

Enterprise Information Extraction: Recent Developments and Open Challenges

Laura Chiticariu Yunyao Li Sriram Raghavan Frederick R. Reiss
IBM Research – Almaden
San Jose, CA, USA
{chiti, yunyaoli, rsriram, ffreiss}@us.ibm.com

ABSTRACT

Information extraction (IE) — the problem of extracting structured information from unstructured text — has become an increasingly important topic in recent years. A SIGMOD 2006 tutorial [3] outlined challenges and opportunities for the database community to advance the state of the art in information extraction, and posed the following grand challenge: “*Can we build a System R for information extraction?*”

Our tutorial gives an overview of progress the database community has made towards meeting this challenge. In particular, we start by discussing *design requirements* in building an enterprise IE system. We then survey recent technological advances towards addressing these requirements, broadly categorized as: (1) *Languages* for specifying extraction programs in a declarative way, thus allowing database-style performance optimizations; (2) *Infrastructure* needed to ensure scalability, and (3) *Development support* for enterprise IE systems. Finally, we outline several *open challenges* and *opportunities* for the database community to further advance the state of the art in enterprise IE systems. The tutorial is intended for students and researchers interested in information extraction and its applications, and assumes no prior knowledge of the area.

Categories and Subject Descriptors

H.0 [Information Systems]: General

General Terms

Design

Keywords

accuracy, declarative, enterprise applications, information extraction, scalability, usability

1. INTRODUCTION

Today’s enterprises generate and consume rapidly increasing quantities of unstructured text data: emails, webpages, news articles, blog posts, online reviews and comments, call center records, and more. This unstructured information is becoming a more and more vital part of running an enterprise, both for daily operations and long-term strategic analysis. While such documents often contain valuable information (e.g., contact information, financial events, opinions about products), they are of limited use if the information cannot be processed automatically by applications. As a consequence, information extraction (IE) — the problem of extracting

structured information from unstructured text — has gained significant traction in several communities: natural language processing, information retrieval, and recently, databases.

Information extraction has been a central topic of interest in the Natural Language Processing (NLP) community, where the emphasis has been on recognizing entities and relationships between such entities from formal documents of reasonable size (e.g., news articles). In recent years, however, the nature of information extraction tasks (e.g., extract reviews from blogs, financial information from regulatory filings) and new document types (e.g., informal, and dirty text in blogs, individual documents of several megabytes in size) required by enterprise applications, has radically changed. It became increasingly clear that fundamental limitations in expressivity and scalability of conventional NLP approaches prevent them from handling such tasks gracefully.

In their earlier SIGMOD 2006 tutorial, Doan et al. [3] challenged the database community to “*build a System R for Information Extraction*” — that is, to build an information extraction system that meets the practical needs of real-world enterprise applications. Recent work in the database community has largely followed this vision, focusing on the needs of several important applications. Prominent examples of these applications include:

- *Semantic Search*: Extract structured information from text documents, and use this information as metadata to enhance the accuracy of keyword searches.
- *Data as a Service*: Extract and clean useful information hidden in publicly available documents, creating a valuable collection of structured data that can be rented or shared over the Internet.
- *Business Intelligence*: Mine text data streams such as blog entries or call center records for information about consumer sentiment, product pipeline problems, or important economic events.
- *Data-driven Mashups*: Extract structured information from unstructured feeds and join this structured information with other enterprise data to build new data mashups.

Efficient automatic extraction of structured data from unstructured data is the key enabler of such enterprise-level data-intensive applications. We identify a set of requirements that these types of applications give rise to, that we call *Enterprise Information Extraction*, or *Enterprise IE*. Broadly, these requirements are:

- *Scalability*. Enterprise applications operate over vast amounts of text data, often orders of magnitude larger than the quantities used in classical IE work. An enterprise IE system should be able to operate at these scales without compromising its efficiency and memory consumption.
- *Accuracy*. The value of an enterprise information extraction application lies in its ability to produce extremely accurate, re-

liable results. Anything less undermines the usefulness of the application.

- *Usability*. Building an accurate IE system is an inherently labor-intensive process. Therefore, the usability of an enterprise IE system in terms of ease of development and maintenance is an important requirement.

In this tutorial we discuss recent technological advances made by the database community towards addressing the enterprise IE requirements outlined above.

2. TUTORIAL OUTLINE

2.1 Part 1: Declarative Approaches

Traditionally, IE systems have been built from individual extraction components consisting of rules or machine learning models. These individual components are then connected procedurally in a programming language like C++, Perl or Java.

Recent work in the database community has developed an alternative *declarative* approach to information extraction. Instead of using procedural logic to implement the extraction task, declarative IE systems separate the description of *what* to extract from *how* to extract it, allowing the rule developer to build complex extraction programs without worrying about performance considerations. We discuss notable examples of declarative IE languages developed since 2006, including AQL/SystemT [6], PSOX [1], SQoUT [4], xLog [2], and RAD [5]. As we show in the rest of the tutorial, the declarative approach enables significant improvements in both scalability and accuracy, while making it possible to build new classes of IE tools.

2.2 Part 2: Scalable Infrastructure

Scalability, both in terms of the amount of data and the complexity of the rule set, is an important challenge for building an enterprise information extraction system. The declarative approach has demonstrated important scalability benefits because it separates the semantics of a set of rules from their implementation. This separation permits the use of a “database-style” system design, with a *rule optimizer* that generates efficient execution plans and an *algebraic runtime* that implements these plans by tying together various pre-built operators. While may seem quite natural to a database researcher, this architecture represents a serious departure from previous work in information extraction and enables significant performance improvements.

Recently, several IE systems have been developed using an optimizer/runtime design. We discuss how these systems have taken different approaches within the context of this overall design, and we identify three dimensions that distinguish these approaches, both from each other and from traditional database systems: query rewriting vs. cost-based optimization, fine-grained vs. coarse-grained algebraic operators, and document (local) vs. collection (global) runtime.

2.3 Part 3: Development Support

Usability is an important factor in data management systems. In the context of enterprise IE systems, usability is even more important for several reasons: extraction programs are orders of magnitude larger and more complex compared to (semi-)structured queries, the extracted results are not exact, and the extraction rules change over time (e.g., when the system is adapted to a different domain).

A declarative approach to IE enables development support that was simply unthinkable of in the context of approaches developed in the NLP community, where the extraction specification is either a

black-box (in statistical approaches) or too procedural (in grammar-based approaches). We survey several advancements in building development support, including novel notions of data provenance proposed for explaining false negatives (i.e., missing answers) and methods for (semi-)automatically refining extraction programs based on labeled data and user feedback.

2.4 Part 4: Open Challenges

There are two major open challenges for future research on enterprise IE:

Optimization. While applying classical query optimization techniques to the IE domain, recent work has also exposed several new problems that are specific to information extraction. There is work to be done in developing better cost and selectivity models for low-level extraction primitives like regular expressions, as well as for complex “black box” extraction modules. Examples of other challenges include building more flexible parametric execution plans for heterogeneous document collections and detecting common subexpressions in IE rules.

Better Development Support. Even with recent advances discussed earlier, the development and maintenance of an enterprise IE system can still be a labor-intensive process. Debugging tools that trace the source of errors and provide suggestions on how to fix the errors are in great demand. Tools that enable the incorporation of community user feedback would greatly ease the burden of maintenance. Tools that enable existing IE systems to be seamlessly adapted and customized for new domains or applications can be of great value in making enterprise IE systems reusable.

About the presenters:

Laura Chiticariu is a Research Staff Member at IBM Research – Almaden. She received her Ph.D from U.C. Santa Cruz in 2008. Her current research focuses on using provenance to improve developmental support in information extraction systems.

Yunyao Li is a Research Staff Member at the IBM Research – Almaden. She received her Ph.D from the University of Michigan, Ann Arbor in 2007. Her research focuses on improving information accessibility for enterprise applications.

Sriram Raghavan manages the Search and Analytics Group at IBM Research – Almaden. He has been involved in developing and building enterprise search and information extraction applications for the past 6 years, since graduating from Stanford University in 2004.

Frederick Reiss is a Research Staff Member at the IBM Research – Almaden. He received his Ph.D. from U.C. Berkeley in 2006. His research focuses on improving the scalability of text analytics in enterprise applications.

3. REFERENCES

- [1] P. Bohannon et al. Purple SOX Extraction Management System. *SIGMOD Record*, 37(4):21–27, 2008.
- [2] A. Doan et al. Information extraction challenges in managing unstructured data. *SIGMOD Record*, 37(4):14–20, 2008.
- [3] A. Doan, R. Ramakrishnan, and S. Vaithyanathan. Managing Information Extraction: State of the Art and Research Directions. In *SIGMOD*, pages 799–800, 2006.
- [4] A. Jain et al. Building Query Optimizers for Information Extraction: the SQoUT Project. *SIGMOD Record*, 37(4):28–34, 2008.
- [5] S. Khaitan et al. RAD: A Scalable Framework for Annotator Development. In *ICDE*, 2008.
- [6] R. Krishnamurthy et al. SystemT: a System for Declarative Information Extraction. *SIGMOD Record*, 37(4):7–13, 2008.