# 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.