification of the significant problems in the field of research.
- An outline of the current knowledge of the problem domain, as well as the state of existing solutions.
- A presentation of any preliminary ideas, the proposed approach and the results achieved so far
- A sketch of the applied research methodology.
- A description of the PhD project's contribution to the problem solution.
- A discussion of how the suggested solution is different, new, or better as compared to existing approaches to the problem.

The intention of this workshop is to support and inspire PhD students during their ongoing research efforts. Therefore, it is necessary that primary authors will have neither achieved their PhD degree nor submitted their thesis before the PhD workshop (June 28, 2009). To enforce this rule we require primary authors to disclose their expected graduation date and their advisor's name when submitting. Accepted papers will be published in the workshop proceedings, which will appear on the SIGMOD DISC.

Workshop organizers

Co-chairs

- Jaewoo Kang, Korea University, Korea
- Stratis D. Viglas, University of Edinburgh, UK

Steering Committee

- Tok Wang Ling, National University of Singapore, Singapore
- Xiaofeng Meng, Renmin University of China, China
- Ge Yu, Northeastern University, China

Program Committee

- Daniel Abadi, Yale University, USA
- Shivnath Babu, Duke University, USA
- Yon Dohn Chung, Korea University, Korea
- Irini Fundulaki, ICS-Forth, University of Crete, Greece
- Floris Geerts, University of Edinburgh, UK
- Anastasios Kementsietsidis, IBM Research, USA
- Rajasekar Krishnamurthy, IBM Research, USA
- Dongwon Lee, Pennsylvania State University, USA
- Kyriakos Mouratidis, Singapore Management University, Singapore
- Dan Olteanu, Oxford University, UK
- Sanghyun Park, Yonsei University, Korea
- Neoklis Polyzotis, University of California - Santa Cruz, USA
- Ravishankar Ramamurthy, Microsoft Research, USA
- Zografoula Vagena, Microsoft Research, UK
- Shuigeng Zhou, Fudan University, China

CALL FOR PAPERS

Sponsored by

MobiDE 2009: Eighth International ACM Workshop on Data Engineering for Wireless and Mobile Access
10 Year Anniversary

In-cooperation with

June 29, 2009, Providence, USA
(in conjunction with SIGMOD/PODS 2009)
http://www.cs.fsu.edu/mobide09/

Important Dates:

March 18:	Abstracts
March 25:	Papers/Demo
May 18:	Notification
June 1:	Camera Ready

(all deadlines are midnight EST)

This is the eighth of a successful series of workshops that aims to act as a bridge between the data management, wireless networking, and mobile computing communities.

The 1st MobiDE workshop took place in Seattle (August 1999), in conjunction with MobiCom 1999; the 2nd MobiDE workshop took place in Santa Barbara (May 2001), together with SIGMOD 2001; the 3rd MobiDE workshop took place in San Diego (September 2003), together with MobiCom 2003; the 4th MobiDE workshop was held in Baltimore (June 2005. In 2006, MobiDE was organized in Chicago (June 2006). The 6th MobiDE was held in Beijing, China (June 2007). Last year's event took place in Vancouver, Canada (June 2008). This year's event marks the 10-year anniversary of the workshop. As in the past 10 years, the workshop will continue to serve as a forum for researchers and technologists to discuss the state-of-the-art, present their contributions, and set future directions in data management for mobile and wireless access.

The topics of interest related to mobile and wireless data engineering include, but are not limited to:

* ad-hoc networked databases
* consistency maintenance and management
* context-aware data access and query processing
* data caching, replication and view materialization
* data publication modes: push, broadcast, and multicast
* data server models and architectures
* database issues for moving objects: storing, indexing, etc.
* m-commerce
* mobile agent models and languages
* mobility-aware data mining and warehousing
* mobile database security
* mobile databases in scientific, medical and engineering applications
* mobile peer-to-peer applications and services
* mobile transaction models and management
* mobile web services
* mobility awareness and adaptability
* pervasive computing
* prototype design of mobile databases
* quality of service for mobile databases
* sensor network databases
* transaction migration, recovery and commit processing
* wireless multimedia systems
* wireless web

General Chairs:

Yannis Kotidis
Athens University of Economics and Business
kotidis@aueb.gr

Pedro Jose Marron
University of Bonn
pjmarron@cs.uni-bonn.de

Program Chairs:

Le Gruenwald
University of Oklahoma
ggruenwald@ou.edu

Demetris Zeinalipour
University of Cyprus
dzeina@cs.ucy.ac.cy

Publicity Chair:

Feifei Li
Florida State University
lifeifei@cs.fsu.edu

Demonstration Chair:

Zografoula Vagena
Microsoft Research Cambridge
zografv@microsfot.com

12th International Workshop on the Web and Databases (WebDB 2009)

Providence, Rhode Island - June 28, 2009

Co-located with ACM SIGMOD 2009

Workshop Chairs

Alexandros Labrinidis
University of Pittsburgh, USA
Michalis Petropoulos
SUNY Buffalo, USA

Program Committee

Karl Aberer EPFL, Switzerland **Sihem Amer-Yahia** Yahoo!, US **Andrey Balmin** IBM Almaden, US **Denilson Barbosa** University of Alberta, Canada **Michael Benedikt** University of Oxford, UK **José A. Blakeley** Microsoft, US **Michael Carey** UC Irvine, US **Kevin Chen-Chuan Chang** UIUC, US; **Vassilis Christophides** ICS-FORTH, Greece **Alin Deutsch** UC San Diego, US **AnHai Doan** University of Wisconsin-Madison, US **Peter Dolog** Aalborg University, Denmark **Luna Dong** AT&T Labs, US **Schahram Dustdar** TU Vienna, Austria **Piero Fraternalli** Politecnico di Milano, Italy **Juliana Freire** University of Utah, US **Luis Gravano** Columbia, US **Vagelis Hristidis** Florida International University, US **Rick Hull** IBM T.J. Watson, US **Bertram Ludaescher** UC Davis, US **Qiong Luo** HKUST, Hong Kong **Jayant Madhavan** Google, US **Amelie Marian** Rutgers University, US **Paolo Merialdo** Universita di Roma Tre, Italy **Gerome Miklau** UMass-Amherst, US **Chris Olston** Yahoo!, US **Yannis Papakonstantinou** app2you.com, US **Jayavel Shanmugasundaram** Yahoo!, US **Wang-Chiew Tan** UC Santa Cruz, US **Vasilis Vassalos** AUEB, Greece **Yannis Velegrakis** University of Trento, Italy **Jeffrey Yu** Chinese University of Hong Kong, Hong Kong

Web Chair

Demian Lessa
SUNY Buffalo, USA

Call For Papers

WebDB Goals

The WebDB workshop focuses on providing a forum where researchers, theoreticians, and practitioners can share their knowledge and opinions about problems and solutions at the intersection of data management and the Web.

Important Dates

Submission deadline: Wednesday, April 22, 2009 (11:59pm EST)
Notification: Tuesday, May 26, 2009
Camera-Ready Due: Monday, June 8th, 2009
Workshop: Sunday, June 28, 2009

Topics of Interest

This year WebDB will focus on **Database as a Web Service** and on **Mobile Web**, but papers on all aspects of the Web and Databases are solicited. Topics of interest include (but are not limited to):

Business processes for applications on the Web; Data integration over the Web; Data Mashup frameworks; Data-oriented aspects of Web application development environments; Data Services; Data-intensive applications on the Web; Database Support for social Web 2.0 applications; Information retrieval in semistructured data and the Web; Location-aware Web applications; Methodologies and tools for Web data publishing; Pay-As-You-Go Data Integration; Publish/subscribe systems; Query languages and systems for XML and Web data; Semistructured data management; The Semantic Web; Web Community Data Management Systems; Web Information Extraction; Web privacy and security; Web services and distributed computing over the Web; Web-based alerting systems; Web-based distributed data management.

Submission Instructions

Authors are invited to submit original, unpublished **research papers** that are not being considered for publication in any other forum. Papers submitted should be six pages long. Besides regular paper submissions, WebDB 2009 also welcomes the submission of software **demonstration proposals**, to foster interaction on hot topics and ongoing work. Demo proposals should be two pages long. Demonstration proposals should outline the context and highlights of the software to be presented, and briefly describe the demo scenario. We encourage the joint submission of research papers describing new concepts and fundamental results and of demo proposals of software developed based on those new concepts. Electronic versions of the papers and demos will be included in the ACM Digital Library.

http://webdb09.cse.buffalo.edu

Gramado
Brazil

28th International Conference on Conceptual Modeling

CALL FOR PAPERS

CONFERENCE ORGANIZATION

Conference General Chair
José Palazzo M. de Oliveira, UFRGS, BRA

Program Committee Co-Chairs
Alberto H. F. Laender, UFMG, BRA
Silvana Castano, UNIMI, ITA
Umesh Dayal, HP Labs, USA

Steering Committee Liaison
Arne Sølvberg, NTNU, NOR

Workshop Chairs
Carlos A. Heuser, UFRGS, BRA
Günther Pernul, U. Regensburg, GER

Tutorial Chairs
Daniel Schwabe, PUC-RJ, BRA
Stephen W. Liddle, BYU, USA

Panel Chair
David Embley, BYU, USA

Industrial Chair
Fabio Casati, U Trento, ITA

Demos and Posters Chairs
Altigran S. da Silva, UFAM, BRA
Juan-Carlos Mondéjar, U. Alicante, ESN

PhD Colloquium Chairs
Stefano Spaccapietra, EPFL, SWI
Giancarlo Guizzardi, UFES, BRA

Local Arrangements Chair
José Valdeni de Lima, UFRGS, BRA

IMPORTANT DATES

16.Mar.09 - Paper abstract submission
30.Mar.09 - Full paper submission
01.Jun.09 - Notification
22.Jun.09 - Camera-ready submission
29.Jun.09 - Tutorial proposals
29.Jun.09 - Panel proposals
29.Jun.09 - Poster & demo proposals

SUBMISSION GUIDELINES

Proceedings will be published by Springer in the LNCS series. Authors must submit manuscripts with up to 14 pages using the LNCS style. Authors are asked to submit an abstract, and then to upload the full paper. All submissions will be electronically only.

MORE INFORMATION AT THE WEBSITE:
http://www.inf.ufrgs.br/ER2009

The International Conference on Conceptual Modeling is a leading forum for presenting and discussing current research and applications in which the major emphasis is on conceptual modeling. ER 2009 will be held at the beautiful city of Gramado, a small touristic town in the Brazilian south. We solicit submission of original research, as well as experience and vision papers from both researchers and practitioners. Examples of topics of interest include, but are not limited to, conceptual modeling as applied to:

- Information Modeling Concepts, including Ontologies;
- Ontological and Conceptual Correctness in Modeling;
- Logical Foundations of Conceptual Modeling;
- Web Information Systems;
- Mobile information systems and pervasive computing;
- Service-Oriented Computing and Enterprise Architecture;
- The Semantic Web;
- Semistructured Data and XML;
- Information and Database Integration;
- Information Retrieval, Organization, Summarization, and Visualization;
- Design Methodologies and their Evaluation;
- Software Engineering and Tools;
- Requirements Engineering;
- Reuse, Patterns, and Object-Oriented Design;
- Reverse Engineering and Reengineering;
- Quality and Metrics;
- Empirical Studies of Conceptual Modeling;
- Conceptual Change and Schema Evolution;
- Maintenance of Information Systems;
- Management of Integrity Constraints;
- Active Concepts in Conceptual Modeling;
- Spatial, Temporal, Multimedia, and Multi-chanel Aspects;
- Metadata, its Interpretation and Usage;
- User Interface Modeling;
- Knowledge Management Systems;
- Groupware and Workflow Management;
- Data warehousing, data mining, and business intelligence;
- E-Learning, E-Business and E-Government; and
- Other Advanced and Cross-Disciplinary Applications.

WORKSHOPS

- ACM-L: Active Conceptual Modeling of Learning
- Conceptual Modeling in the Large
- ETheCoM: Evolving Theories of Conceptual Modelling
- FP-UML: Workshop on Foundations and Practices of UML
- LbM: Logic Based Modeling
- M2AS'09: Intl. Work. on Modeling Mobile Applications and Services
- MOST-ONISW: Joint Intl. Work. on Metamodels, Ontologies, Semantic Technologies, and Information Systems for the Semantic Web
- QoIS: Quality of Information Systems
- RIGiM: Requirements, Intentions and Goals in Conceptual Modeling
- SeCoGIS 2009: Semantic and Conceptual Issues in Geographic Information Systems

WSDM 2009
Call for Participation

Second ACM International Conference on Web Search and Data Mining
February 9-12, 2009, Barcelona, Spain
http://wsdm2009.org/

Co-Sponsored by ACM SIGIR, SIGKDD, SIGMOD and SIGWEB
Early registration deadline: 7th January 2009

WSDM (pronounced "wisdom") is a young ACM conference intended to be **the publication venue** for research in the areas of search and data mining. Indeed, the pace of innovation in these areas prevents proper coverage by conferences of broader scope. The high attendance at the first WSDM, held at Stanford University in February of 2008, is a confirmation of this trend.

This year the conference will be held in Barcelona, 9-12 February, and it will feature an exciting technical program. This year we accepted 29 papers out of 170 submissions (17% acceptance rate). Besides the technical talks of the accepted papers (see the list on next page), the conference will feature three invited keynote talks by:

- **Gerhard Weikum** (Research Director at the Max-Planck Institute for Informatics, Germany),
- **Jeffrey Dean** (Google Fellow),
- **Ravi Kumar** (Yahoo! Research)

The conference will also host three workshops:

- The Sixth Workshop on Algorithms and Models for the Web-Graph (WAW 2009)
- Workshop on Exploiting Entities for Information Retrieval (ESAIR II)
- Workshop on Web Search Click Data (WSCD)

We hope to see you in Barcelona next February!

Conference Chair: Ricardo Baeza-Yates, Yahoo! Research

Program Committee Co-Chairs: Paolo Boldi, Universita di Milano
Berthier Ribeiro-Neto, Google

Steering Committee:

Rakesh Agrawal (Microsoft)
Ricardo Baeza-Yates (Yahoo! Research)
Ziv Bar-Yossef (Technion, Google)
Soumen Chakrabarti (IIT Bombay)
Monika Henzinger (EPFL, Google)
Jon Kleinberg (Cornell)
Rajeev Motwani (Stanford)
Prabhakar Raghavan (Yahoo!)

Accepted Papers

* Benjamin Piwowarski, Georges Dupret and Rosie Jones. *Mining User Web Search Activity with Layered Bayesian Networks or How to Capture a Click in its Context.*

* Simon Overell, Borkur Sigurbjornsson and Roelof van Zwol. *Classifying Tags using Open Content Resources*

* Fan Guo, Chao Liu and Yi-Min Wang. *Efficient Multiple-Click Models in Web Search*

* Xuerui Wang, Andrei Broder, Evgeniy Gabrilovich, Vanja Josifovski and Bo Pang. *Robust Cross-Language Query Classification with External Web Evidence*

* Xiang Wang, Kai Zhang, Xiaoming Jin, Dou Shen. *Mining Common Topics from Multiple Asynchronous Text Streams*

* Ryen White, Susan Dumais, Jaime Teevan. *Characterizing the Influence of Domain Expertise on Web Search Behavior*

* Maggy Anastasia Suryanto, Ee-Peng Lim, Aixin Sun, Roger Chiang. *Quality-Aware Collaborative Question Answering: Methods and Evaluation*

* Adish Singla and Ingmar Weber. *Camera Brand Congruence in the Flickr Social Graph*

* Hongbo Deng, Irwin King, Michael Lyu. *Effective Latent Space Graph-based Re-ranking Model with Global Consistency*

* Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson and Samuel Ieong. *Diversifying Search Results*

* Ravi Kumar, Kunal Punera, Torsten Suel, Sergei Vassilvitskii. *Top-k Aggregation Using Intersection of Ranked Inputs*

* Jaime Teevan, Meredith Ringel Morris and Steve Bush. *Discovering and Using Groups to Improve Personalized Search*

* Michael Bendersky and Bruce Croft. *Finding Text Reuse on the Web*

* Kazuhiro Seki and Kuniaki Uehara. *Adaptive Subjective Triggers for Opinionated Document Retrieval*

* Daniel Ramage, Paul Heymann, Christopher Manning and Hector Garcia-Molina. *Clustering the Tagged Web*

* Yin Yang, Nilesh Bansal, Wisam Dakka, Panagiotis Ipeirotis, Nick Koudas and Dimitris Papadias,. *Query by Document*

* Chinmay Karande, Kumar Chellapilla and Reid Andersen. *Speeding up Algorithms on Compressed Web Graphs*

* Marijn Koolen, Gabriella Kazai and Nick Craswell. *Wikipedia Pages as Entry Points for Book Search*

* Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka. *Measuring the Similarity between Implicit Semantic Relations using Web Search Engines*

* Ling Chen, Phillip Wright and Wolfgang Nejdl. *Improving Music Genre Classification Using Collaborative Tagging Data*

* Alvaro Pereira, Ricardo Baeza-Yates, Nivio Ziviani and Jesus Bisbal. *A Model for Fast Web Mining Prototyping*

* Eytan Adar, Jaime Teevan, Susan Dumais and Jonathan Elsas. *The Web Changes Everything: Understanding the Dynamics of Web Content*

* Rakesh Agrawal, Alan Halverson, Krishnaram Kenthapadi, Nina Mishra, P. Tsaparas. *Generating Labels from Clicks*

* Tapas Kanungo. *Predicting Readability of Short Web Summaries*

* Fernando Diaz. *Aggregation of News Content Into Web Results*

* Eytan Adar, Michael Skinner and Daniel Weld. *Information Arbitrage in Multi-Lingual Wikipedia*

* Songhua Xu and Francis Lau. *A New Visual Interface for Search Engines*

* Jaap Kamps and Marijn Koolen. *Is Wikipedia Link Structure Different?*

* Marc Najork, Sreenivas Gollapudi and Rina Panigrahy. *Less is More: Sampling the Neighborhood Graph Makes SALSA Better and Faster*