2 Background

2.1 Information Extraction

Information Extraction (IE) is a process, which takes unseen texts as input and produces, fixed format, unambiguous data as output. This data may be used directly for display to users, or may be stored in a database or spreadsheet for later analysis, or may be used for indexing purposes in Information Retrieval (IR) applications (Cunningham 1999).

It is instructive to compare IE and IR: whereas IR simply finds texts and presents them to the user, the typical IE application analyses texts and presents only the specific information from them that the user is interested in. For example, a user of an IR system wanting information on the share price movements of companies with holdings in Bolivian raw materials would typically type in a list of relevant words and receive in return a set of documents (e.g. newspaper articles) which contain likely matches. The user would then read the documents and extract the requisite information themselves. They might then enter the information in a spreadsheet and produce a chart for a report or presentation. In contrast, an IE system user could, with a properly configured application, automatically populate their spreadsheet directly with the names of companies and the price movements. There are advantages and disadvantages to IE with respect to IR. IE systems are more difficult and knowledge-intensive to build, and are to varying degrees tied to particular domains and scenarios (see below). They are also (for most tasks) less accurate than human readers. IE is more computationally intensive than IR. However, in applications where there are large text volumes IE is potentially much more efficient than IR because of the possibility of reducing the amount of time analysts spend reading texts. Also, where results need to be presented in several languages, the fixed format, unambiguous nature of IE results makes this straightforward in comparison with providing full translation facilities.

There are five types of information extraction (or information extraction tasks) currently under R&D (as defined by the leading forum for this research, the Message Understanding Conferences (Cunningham, 1999, Grishman 1996, SAIC 1998).

Named Entity recognition (NE)
    Finds and classifies names, places etc.

Co-reference Resolution (CO)
    Identifies identity relations between entities in texts.
Template Element construction (TE)
    Adds descriptive information to NE results (using CO).

Template Relation construction (TR)
    Finds relations between TE entities.

Scenario Template production (ST)
    Fits TE and TR results into specified event scenarios.

In simpler terms: NE is about finding entities; CO about which entities and references (such as pronouns) refer to the same thing; TE about what attributes entities have; TR about what relationships between entities there are; ST about events that the entities participate in. Consider these sentences:

The shiny red rocket was fired on Tuesday. It is the brainchild of Dr. Big Head. Dr. Head is a staff scientist at We Build Rockets Inc.

NE discovers that the entities present are the rocket, Tuesday, Dr. Head and We Build Rockets Inc. CO discovers that “it” refers to the rocket. TE discovers that the rocket is “shiny red” and that it is Head’s “brainchild”. TR discovers that Dr. Head works for We Build Rockets Inc. ST discovers that there was a rocket launching event in which the various entities were involved.

From a user point-of-view, NE, TE, TR and ST are the most relevant IE tasks (CO, as noted below, is necessary as an adjunct to the other tasks, but is of limited direct usefulness to the IE system user). NE, TE, TR and ST provide progressively higher-level information about texts. More detail description about tasks of Information Extraction can be found in (Cunningham, 1999).

Another direction of research for automated information extraction is to apply knowledge discovery techniques to the complete textual content of the documents (in a so-called “full text” approach as opposed to approaches only considering indexing key words). This direction is not the same as the definition from Message Understanding Conference (MUC). Indeed, Information Extraction defined by MUC is oriented to extraction “known” information, that is, to extract information whose type is already known by user, while this direction extracts implicit, previously unknown, and potentially useful information from documents. Specialised techniques specifically operating on textual data then become necessary to extract information from such kind of collections of texts. These techniques are gathered under the name of Text Mining and, in order to discover and use the implicit structure (e.g. grammatical structure) of the texts, they may integrate some specific
Natural Language Processing (used for example to pre-process the textual data) (Rajman 1997).

This paper has focused the first type of Information Extraction that defined by MUC.

In the industrial and business domains, the tasks NE and TE are the most interesting. For example, the marketing division wants to extract useful market analysing data from some product catalogue. Customer services can use information extraction to sort and deliver incoming e-mail to responsible consultor or make automatic replay. Automation factories can gather error data from log files produced by digital machine using information extraction.

A typical information extraction process could be described as following: knowledge engineer defines at first empty template, which represents knowledge description or data structure. Then, this template and document are input into the system. System processes at first name extraction. The extracted useful information is then filled into the empty template. The output can be database, XML file, knowledge description or other report documents. Hence, our system focuses mainly on the tasks name extraction (NE) and template filling (TE).

2.2 Document type

Before we go into detail to the technical view of information extraction, we will introduce the different documents types, because the complexity of the information extraction task is fundamentally governed by the characteristics of the documents from which the information is to be extracted. We classify documents with two properties: structure and format property. The first property describes the logical structure of documents, while the last one describes the representation format of documents.

2.2.1 Structured, Semi-Structured and Free Documents

Extraction of information from formal, well-structured documents, where a strong syntax exists for the description of salient information, can be achieved with 100% accuracy through the construction of a parser. Such documents are for example well-formed XML documents or documents generated by robots automatically which have strictly defined fields and position order. Typical examples for such documents are Internet online stock quote; configuration file for industrial automation; log files generated by software monitoring with pre-defined format etc. Conversely, where the source document is completely informal, with no restriction as to where or in what format information is to be written, then the knowledge extraction task is akin
to a full-scale natural language understanding application. Typical free documents in the industrial field are project descriptions, financial news etc.

The middle ground is where many interesting practical applications for Information Extraction currently lie. For these types of application, the document can be referred to as semi-formal. This term is taken loosely here to mean a number of things:

- There is some formality in the basic syntax of the document that can be exploited by the extracting component in order to more readily identify feature instance occurrences.

- The boundary of an individual template instances is reflected in the structure of the document itself.

- The relative position/ordering of feature instances within the textual bounds of a given template instance is standardised and can generally be exploited to aid their recognition.

- The cardinality of a given feature value within a template instance is predictable.

All of the above aspects are in some way applicable for the class of documents tackled with the most interesting. There are lots of domains in this document gene: product catalogue, monthly financial analyse and report, newsgroups, online hotel information, online last minute offers, error reports with email etc. It is assumed that enough syntactic formality exists to make feasible the creation of high quality IE-processing, e.g. based on hand-coded rules. Nevertheless, it is assumed that enough syntactic variation persists to make the construction of perfect IE-processing (i.e. 100% precision / recall) an expensive and difficult task.

That the limits of a template instance can be ascertained from the structure of a document is a strong assumption. This implies some syntactic convention for delimiting the start and/or end of an instance (it may even be that each instance is itself a separated document). Moreover, it is assumed that the document area for one template instance does not infringe on that of another. In other words, the problems of identifying how many template instances exist and of determining to which template instance(s) a given feature instance belongs are assumed trivial. This apparently strong assumption is often justified in many real-life applications, such as filtering email messages or newsgroup entries etc.

Currently, under these three document types (free, semi-structured and structured), the last two types are more interesting practical applications than
the first one for information extraction. Therefore, this paper focuses on IE for semi-structure and structured documents. However, IE methods and software architecture defined for semi-structured and structured documents may be extended to deal with free text in the future (see also Section 11).

2.2.2 Normal Documents and Web (HTML) Documents

The initial task of IE focused only on such document, which are written in plain text with ASCII format. No formatting and structuring information is embedded in the documents. In comparison with documents with embedded formatting and structuring information, we name such plan text documents as normal document.

The most well known documents with embedded formatting and structuring information are web sites. The World Wide Web (WWW), with its explosive growth and ever-broadening reach, is swiftly becoming the default knowledge resource for many areas of endeavor. WWW has become the most common source to get information and resource. Hence, Information Extraction from Web sites becomes an important task in the last 10 years.

The widely used document format in WWW is Hyper Text Markup Language (HTML), which is a simple markup language used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. Compared to normal (not HTML) documents, Web documents consist of rich tag information (such as “<td>”, “<br>” etc.) for layout structuring and formatting. Although such tag information is not the expected underlying values that are extracted by IE-system, it provides helpful structure information, which can be used to generate extraction rules.

If we classify the web documents again with structure property, we will see that web documents can be free text, structured or semi-structured. For instance, if web servers get some requests, extract useful information from structured database, and present the results on the web. Such automatically generated Web pages (e.g. weather report, stock quote etc.) are normally strictly structured and can be considered as structured documents. Some Web pages, on the other hand, are primarily designed for human browsing rather than for understanding by computer programs. However, they have implicit structure and order, which could be defined by some layout template guidelines. A typical example is a set of web sites for description a product family. Although the contents of web sites are not always the same, but they have same layout and structure style. Such documents are in the sense semi-structured (e.g. product catalogue, searching result etc.). The other Web documents are created merely individually and therefore can not be
characterised as one of these two types. They are actually free text documents (e.g. personal homepages).

The recent direction of representation information on the Web is using XML and XSL (W3C). In the XML-family, XML describes layout independent contents, while XSL is used to define the presentation in browser. W3C Organisation introduced also Semantic Web and RDF to represent Machine-Understandable information in Web pages (W3C, Berners 1998). With Semantic Web, Web pages can be written with well-formed XML format. The documents are then more structured than semi-structured or free text style. If online documents are written in XML formats, standard extraction technique, such as XSL-Patterns, XQL etc., can be used to get the desired information (see also Section 7.1.2). However, today most of Web pages are still written with HTML. Mechanically extracting of their whole contents is in general a rather difficult job if not impossible. Most software systems that extract information from web pages do so on the basis of hand-coded or automatically learned wrappers (see Section 3.5).

Table 1 summaries different document types and their typical application domains.

<table>
<thead>
<tr>
<th>Document Type</th>
<th>Structured</th>
<th>Semi-Structured</th>
<th>Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>CIA-Factbook</td>
<td>Newsgroup</td>
<td>Financial reports</td>
</tr>
<tr>
<td></td>
<td>Configuration file</td>
<td>Email (Newsletter, error</td>
<td>Novel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reports etc.)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Industrial submission</td>
<td></td>
</tr>
<tr>
<td>Online (HTML)</td>
<td>Stock Quote</td>
<td>Auction (e.g. eBay)</td>
<td>Web news</td>
</tr>
<tr>
<td></td>
<td>Weather forecast</td>
<td>Product catalogue</td>
<td>Private Homepages</td>
</tr>
</tbody>
</table>

Table 1 Summary of document types