Information Extraction

Information Extraction is the **automatic identification and structured representation of relevant information** in documents:
- extract well-defined pieces of relevant information from collections of document
- goal: populate a database (e.g. metadata)

**General Functionality**
- **Input**
  - Templates coding relevant information, e.g. metadata attributes
  - set of real world texts
- **Output**
  - set of instantiated templates filled with relevant text fragments
Application Scenarios for Information Extraction

- Indexing: Creating indexes for information retrieval systems
  - Automated determination of metadata of documents
- Question Answering
  - Answer an arbitrary question by using textual documents as knowledge base
- Mail distribution
  - Identification of recipients in incoming letters of a company
- Converting unstructured text to structured data
  - Automatic insertion of data into operative application systems and databases
- Evaluation of surveys
  - Capturing and analysis of questionnaires

Information extraction depends on ...

... structural degree of input data
- structured: tables with typed data like numbers
- semi-structured: XML, tables with text
- non-structured: text

... format
- electronic information
  - coded
  - non-coded
- paper documents

... structural degree of output data
- text summary
- fulltext index
- structured data: database, attributes, classification
7.1 Information Extraction from Text Documents

Lexical Analysis

- **Token scanner:**
  - Identification of text structure (e.g. paragraphs, title etc.) and special strings (tokens) like date, time, punctuations
  - HTML or XML-parsers can be applied for markup documents

- **Lexical analysis (morphology):**
  - Determination of word forms (singular-plural)
  - Determination of the kind of word (verb,noun)
    - Part of Speech tagging, POS
  - in German: composita analysis (in German)
Automatic Classification

- Each document is described by a set of features
- Each class is described using the same kind of features
- A document is associated to the class(es) where the features are most similar. This can be tested using rules or similarity measures.

Rule-based Text Classification

- The features are keywords that are either associated to a document as metadata or that occur in the documents
- Example: Assume there are three classes: business, computer science, information systems

  The keywords in this example are: process, OOP, accounting, ERP, database

- The classifier can be represented as a set of rules:
  
  IF a document has the keywords process, accounting, and ERP
  THEN the document belongs to class „business”

  IF a document has the keywords OOP and database
  THEN the document belongs to class „computer science”

  IF a document has the keywords process, database, and ERP
  THEN the document belongs to class „information systems”
**Fulltext Classification**

- In the full text classification, the features are the terms occurring in the documents (fulltext index).
- The classes are represented as vectors.

\[
\begin{align*}
\text{class descriptions} & = \begin{bmatrix} c_1 & c_2 & c_3 \\ w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \\ w_{13} & w_{23} & w_{33} \\ w_{14} & w_{24} & w_{34} \\ w_{15} & w_{25} & w_{35} \\ w_{16} & w_{26} & w_{36} \end{bmatrix} \\
\text{classifier} & = \text{C(FD)}
\end{align*}
\]

- The classification of a document is computed using a well-known ranking function well-known from information retrieval (cosinus).

**Automatic Learning of Classification Rules**

- A characteristic set of documents is manually classified.
- A learning component analyses the features of the documents in the classes.

Training phase:

- Feature identification
- Feature representation \( F_D \) of the document
- Classifier \( C \)
- Classification \( C(F_D) \)
Classification Methods

- Specific Document classifiers, e.g.
  - Linear Least Square Fit (LLSF)
  - Latent Semantic Analysis (LSA)
- Adaptation of general Classifiers, e.g.
  - Decision Trees
    - Explicit rules to test document features
  - K Nearest Neighbor
    - Documents are represented as vectors
    - A new document is compared with all documents of the training set
    - The majority of the k most similar documents gives the classification
  - Zentroid
    - Each class is represented by a prototypical vector
  - Neural Network

Information Extraction

- Example: From business news information about job changes should be extracted
- Sample text:

  Peter Smith left Arconia Ltd. The former director retired on 31 March 2007. His successor is Susan Winter. At the same time George Young became sales manager. He followed John Kelly.

Template Instances that should be extracted from the sample text:

<table>
<thead>
<tr>
<th>PersonOut</th>
<th>PersonIn</th>
<th>Position</th>
<th>Organization</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter Smith</td>
<td>Susan Winter</td>
<td>director</td>
<td>Arconia Ltd</td>
<td>31 March 2007</td>
</tr>
<tr>
<td>John Kelly</td>
<td>George Young</td>
<td>sales manager</td>
<td>Arconia Ltd</td>
<td>31 March 2007</td>
</tr>
</tbody>
</table>
Named Entity Recognition

- Mark into the text each string that represents a person, organization, or location name, or a date or time, or a currency or percentage figure.

- Example:

```
<name type=person>Peter Smith</name>, left <name type=organisation>Arconia Ltd.</name>. The former director retired on <date>31 March 2007</date>. His successor is <name type=person>Susan Winter</name>. At the same time <name type=person>George Young</name> became sales manager. He followed <name type=person>John Kelly</name>.
```

Parsing

- Parsing: Identification of phrase structures: noun phrase (NP), verb phrase (VP), ..

```
S
   NP
     Peter Smith
     left
   VP
     Arconia
     Ltd.
```
Coreference Resolution

- Capture information on corefering expressions, i.e. all mentions of a given entity, including those marked in NE and TE (nouns, noun phrases, pronouns).

- Example:
  - „the former director“ refers to „Peter Smith“
  - „His“ refers to „Peter Smith“
  - „He“ refers to „Georgs Young“
  - „At the same time“ refers to „31 March 2007“

Example text:

<name type=person>Peter Smith</name>, left <name type=organisation>Arconia Ltd.</name>. The former director retired on <date>31 March 2007</date>. His successor is <name type=person>Susan Winter</name>. At the same time <name type=person>George Young</name> became sales manager. He followed <name type=person>John Kelly</name>.

Template Unification

- Information for instantiating a single template often is distributed over multiple sentences. This information has to be collected and unified.

- Template Unification can comprise multiple tasks:
  - **Template Element Recognition (TE)**
    Extract basic information related to organization, person, and artifact entities, drawing evidence from everywhere in the text.
  - **Scenario Template Recognition (ST)**
    Extract prespecified event information and relate the event information to particular organization, person, or artifact entities.
  - **Pattern Recognition (PR)**
    Identification of domain specific patterns ("Microsoft founder" = "Bill Gates")
7.2 Information Extraction from (semi-)structured Document

- Integrated consideration of
  - layout structure
  - logical structure
  - content (semantics)

Example:

Source: A. Dengel, DFKI

Information Extraction using Layout, Logical Structure and Content

Example: Letter

- Address of Recipient
  - Layout: General Rules for position of address block
  - Structure: Recipient consists of name and address

- Recipient
  - Content: Knowledge about named entities and context
    „Dear Mr Trasher“
Guiding Extraction by Classification

Knowledge about document structure can target information extraction

1. Classification:
   - Assigning documents to predefined document classes
   - For the document classes the structural objects are defined

2. Information Extraction
   - Identification of relevant information
   - Targeted search in structural elements

Information Extraction from Markup Documents: XML

Predefined markup guides information extraction and recognition:
- Elements (tags, attributes)
- Structure

```
<researcher>
  <name> Knut Hinkelmann </name>
  <affiliation>
    <university> Fachhochschule Nordwestschweiz </university>
    <group> Wirtschaftsinformatik </group>
  </affiliation>
  <address>
    <street> Riggenbachstrasse 16 </street>
    <city> 4600 Olten </city>
  </address>
  <phone> ++41 62 286 00 80 </phone>
  <email> knut.hinkelmann@fhnw.ch </email>
</researcher>
```
7.3 Information Extraction from Paper Documents

- **Scanning**
  - Result: Image of the document (non-coded information)

- **Preprocessing**
  - Correction
  - Optical Character Recognition OCR
    - Intelligent Character Recognition ICR (advanced OCR e.g. hand writing)
  - Result: Content as text (coded information)

- **Classification**
  - Result: Document class (e.g. invoice of Hamilton Inc., ...)

- **Information extraction**
  - Result: Relevant information in structured form (e.g. amount invoiced)

---

**Information Extraction from forms**

- In forms the layout (position) determines the meaning of information
- The layout must be known to the recognition system
- The form must be separated from the entries (content)
**Types of documents**

- **Fixed form**
  - space for entries fixed
  - Example: Invoice

- **Dynamic form**
  - forms with space for free entries (text, tables)

- **Free documents**
  - no predefined layout

---

**Dokumentklassen**

- Um Informationen extrahieren zu können, muss der Aufbau der Dokumente bekannt sein.
- Dokumentklassen sind Dokumente mit gleichartigem Aufbau.
- Dokumentklassen steuern die Informationsextraktion.
  - Zu jeder Dokumentklasse ist definiert, wo welche Information extrahiert wird.
  - Beispiel: Rechnung:
    - Adresse
    - Bank
    - Kunden-Nr.
    - Bankleitzahl
    - Kontonummer
    - Betrag
- Dokumentklassen können sehr spezifisch sein.
  - z.B. Rechnungsformular der Firma Meyer GmbH
  - in diesem Fall ist genau bekannt, wo die gesuchte Information zu finden ist.
- Dokumentklassen können sehr allgemein sein.
  - z.B. allgemeine Arztrechnung
  - in diesem Fall ist mehr Aufwand bei der Suche nach Information auf dem Dokument notwendig.
Phase 1: Preprocessing

Elimination of lines:
lines negatively influence OCR results

Noise elimination

Uside-down-correction

Rotation correction

Rechnung

Problems with OCR/ICR

- Errors in ambiguities
- Wrong segmentation
**Phase 2: Classification**

Using layout and logic structure as additional features for classification

Layout: lines, tables, ...

Table structure and content ...

Predefined search patterns (regular expressions)

---

**Definition of Document Classes in Document Analysis Systems**

Document Definition Interface:

- Use the mouse to mark areas with relevant information
- Define search pattern, regular expression (e.g., for date) etc. for the expected information

---

Sehr geehrte Damen und Herren,

hiermit kündige ich obige Versicherung fristgerecht
Zusendung einer schriftlichen Kündigungsbestätigung

---
**Phase 3: Information Extraction**

Extract relevant Information from
- Form fields with fixed position
- Search patterns
- Tables
- Regular expression

**Depotnummer:** 8, 7, 6, 5, 4, 3, 2, 1, 0

**Kempen, den 02.11.98**

Rechnungs-Nr.: 8952

**Phase 4: Automatic Verification**

- Database matching: Compare extracted information with content of a database (Levensthein distance)

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netto</td>
<td>571,35 DM</td>
<td>Mehrwertsteuer</td>
<td>85,70 DM</td>
</tr>
<tr>
<td>Brutto</td>
<td>657,05 DM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Logical verification: Checking logical or mathematical conditions

Expression: \( \text{EQUAL}('\text{Netto}') \), \( \text{SUM}(\text{ROI('Netto')}, \text{ROI('Mwst'))} \)

Nettosumme + Mehrwertsteuer = Bruttosumme

Field 'Netto'
Field 'Mwst'
Field 'Brutto'
Phase 5: Manual Verification

Document Analysis Tools provide an interface for manual verification.