Information Extraction for Effective Knowledge Management

This paper highlights some of the major challenges in Information Extraction (IE) for effective knowledge management and discusses different scenarios in which the information can be extracted from a given set of documents.

It is well-known that much knowledge in any organization resides in text documents of various categories. Managing and effectively using the document repository is difficult, manual and effort-intensive task. Information Extraction is a process of extracting information from documents containing unstructured natural language text and presenting it in a structured format. This structured information can then be effectively searched, disseminated, reused or mined using data mining techniques to discover valuable knowledge, patterns and insights. Clearly, effective extraction and usage of the knowledge in textual documents holds a key to effective knowledge management.

IE plays a critical role in several practical applications ranging from resumes processing to insurance claim management. In this document, we review the basics of IE technology.

General Terms
Management, Applications, Structured information

Keywords
Information Extraction, Knowledge Management
About the Authors

Girish Keshav Palshikar

Girish Keshav Palshikar is a Principal Scientist and leads the Machine Learning R&D group in Systems Research Lab (SRL), an innovation centre in Tata Consultancy Service Limited, Pune, India. His research interests are in machine learning, data mining, text mining and theoretical computer science. He has more than 60 publications in international journals and conferences.

Rajiv Srivastava

Rajiv Srivastava leads the iWork program in SRL aimed at building tools and systems related to HR analytics. Rajiv is building ResumeCenter, a single-window system to deliver information and insights from processing and analysis of resume repositories. Many parts of ResumeCenter are already functional and deployed for various users in TCS HR. Rajiv also leads the development and deployment of INX (generic IE tool) and RINX (specialized IE tool for resumes).

Shailesh Deshpande

Shailesh Deshpande is a member of the Machine Learning R&D group in Systems Research Lab (SRL). Shailesh has developed a tool called Quest for analyzing and mining survey responses, both structured and unstructured. Quest is in active use in TCS for analyzing PULSE the annual employee satisfaction survey, besides many other surveys. Shailesh has also worked on INX and conducted research on several text mining tasks such as document summarization.

Sachin Pawar

Sachin Pawar is a member of the Machine Learning R&D group in Systems Research Lab (SRL). Sachin has worked on INX and RINX and on the unsupervised gazette creation algorithm in particular.

Savita Bhat

Savita Bhat is a member of the Machine Learning R&D group in Systems Research Lab (SRL). Savita has worked on INX and RINX, as well as their applications. She is working on the use of ranking algorithms in RINX.
Table of Contents

1. Introduction 4
2. Information Extraction (IE) 5
3. Challenges 6
4. Information Extraction Technology 7
5. INX Toolset for IE 8
6. IE Methodology 10
7. Public-Domain and Commercial IE Tools 12
Introduction

With a continuously changing business scenario, IT is viewed as a solution enabler and a tool to create differentiated services. Knowledge Management (KM) comprises a range of processes used in an organization to extract and use the information buried in document repositories. Most organizations have dedicated teams for their internal KM strategy. Clearly, effective extraction and usage of the knowledge in textual documents holds a key to effective KM.

It is well-known that much knowledge in any organization resides in textual documents of various kinds: reports, proposals, requirements, designs, experiment results, plans, manuals, emails, blogs, presentations, spreadsheets, resumes and so on. For performing various tasks, it is often important to locate all the related documents, extract the relevant information from them, consolidate this information into a coherent picture and then use it. The quality of the work is often dependent on whether or not the person has made an effective use of the knowledge existing in these document repositories.

Wherever a large repository of unstructured text documents is present, opportunities abound for information extraction from them. A few concrete practical applications are as follows.

- To identify a team, the Project Leader needs to identify the most relevant people based on the descriptions of education, projects, tools, skills and other experience mentioned in their resumes.
- While preparing a proposal, the Business Development Manager needs to extract details of the relevant projects and tools from profiles (summaries) of past projects.
- To identify best maintenance practices, a Support Manager needs to extract problem symptoms, diagnosis and repair actions from maintenance records.
- For better competitive advantage, the CEO of a company needs to extract relevant financial events (M&A, projects won, new products, technology alliances, new offices, financial results, people movement) of the competitors from a large feed of news items on the Web.
- For effective understanding of customer needs, a Marketing Manager would extract and classify the major complaints, comments, suggestions and sentiments about his products on blogs in the Web.
- To improve employee satisfaction levels, an HR manager, would like to extract major complaints and specific suggestions from an employee survey.
- To help in detecting overpriced or fraudulent insurance claims, Insurance Companies would like to extract and analyze accident, damage, repair, injury, and treatment details from insurance claims.

Most documents contain unstructured free-form text in a natural language (for example, English). Repositories contain large number of documents whose contents is organized in complex structures, which varies from document to documents. That makes managing and effectively using the document repository a difficult, manual and effort-intensive task.
Several text mining technologies are used as part of an effective KM process for managing document repositories; examples: document summarization, document categorization, document clustering, information extraction, topic discovery, sentiment analysis and so on. In this paper, we focus on Information Extraction, which is the most important initial step. The paper is organized into seven sections—Section 2 elaborates on Information Extraction; Section 3 discusses the challenges in Information Extraction. Section 4 is devoted to the Information extraction technology; sections 5 presents the INX Toolset for IE, Section 6 explain the standard IE process. Section 7 discusses the tools available both in public-domain and commercially.

**Information Extraction**

Information Extraction (IE) is a process of extracting information from documents containing unstructured natural language text and presenting it in a structured format. The extracted information is output to structured data repositories such as database tables or XML files. This structured information can then be effectively searched, disseminated, reused or mined (using data mining techniques) to discover valuable knowledge, patterns and insights. The extraction process is often - but not always - guided by means of extraction patterns (Figure. 1).

![Figure 1: The idea of Information Extraction](image)

Typically, the information to be extracted from a text document consists of the following types:

(a) **Generic named entities:** Names of people, places, organizations, dates, times, emails, URLs, phone numbers, addresses, amounts and so on.

(b) **Domain-specific named entities:** Names of organisms, diseases, drugs, enzymes, proteins from biological, medical and health related documents; names of engine components in mechanical equipment maintenance and so on.
(c) **Taxonomic relations**: Standard relations between entities such as IS_KIND_OF or IS_PART_OF, which are not explicitly stated; for example, room IS_PART_OF hotel, furnishing IS_PART_OF room, apple IS_KIND_OF fruit.

(d) **Domain-specific relations**: ORG acquires ORG (IBM acquired ABC Inc.), ternary relation between PERSON, POST and ORGANISATION (Mr. Smith is a Vice President in IBM) and so on.

**Challenges**

Following factors make IE a challenging task:

- Spelling errors for example, crshed and grammatical errors, for example, problem occur in database.
- Use of abbreviations, for example, IBM.
- Incorrect capitalization, for example, could not Print From explorer and punctuation errors, for example, missing full stop.
- Incomplete sentences (text fragments); for example, Replaced main crankshaft.
- Domain-specific vocabulary for example, Windows and SQL Server crashed due to a bug in A31.DLL.
- Many different ways of saying the same thing (repair, fix, mend).
- Information hidden due to use of pronouns (anaphora); for example, He repaired the faulty printer.

Deep statistical and language processing techniques are needed to perform effective IE. In addition, IE from non-English and multi-lingual documents poses its own challenges.
There are three different ways in which information can be extracted from a given set of documents.

1. Manual, Pattern-based IE

The crux of the IE technology is certainly the power and ease of use of the way the extraction patterns are defined. Typically, the extraction patterns are defined using a formalism called regular expressions (RE). One can define a set of RE patterns for an entity. These patterns can be easily matched directly with the given input text and the matched text is extracted, which corresponds to an occurrence of that entity. For example, suppose our IE task is to extract from news item, the financial events that corresponds to award of contracts or projects, by one organization to another. We need to identify and extract occurrences of an entity called Organization (ORG). We can begin with a simple RE pattern to recognize an ORG entity as follows:

A sequence of capitalized words is an ORG if it contains any one of cue words such as Inc., Co., Company, Limited or Bank.

But there are few obvious limitations of such a pattern. For example, some organizations may contain other cue words (for example, Incorporated), some organizations may contain non-capitalized words (for example, Royal Bank of Scotland) and some may not contain any of the known cue word at all (for example, Infosys). While we can keep enhancing our RE pattern to handle some of these special cases, clearly such lexical patterns are inherently brittle. Humans recognize occurrence of an organization in a sentence using patterns that are much deeper. Basically, these patterns are based on the fact that organizations are used in certain specific ways in a sentence. These patterns may be syntactic or semantic; or they may even use common-sense knowledge about the world (for example, an organization can rarely cry or fly). We shall later discuss one specific way to enrich the text representation so that richer patterns can be defined.

Despite the apparent limitations of the pattern-based approach, it is widely used in practice because it enables the IE engineer to incorporate special cases and domain knowledge. Many IE tools come with a library of pre-coded RE patterns for simpler named entities - such as telephone numbers, email addresses, URLs, amounts and so on - so that these entities can be extracted from the end-users without any effort. Many IE tools also offer built-in functionality to detect generic named entities such as PERSON, LOCATION or ORG. However, making effective use of these facilities is still far from simpler and may need the end-user to write ‘higher level’ patterns. For example, one may have to write additional patterns to classify

---

**Information Extraction (IE) Technology**

Satyam announced it has bagged a five-year $200 million (about Rs 900 Crore) maintenance-cum-business transformation deal from Applied Materials Inc.

<table>
<thead>
<tr>
<th>To_Company</th>
<th>From_Company</th>
<th>Value</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satyam</td>
<td>Applied Materials Inc.</td>
<td>USD 200000000</td>
<td>5 years</td>
</tr>
</tbody>
</table>

*Figure 3 : Information extraction of financial events from a news story*
PERSON names into different categories; for example, police officers, victims, witnesses or accused in documents related to crime or into directors, actors, screen-play writers and singers in film-related documents.

2. Gazette-based IE
A rather simple approach to IE is to make use of a pre-defined list, called a **gazette** or **gazetteer**, of all possible values of a named entity. Whenever any of these examples occur in a document, they are extracted. For example, one can prepare a list of all academic degrees – Bachelor of Arts, Master of Technology, Doctor of Philosophy or Diploma in Mechanical Engineering – along with their acronyms (B.A., M.Tech., Ph.D., D.M.E.), their possible variations, for example, BA without the periods and abbreviations, for example, Dip. Mech. Engg.. Then it is much simpler to search the given document to find whether any of these degrees are present in it and extract them if present. Of course, gazettes can be prepared only for those named entities which have a finite number of possible values for example UNIVERSITY, PROGRAMMING_LANGUAGE and so on., and not for entities such as PERSON, ORG or LOCATION which have potentially unlimited number of values. While gazette-based IE is fast and accurate, the crux of the matter is in preparing the gazette, which is as complete and accurate as possible.

A number of machine learning algorithms have been devised for automatically constructing such gazettes, starting with a few seed examples and an unlabeled corpus of documents.

3. Machine Learning for IE
Another powerful approach is based on the machine learning algorithms which automatically learn the IE patterns by generalizing from a given set of examples. The user first creates a training dataset, which is a collection of documents in which all occurrences of the named entity of interest are manually marked or tagged. Many supervised machine learning algorithms called **Classification algorithms** are available which process these training examples and learn a set of rules to identify the occurrences of the named entity. To come up with these set of rules, the machine learning algorithms basically take into account **features**, such as words surrounding an occurrence of the named entity. The learned rules can then be used to predict occurrences of the named entity, without any reference to the examples in the training dataset. After this training phase, the trained **Classifiers** (that is, the set of rules) are applied to previously unseen set of documents and then the named entity occurrences identified by applying the rules are examined to see whether they are correct or wrong. Machine learning algorithms such as decision trees, naïve Bayes, support vector machines, conditional random fields and maximum entropy are often used for this task. The critical issue in this approach is the amount of time and efforts required to create a sufficiently large and representative labeled training dataset.

**INX Toolset for IE**

INX is an Information Extraction engine from TCS. INX is currently available on MS-Windows platform. Currently, INX supports IE from ASCII text and WORD documents and the extracted information is output to relational tables or XML files. INX users define extraction patterns by means of C#-style regular expressions. INX provides large library of built-in text cleaning and pre-processing functions that can be used to prepare the documents for extraction.
INX also comes with a pre-built set of functionality to automatically extract a number of generic named entities such as PERSON, ORG, LOCATION, PHONE, DATE, AMOUNT, EMAIL, URL and so on. INX also comes with a built-in machine learning algorithm for automatically constructing a gazette for a given named entity, starting with a few seed examples and an unlabeled corpus of documents. INX also supports the machine learning based approach for supervised named entity extraction.

However, the key IE technology in INX really the pattern-based IE, enhanced to allow the user to define a rich variety of syntactic and semantic patterns to extract domain-specific named entities. Given a textual document, INX adds a lot of syntactic and semantic information to it and creates an enriched text format (ETF) of representation. The regular expressions (extraction patterns) are defined to work over this ETF representation of the given documents, rather than on the raw text itself. This allows the user to define high-level (abstract and conceptual) extraction patterns, independent of specific words (Fig. 4). INX has been used in several practical applications; [1], [4], [6], [7].

Defining the extraction patterns is a complex and time-consuming task, demanding considerable expertise from the end-users. For this purpose, we are in the process of packaging INX along with appropriate text pre-processing and extraction patterns to automatically extract pre-determined type of information from specific types of documents. A version of INX, called RINX [7] [9], automatically extracts information from resumes. Another version of INX, called MINX, automatically extracts information from equipment maintenance logs.

![Example of enriched text representation and a semantic IE pattern in INX.](image)

**Figure 4:** Example of enriched text representation and a semantic IE pattern in INX.
IE Methodology

We now briefly describe the steps that need to be followed in any standard IE process.

1. Business Case Definition
In this step, we identify a document repository. Then we identify a set of business questions, whose answers depend on extracting and analyzing the contents of the documents in the repository. For example, the repository may contain automobile insurance claims. A possible business question is: can we identify claims which might be based on ‘staged’ accidents? This requires extracting information about the accident descriptions mentioned in the claims and then analyzing the extracted data to find ‘suspicious’ accidents.

2. IE Problem Definition
Next, we identify the subset of documents to be given for IE for example, only accident descriptions, claimant and witness statements, police reports and adjuster notes. We may further find that the relevant information is available only in specific sections; that is, we identify the relevant sections to be given for IE. Next, we carefully define the type of information (that is, entities) to be extracted from these documents. For the insurance claim example, the IE template may contain accident date, time, place, type of place (for example, highway, by-lane, square, parking lot), number and details of vehicles involved, number and details of passengers in each vehicle, number and details of witnesses, type of accident (head-on, rear end, sideways, hit an object), sequence of events in the accident, whether hit and run and so on.

We should also describe the end-use of the extracted information; for example, how the extracted information will facilitate the decision of whether the accident was staged or not. This is a crucial step and often decides the success or failure of the IE efforts. We then define the quality measures for validating the extracted information. We also define any standardization of extracted information; for example, standard representation formats for dates, times and currency. It helps if we create examples of various types of entities to be extracted; for example, we manually prepare a sample extraction from a few insurance claims. We define accuracy measures required from the output of the IE process.

3. Study of Documents
In this step, we analyze the available input documents to identify:
- Various characteristics of the text involved
- Categories of errors and mistakes present in the text.
- Patterns relevant to the entities that need to be extracted.

We formulate a problem definition for IE based on the goal of extraction, application of extracted entities and so on, that includes definitions of cleaning steps, structure of the information to be extracted, post-processing and standardization of the extracted information and so on.
4. Document Pre-processing
Documents in the repository are cleaned and made ready for extraction. The cleaning steps often include:
- Conversion of documents in WORD, PDF and other formats to ASCII text.
- Removal of unwanted contents from the documents; for example, Unicode characters, headers, footers, pictures and so on.
- Conversion of tables in the document to linear text.
- Spelling and grammar corrections.
- Sentence boundary detection.
- Expansions of abbreviations, short-forms and so on.
- Anaphora resolution.

5. Extraction of Standard Entities
Many IE tools provide a set of standard entities, such as EMAIL, PHONE or AMOUNT. If the entities we are interested in can be mapped to any of these built-in entities, they can be directly extracted and checked for accuracy.

6. Gazette Creation
For entities that can be extracted using gazettes:
- Create the appropriate gazettes.
- Check the gazettes for accuracy and completeness, by correcting or removing wrong entries.
- Standardize the entries in the gazettes.

7. Machine Learning of IE Patterns
Entities for which manually defining patterns are too difficult or cumbersome:
- Create a labeled training dataset.
- Apply the chosen machine learning algorithm in the IE tool.
- Test the learned rules for accuracy on an unseen test corpus.
- If the accuracy is not sufficiently good, improve the training dataset often by adding more labeled documents. One may also engineer suitable features to improve the accuracies, in case feature-based classifiers are used.

8. Extraction Recipes
For all other entities, we use the IE problem definition to identify the rules and extraction patterns for the entities to be extracted. We use the combination of these rules and patterns to prepare a recipe of extraction. This is a critical step since every technique and combinations of techniques have their own pros and cons and the most efficient and most appropriate combination must be chosen keeping in mind the patterns and features of input data.
9. Implementation/Customization and Prototype
This step involves development of the prototype based on sample input documents and the extraction recipes formulated. We implement the extraction rules and test them on the sample input documents. We fine-tune the extraction template with rules and recipes based on the accuracy results of IE output on sample inputs.

10. Software Development and Testing
This phase involves generalizing the prototype to ensure that the extraction recipe works on all documents. We fine-tune the IE recipes, if needed. Software developed is validated using different types of testing such as unit test, stress testing, integration testing and system testing. After completing the testing step and fixing all open problems identified during testing, the system is delivered to the client for User Acceptance Testing and Deployment.

Public-Domain and Commercial IE Tools
Many tools are available both in public-domain; for example, GATE, JULIE, OpenNLP, Stanford NER, GExp, Mallet and commercially, Attensity, OpenCalais, ClaraBridge, SAS Text Analytics, Business Objects, IBM Intelligent Miner and LingPipe. Some tools perform only generic named entity extraction. Some IE tools are specialized for particular domains, for example, Ariadne Genomics Medscan Reader for bio-medical documents or RINX for resumes.
List of Abbreviations

The following table details the list of abbreviations used in this document:

<table>
<thead>
<tr>
<th>Abbreviation/ Acronym</th>
<th>Expansion</th>
</tr>
</thead>
<tbody>
<tr>
<td>IE</td>
<td>Information Extraction</td>
</tr>
<tr>
<td>KM</td>
<td>Knowledge Management</td>
</tr>
<tr>
<td>ETF</td>
<td>Enriched text Format</td>
</tr>
<tr>
<td>RE</td>
<td>Regular Expressions</td>
</tr>
<tr>
<td>ORG</td>
<td>Organization</td>
</tr>
<tr>
<td>RINX</td>
<td>Resume Information eXtractor</td>
</tr>
</tbody>
</table>

References

About Tata Consultancy Services (TCS)

Tata Consultancy Services is an IT services, consulting and business solutions organization that delivers real results to global business, ensuring a level of certainty no other firm can match. TCS offers a consulting-led, integrated portfolio of IT and IT-enabled infrastructure, engineering and assurance services. This is delivered through its unique Global Network Delivery Model™, recognized as the benchmark of excellence in software development. A part of the Tata Group, India’s largest industrial conglomerate, TCS has a global footprint and is listed on the National Stock Exchange and Bombay Stock Exchange in India.

For more information, visit us at www.tcs.com

Contact
For more information, contact innovation.info@tcs.com

Subscribe to TCS White Papers
Feedburner: http://feeds2.feedburner.com/tcswhitepapers

IT Services
Business Solutions
Outsourcing