Input and Outputs

Inputs:
- A collection of texts

Outputs:
- An answer key (filled template) for each input text

Evaluation:
- Answer keys compared against human-created 'gold standard' answer keys
  - recall & precision

Outline of This Lecture

- Architecture of an Information Extraction System
- Overview of FASTUS
- Named Entity Recognition

An Example Document

San Salvador, 19 Apr 89 (ACAN-EFE) -- [TEXT] Salvadoran President-elect Alfredo Cristiani condemned the terrorist killing of Attorney General Roberto Garcia Alvarado and accused the Farabundo Marti National Liberation Front (FMLN) of the crime.

... Garcia Alvarado, 56, was killed when a bomb placed by urban guerrillas on his vehicle exploded as it came to a halt at an intersection in downtown San Salvador.

... Vice President-elect Francisco Merino said that when the attorney general's car stopped at a light on a street in downtown San Salvador, an individual placed a bomb on the roof of the armored vehicle.
A Corresponding Filled Template

Incident: Date
19 Apr 89

Incident: Location
El Salvador: San Salvador (CITY)

Incident: Type
Bombing

Perpetrator: Individual ID
urban guerrillas

Perpetrator: Organization ID
FMLN

Perpetrator: Confidence
Suspected or Accused by Authorities: FMLN

Physical Target: Description
vehicle

Physical Target: Effect
Some Damage: vehicle

Human Target: Name
Roberto Garcia Alvarado

Human Target: Description
attorney general: Roberto Garcia Alvarado

Human Target: Effect
Death: Roberto Garcia Alvarado

Getting the Inputs: Document Filtering

• Selecting texts for analysis by Information Retrieval is not foolproof

• Can’t assume that all the texts provided are relevant
  — In the Tipster project on microelectronics, 7 of the 1000 articles provided discussed potato chips
  — In MUC-4, attacks on military targets were not considered relevant since by definition terrorist incidents have civilian targets
  — Many texts in the MUC corpus are reports of speeches in which terrorism is condemned rather than reports of incidents

Getting the Inputs: Problems with Relevant Texts

• How much of the text is relevant?
  — Typically only a few paragraphs contain information of interest

• How many answer keys should there be for one text?
  — A newswire report can mention several terrorist incidents
  — A mail message may include several conference announcements

System Architecture

• General Architecture: pipeline approach

• Basic idea:
  — use a cascaded set of modules to separate processing into several stages
  — output of each stage serves as input to next stage
  — Each module will use specific techniques to tackle the problem
  — earlier stages work on smaller units and are largely domain-independent
A System Architecture [Appelt and Israel]

- Tokenisation
- Morphological and Lexical Processing
- Syntactic Analysis
- Domain Analysis

See http://www.ai.sri.com/~appelt/ie-tutorial/

©2003 Macquarie University

A More Detailed System Architecture [Hobbs]

- Text Zoning
  - Preprocessing
  - Preparing
  - Parsing
  - Semantic Interpretation
  - Coreference Resolution
  - Fragment Combination
  - Lexical Disambiguation
  - Template Generation

See http://www.itl.nist.gov/sui/894.02/related_projects/tipster/gen_ie.htm

©2003 Macquarie University

Correspondences

<table>
<thead>
<tr>
<th>Appelt and Israel</th>
<th>Hobbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokenisation</td>
<td>Preprocessing</td>
</tr>
<tr>
<td>Morphological and Lexical Processing</td>
<td>Preprocessing?, Lexical Disambiguation?</td>
</tr>
<tr>
<td>Syntactic Analysis</td>
<td>Preprocessing?; Parsing; Fragment Combination</td>
</tr>
<tr>
<td>Domain Analysis</td>
<td>Lexical Disambiguation?; Semantic Interpretation; Coreference Resolution</td>
</tr>
</tbody>
</table>

©2003 Macquarie University

Text Zoning

- Turns a text into a set of useful text segments
  - For example, identifying useful or important component parts of the text such as headers, paragraphs, clusters of paragraphs, tables;
  - May be "topic"-based, using cue words or statistics
  - Depends on the structures of the texts in the domain of application
- Discards unwanted segments of the text
  - For example, mail headers, signature blocks ... again, depends on what segments are important for the domain of application

©2003 Macquarie University
Text Zoning

- Can make use of explicit logically-oriented markup
  - HTML, XML, SGML
  - Word’s Rich Text Format (maybe), LaTeX
- In the absence of a logically-oriented markup, use typographic information
  - Centered blocks
  - Paragraph breaks
- Low level markup (PostScript, PDF) not so useful

Text Zoning

- Subsequent processing may focus on specific zones
- For example:
  - searches for information about the date of a message can be restricted to the mailer headers
  - Information in tables can be handled by special purpose code
  - Text-zoning code is usually very specific to the kinds of text being handled

Text Zoning

- Example:
  - Messages sent to the Linguist bulletin board are distributed as digests, with multiple messages concatenated before redistribution; a top level summary header is added
  - Overall form and means of message separation are generally consistent from one digest to the next
  - So, special purpose code can be written that knows how to take apart documents from this source

Preprocessing

- Takes as input a stream of characters
- Carries out tokenisation and sentence segmentation:
  - Converts a text segment into a sequence of sentences, each of which is a sequence of lexical items
  - Will disambiguate full stops to distinguish use in abbreviations from sentence terminators
Preprocessing

- Each lexical item is a word with lexical attributes that can be used in subsequent processing
  - Lexical attributes typically derived from lexical resources
- Part-of-speech tagging may be carried out at this point
  - Popular tag set: Penn Treebank
- Named entities such as dates, times, people’s names and company names may be identified
- Spelling correction also carried out at this point: real texts contain errors

The Penn Treebank Tagset

<table>
<thead>
<tr>
<th>CC</th>
<th>Coordinating conjunction</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>Determiner</td>
</tr>
<tr>
<td>FW</td>
<td>Foreign word</td>
</tr>
<tr>
<td>JJ</td>
<td>Adjective</td>
</tr>
<tr>
<td>JJR</td>
<td>Adjective, comparative</td>
</tr>
<tr>
<td>JJJS</td>
<td>Adjective, superlative</td>
</tr>
<tr>
<td>MD</td>
<td>Modal</td>
</tr>
<tr>
<td>NNS</td>
<td>Noun, plural</td>
</tr>
<tr>
<td>NNPP</td>
<td>Proper noun, plural</td>
</tr>
<tr>
<td>POS</td>
<td>Possessive ending</td>
</tr>
<tr>
<td>PP</td>
<td>Possessive pronoun</td>
</tr>
<tr>
<td>RBR</td>
<td>Adverb, comparative</td>
</tr>
<tr>
<td>RBS</td>
<td>Adverb, superlative</td>
</tr>
<tr>
<td>RP</td>
<td>Particle</td>
</tr>
<tr>
<td>SYM</td>
<td>Symbol</td>
</tr>
</tbody>
</table>

The Penn Treebank Tagset

<table>
<thead>
<tr>
<th>TO</th>
<th>to</th>
<th>UH</th>
<th>Interjection</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>Verb, base form</td>
<td>VBD</td>
<td>Verb, past tense</td>
</tr>
<tr>
<td>VBG</td>
<td>Verb, gerund/present participle</td>
<td>VBN</td>
<td>Verb, past participle</td>
</tr>
<tr>
<td>VBP</td>
<td>Verb, non-3rd ps. sing. present</td>
<td>VBZ</td>
<td>Verb, 3rd ps. sing. present</td>
</tr>
<tr>
<td>WDT</td>
<td>wh-determiner</td>
<td>WP</td>
<td>wh-pronoun</td>
</tr>
<tr>
<td>WP</td>
<td>Possessive wh-pronoun</td>
<td>WRB</td>
<td>wh-adverb</td>
</tr>
<tr>
<td>#</td>
<td>Pound sign</td>
<td>$</td>
<td>Dollar sign</td>
</tr>
<tr>
<td>.</td>
<td>Sentence-final punctuation</td>
<td>,</td>
<td>Comma</td>
</tr>
<tr>
<td>:</td>
<td>Colon, semi-colon</td>
<td>(</td>
<td>Left bracket character</td>
</tr>
<tr>
<td>)</td>
<td>Right bracket character</td>
<td>&quot;</td>
<td>Straight double quote</td>
</tr>
<tr>
<td>&quot;</td>
<td>Left open single quote</td>
<td>&quot;</td>
<td>Left open double quote</td>
</tr>
<tr>
<td>'</td>
<td>Right close single quote</td>
<td>&quot;</td>
<td>Right close double quote</td>
</tr>
</tbody>
</table>
Filtering

- Throws away sentences considered to be irrelevant
- Primary consideration here is processing time: no point in expending machine cycles on sentences which are not important to the task
- Relevance decisions can use manually or statistically derived keywords
- Space vs accuracy trade-off: cheaper (ie faster) heuristics for determining relevance may make more mistakes
- Relevance may be zone-dependent

Preparsing

- Observation:
  - In going from a sequence of words to a parse tree, some structures can be identified more reliably than others
- Examples:
  - Noun groups: “the six dead terrorists in the vehicle were…”
  - Appositives: “John Bull, the forty-year old CEO, said…”
  - Some prepositional phrases: “the CEO of the company said”

Preparsing

- Heuristic for determining what to put in preparsing as opposed to parsing
  - Use preparsing for small structures that can be identified with high reliability
  - Leave contentious decisions until later
- Typically uses finite state grammars and special word lists

Parsing

- Takes as input a sequence of lexical items and small-scale structures built by the parser
- Produces as output a set of parse tree fragments, corresponding to subsentential units
- Many parsing techniques are available
  - chart parsing is a popular choice
Parsing

- In a full-blown NLP system, the aim here would be to construct a full parse tree for each sentence.
- For Information Extraction, this is impractical:
  - typical goal is to determine the major elements in the sentence, such as noun phrases and verb complexes
  - usually no attempt made to build an overarching syntactic structure for the sentence as a whole
- Hobbs: “No parser in existence can find full parses for more than 75% or so of the sentences [in real world text].”

Fragment Combination

- Takes as input a set of parse tree fragments derived from a sentence
- Tries to combine the fragments into a representation for the entire sentence

- Generally based on heuristics:
  - overcomes the problems of not having a rich enough syntactic analysis for the entire sentence
  - domain-based heuristics much faster, especially for the long sentences found in real text
Semantic Interpretation

- Generates a semantic structure or logical form or event frame from a parse tree or a collection of parse tree fragments
- What's a semantic structure?
  - An explicit representation of the relationships between the participants in a sentence
  - Who did what to whom (and when, if mentioned)
- Goal is to map syntactic structures into structures that encode information relevant for template filling

Lexical Disambiguation

- Turns a semantic structure with general or ambiguous predicates into a semantic structure with specific, unambiguous predicates
- This task may be carried out in a number of different places in a system
- In restricted domains this may not be an issue — the 'one sense per document' assumption
  - only one sense of the word is used in the complete domain

Coreference Resolution

- Identifies different descriptions of the same entity in different parts of the text and relates them in some way
- A range of anaphoric relationships may need to be dealt with:
  - identity (different ways of referring to the same thing):
    - "Bill Gates ... he ... Microsoft's founder ..."
  - meronymy (part-of relationships between entities)
    - "A new program ... the documentation is weak ..."
  - reference to events
    - "the murder of the civilians was a new development ..."

Coreference Resolution

- Techniques:
  - Number and gender agreement for pronouns
    - "Bill Gates met with Esther Dyson ... she later stated ..."
  - Semantic consistency based on taxonomic information:
    - "Toyota Motor Corp ... the Japanese automaker"
  - Some notion of 'focus'
    - Pronouns typically refer to something mentioned in the previous sentence
Template Generation

- Derives final output templates from the semantic structures
- Carries out low-level formatting and normalisation of data

Outline of This Lecture

- Architecture of an Information Extraction System
- Overview of FASTUS
- Named Entity Recognition

FASTUS

- FASTUS = Finite State Automaton Text Understanding System
- The “Star” in MUC-4
- Developed at SRI International, California
- Basic idea:
  - use a cascaded set of finite state automata to separate processing into several stages
  - output of each stage serves as input to next stage
  - earlier stages work on smaller units and are largely domain-independent

Levels of Processing in FASTUS

1. Complex words, including multiwords and proper names
2. Basic phrases: noun groups, verb groups and particles
3. Complex phrases: complex noun groups and verb groups
4. Domain events: build structures for events of interest
5. Merging structures: merge structures about the same entities or events
Example Input Text

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.

The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and 'metal wood' clubs a month.

Target Output for Example

| TIE-UP-1: |
|-----------------|-----------------------|
| Relationship:   | TIE-UP
| Entities:       | "Bridgestone Sports Co."
|                 | "a local concern"
|                 | "a Japanese trading house"
| Joint Venture Company: | "Bridgestone Sports Taiwan Co."
| Activity:       | ACTIVITY-1
| Amount:         | NT$2,000,000

Target Output for Example

ACTIVITY-1:

| Activity:         | PRODUCTION
| Company:          | "Bridgestone Sports Taiwan Co."
| Product:          | "iron and 'metal wood' clubs"
| Start Date:       | DURING: January 1990

Levels of Processing in FASTUS

1. Complex words, including multiwords and proper names
2. Basic phrases: noun groups, verb groups and particles
3. Complex phrases: complex noun groups and verb groups
4. Domain events: build structures for events of interest
5. Merging structures: merge structures about the same entities or events
Stage 1: Complex Words

- Multisyllabic:
  - 'set up', 'trading house', 'joint venture'
- Company names:
  - 'Bridgestone Sports Taiwan Co'
- People’s names, locations, dates, times, other basic entities
  - PoS tagging + NE recognition
- Context rules are used to handle unknown names

Stage 2: Basic Phrases

- Two kinds of structures in natural language:
  - Those that require world knowledge to disambiguate
  - Those that can be processed reliably using purely syntactic knowledge
- Structures that can be processed reliably:
  - Noun groups: head noun + modifiers to the left
  - Verb groups: verb + auxiliaries + intervening adverbs

Stage 2: Basic Phrases—An Example

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Bridgestone Sports Co.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb Group</td>
<td>said</td>
</tr>
<tr>
<td>Noun Group</td>
<td>Friday</td>
</tr>
<tr>
<td>Noun Group</td>
<td>it</td>
</tr>
<tr>
<td>Verb Group</td>
<td>has set up</td>
</tr>
<tr>
<td>Noun Group</td>
<td>a joint venture</td>
</tr>
<tr>
<td>Preposition</td>
<td>in</td>
</tr>
<tr>
<td>Location</td>
<td>Taiwan</td>
</tr>
<tr>
<td>Preposition</td>
<td>with</td>
</tr>
</tbody>
</table>

| Noun Group       | a local concern        |
| Conjunction      | and                    |
| Noun Group       | a Japanese trading house|
| Verb Group       | to produce             |
| Noun Group       | golf clubs             |
| Verb Group       | to be shipped          |
| Preposition      | to                     |
| Location         | Japan                  |

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.
Stage 2: Basic Phrases

- Noun and verb groups are recognised by finite state grammars
- Examples of noun groups:
  - approximately 5 kg
  - more than 30 people
  - the newly elected president
  - the largest leftist political force
  - a government and commercial project

Stage 3: Complex Phrases

Larger structures are built:
- Appositives are attached to head nouns
  - The joint venture, Bridgestone Sports Taiwan Co.,
- Measure phrases are constructed
  - 20,000 iron and 'metal wood' clubs a month
- 'of' and 'for' prepositional phrases are attached to heads:
  - production of 20,000 iron and 'metal wood' clubs a month
- Noun group conjunctions are built
  - a local concern and a Japanese trading house

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.

The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and 'metal wood' clubs a month.

Relationship: TIE-UP
Entities: Joint Venture Company: "Bridgestone Sports Taiwan Co."
Activity: Amount: —
Stage 3: Complex Phrases

The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and 'metal wood' clubs a month.

Activity: PRODUCTION  
Company: ___  
Product: "iron and 'metal wood' clubs"  
Start Date: ___

Stage 3: Complex Phrases

- Complex verb groups that have the same meaning are mapped to a canonical reading:
  - GM formed a joint venture with Toyota.
  - GM announced it was forming a joint venture with Toyota.
  - GM signed an agreement forming a joint venture with Toyota.
  - GM announced it was signing an agreement to form a joint venture with Toyota.
  - → GM FORMED a joint venture with Toyota.

Stage 4: Domain Events

- Input = a list of complex phrases in order of occurrence
  - anything that is not included in a basic or complex phrase in Stage 3 is ignored here
- Patterns for events of interest are encoded as finite state machines
- State transitions are effected by phrases and the head words in those phrases:
  - ⟨Company NounGroup⟩
  - ⟨Formed PassiveVerbGroup⟩

Stage 4: Domain Events

Pattern: ⟨Company/ies⟩ ⟨Set-up⟩ ⟨Joint-Venture⟩ with ⟨Company/ies⟩

Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house...

Relationship: TIE-UP  
Entities: "Bridgestone Sports Co."  
"a local concern"  
"a Japanese trading house"  
Joint Venture Company: ___  
Activity: ___  
Amount: ___
Stage 4: Domain Events

Pattern: (Produce) (Product)

... to produce golf clubs to be shipped to Japan.

Activity: PRODUCTION
Company: 
Product: "golf clubs"
Start Date: 

Stage 4: Domain Events

Pattern: (Company) (Capitalized) at (Currency)

The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars ...
Stage 5: Merging Structures

- Criteria for merging:
  - Internal structure of noun groups
  - 'nearness'
  - Compatibility of structures

---

Stage 5: Merging Structures

- Activity: PRODUCTION
  - Company: "Bridgestone Sports Taikwan Co.*
  - Product: "Iron and 'metal wood' clubs"
  - Start Date: DURING: January 1990

- Activity: PRODUCTION
  - Company: "Iron and 'metal wood' clubs"
  - Product: "Iron and 'metal wood' clubs"
  - Start Date: DURING: January 1990

Stage 5: Inferential Coreference

- A joint venture has been mentioned
- A joint venture implies the existence of an activity
- An activity has been mentioned
- So: we can infer that the activity that has been mentioned is the same as the activity that has been implied
Stage 5: Inferential Coreference

Outline of This Lecture

- Architecture of an Information Extraction System
- Overview of FASTUS
- Named Entity Recognition

What are Named Entities?

- The MUC Named Entity Task
  - Temporal expressions
  - Number expressions
  - Entity names

"It's a chance to think about first-level questions", said Ms. <enamex type="PERSON">Cohn</enamex>, a partner in the <enamex type="ORGANIZATION">McGlashan & Sarrail</enamex> firm in <enamex type="LOCATION">San Mateo</enamex>, <enamex type="LOCATION">Calif.</enamex>

Why Named Entity Recognition?

- Names may contain unknown words
- Identification of names simplifies parsing
- IE template slots are typically filled with names
- Specific template slots typically require specific names
- Answers to factoid questions typically are names
- Specific types of questions may require finding specific types of named entities
Temporal Expressions

- **TIMEX**
- Tagged tokens are categorized via the TYPE attribute:
  - DATE: complete or partial date expression
  - TIME: complete or partial expression of time of day
  - DURATION: a measurement of time elapsed or period of time during which something lasts

---

Temporal Expressions

- “twelve o’clock noon”
  <TIMEX TYPE="TIME"> twelve o’clock noon </TIMEX>
- “four o’clock in the morning”
  <TIMEX TYPE="TIME"> four o’clock in the morning </TIMEX>
- “5 p.m. EST”
  <TIMEX TYPE="TIME"> 5 p.m. EST </TIMEX>
- “January 1990”
  <TIMEX TYPE="DATE"> January 1990 </TIMEX>
- “fiscal 1989”
  <TIMEX TYPE="DATE"> fiscal 1989 </TIMEX>

---

Temporal Expressions

- TIMEX
  - dates, times and durations
- Can be captured by regular expressions
- Need to handle elided elements properly

---

Ranges

- “175 to 180 million Canadian dollars”
  <NUMEX TYPE="MONEY"> 175 </NUMEX> to <NUMEX TYPE="MONEY"> 180 </NUMEX> million Canadian dollars
- “twelve twenty to three _p_m”
  <TIMEX TYPE="TIME"> twelve twenty </TIMEX> to <TIMEX TYPE="TIME"> three _p_m </TIMEX>
- “from 1990 through 1992”
  from <TIMEX TYPE="DATE"> 1990 </TIMEX> through <TIMEX TYPE="DATE"> 1992 </TIMEX>
- “from five years to 15 years”
  from <TIMEX TYPE="DURATION"> five years </TIMEX> to <TIMEX TYPE="DURATION"> 15 years </TIMEX>
- “between ten and fifteen percent”
  between <NUMEX TYPE="PERCENT"> ten </NUMEX> and <NUMEX TYPE="PERCENT"> fifteen percent </NUMEX>
Number Expressions

- **NUMEX**
- Values for the TYPE attribute:
  - **MONEY**: monetary expression
  - **MEASURE**: standard numeric measurement phrases such as age, area, distance, energy, speed, temperature, volume, and weight, plus syntactically-defined measurement phrases
  - **PERCENT**: percentage (a fraction expressed in terms of hundredths)
  - **CARDINAL**: a numerical count or quantity of some object (in the form of whole numbers, decimals, or fractions)

- "20 million New Pesos"
  <NUMEX TYPE="MONEY">20 million New Pesos</NUMEX>
- "$42.1 million"
  <NUMEX TYPE="MONEY">$42.1 million</NUMEX>
- "million-dollar conferences"
  <NUMEX TYPE="MONEY">million-dollar</NUMEX> conferences
- "one point four million dollars"
  <NUMEX TYPE="MONEY">one point four million dollars</NUMEX>
- "three dollars and three quarters"
  <NUMEX TYPE="MONEY">three dollars and three quarters</NUMEX>

Simple Regular Expressions

- **ENUMEX**
  - money, measures, percents and cardinal numbers
- Can be captured by regular expressions
- Again, need to handle elided elements properly

- **Postal codes/zip codes**
- **Student ID numbers**
- **Telephone numbers**
Entity Names

- **Values for the TYPE attribute:**
  - **PERSON:** named person, family, or certain designated non-human individuals
  - **ORGANIZATION:** named corporate, governmental, or other organizational entity
  - **LOCATION:** name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.) and astronomical locations

- **"U.S. exporters"**
  - `<ENAMEX TYPE="LOCATION">U.S.</ENAMEX>` exporters

- **"Apple computers"**
  - `<ENAMEX TYPE="ORGANIZATION">Apple</ENAMEX>` computers

- **"the oklahoma bombing"**
  - `<ENAMEX TYPE="LOCATION">oklahoma</ENAMEX>` bombing

- **"a delta jetliner"**
  - a `<ENAMEX TYPE="ORGANIZATION">delta</ENAMEX>` jetliner

- **"Hyundai of Korea, Inc."**
  - `<ENAMEX TYPE="ORGANIZATION">Hyundai of Korea, Inc.</ENAMEX>`

- **"Hyundai, Inc. of Korea"**
  - `<ENAMEX TYPE="ORGANIZATION">Hyundai, Inc.</ENAMEX>` of `<ENAMEX TYPE="LOCATION">Korea</ENAMEX>`

- **"the distilled spirits council of the united states"**
  - the `<ENAMEX TYPE="ORGANIZATION">distilled spirits council</ENAMEX>` of the `<ENAMEX TYPE="LOCATION">united states</ENAMEX>`

- **ENAMEX:**
  - persons, locations and organisations

- **Sources of information:**
  - Can use external resources or knowledge sources such as gazetteers, but there will always be names you haven't seen before
  - Look for information inside the document
Gazetteers

- There are many on the web: search for "gazetteer" or "name lists"
- Global Gazetteer
  - [http://www.calle.com/world/](http://www.calle.com/world/): a directory of 2,880,532 of the world’s cities and towns
- US Census data
- Australian Place Names

Information Inside the Document

- For any given instance of a name there can be internal and external evidence
- Internal evidence comes from the name itself
- External evidence comes from other content in the document

Entity Names: Problematic Cases

- Tricky cases:
  - Is Arthur Anderson a person or an organisation?
  - Is Washington a location or a person?
  - Is Granada a company name or a location?
- Sentence initial casing can cause ambiguity:
  - "Suspended Ceiling Contractors Ltd denied the charge."

Entity Names: Problematic Cases

- prepositions
  - City University of New York vs. Museum of Modern Art in New York City
- conjunctions
  - IBM and Bell Labs vs. Victoria and Albert Museum
Entity Names: External Evidence

- Can use external context to help determine categorisation
  - Specific words provide clues: "General Motors analyst"
  - Corporate designators — "Ltd", "Inc", "Pty", ...
  - Titles — "Dr", "Mr", "Rt Hon", ...
  - Subcategorisation requirements
    - Human-subject verbs, e.g. "say"

Rules for Person Names

- Use a list of known first names
- If you find a capitalised word:
  - Check to see if it is a known first name
  - Check to see if following word is capitalised

A Strategy for Entity Names

- Use rules that take account of context
- Only tag if context is suggestive or non-contradictory
  - "in the Washington area"
- Look for evidence elsewhere in the text

Looking for Evidence

- Suppose "Lockheed Martin Production" is a candidate organisation ENAMEX on the basis of context rules:
  - Look for partial orders of composing words: "Lockheed Martin", "Lockheed Production", "Martin Production", "Lockheed", "Martin" ...
  - Mark as possible organisations
Context Rules – Edinburgh MUC System

- To write context rules you need to build up a hierarchy of patterns
- Basic character patterns: Xxxx+, D, ...
- Part of speech tags: JJ, NN, ...
- Semantic categories defined by lists of values: REL, PROF, LOC, ...
- Semantic categories defined by rules: PERSON-NAMES, PROF, ...

Context Rules from the Edinburgh MUC System

<table>
<thead>
<tr>
<th>Context Rule</th>
<th>Assign</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xxxx+ is at JJ* PROF</td>
<td>PERS</td>
<td>Yuri Gromov is a former director</td>
</tr>
<tr>
<td>PERSON-NAMES is at JJ* REL</td>
<td>PERS</td>
<td>John White is a brother</td>
</tr>
<tr>
<td>Xxxx+, a JJ* PROF</td>
<td>PERS</td>
<td>White, a retired director</td>
</tr>
<tr>
<td>Xxxx, whose REL</td>
<td>PERS</td>
<td>Hunberg, whose stepfather</td>
</tr>
<tr>
<td>Xxxx+ himself</td>
<td>PERS</td>
<td>White himself</td>
</tr>
<tr>
<td>Xxxx+, DD+,</td>
<td>PERS</td>
<td>White, 33</td>
</tr>
<tr>
<td>shares of Xxxx+</td>
<td>ORG</td>
<td>shares of Eagle</td>
</tr>
<tr>
<td>PROF of at with Xxxx+</td>
<td>ORG</td>
<td>director of Trinity Motors</td>
</tr>
<tr>
<td>init LOC</td>
<td>LOC</td>
<td>in Washington</td>
</tr>
<tr>
<td>Xxxx+ area</td>
<td>LOC</td>
<td>Berbiljan area</td>
</tr>
</tbody>
</table>

Statistical Approaches to NE Recognition

- A reformulation of the problem:
  - We have $n$ category types
  - For /in $n$, a specific word can be classified as:
    - /start: the word starts category /
    - /continue: the word continues category /
    - /end: the word ends category /
    - /unique: the word starts and ends category /
    - other: the word is not part of any named entity
  - NE-recognition is now seen as a problem of word classification

<table>
<thead>
<tr>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xxxx+</td>
<td>Sequence of capitalised words</td>
</tr>
<tr>
<td>DD</td>
<td>A digit</td>
</tr>
<tr>
<td>PROF</td>
<td>A profession (director, manager, analyst ...)</td>
</tr>
<tr>
<td>REL</td>
<td>A relative (sister, nephew, ...)</td>
</tr>
<tr>
<td>JJ*</td>
<td>A sequence of zero or more adjectives</td>
</tr>
<tr>
<td>LOC</td>
<td>A known location</td>
</tr>
<tr>
<td>PERSON-NAMES</td>
<td>A valid person name recognised by a name grammar</td>
</tr>
</tbody>
</table>
Example — Decision Lists

• Apply a hierarchy of tests until a decision is taken
• These tests are based on a set of features (see next slide)
• Goal: automatically build the hierarchy of tests
  — Choose the test that reduces the uncertainty about the set of target classes the most (e.g. split the set of possible classes evenly)
• Popular learning algorithms — developed by Ross Quinlan:
  — ID3
  — C4.5

Examples of Features (Gallipi, COLING 96)

<table>
<thead>
<tr>
<th>Type</th>
<th>Feature</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>Proper Noun</td>
<td>Aristotle</td>
</tr>
<tr>
<td></td>
<td>Common Noun</td>
<td>philosophy</td>
</tr>
<tr>
<td>Disgnator</td>
<td>Company</td>
<td>Corp_Ltd</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>Mr. President</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>Country, State, City</td>
</tr>
<tr>
<td></td>
<td>Date</td>
<td>Month, Day of week</td>
</tr>
<tr>
<td>Morphology</td>
<td>Capitalization</td>
<td>A, B</td>
</tr>
<tr>
<td></td>
<td>Company suffix</td>
<td>.corp, .tee</td>
</tr>
<tr>
<td></td>
<td>Word length</td>
<td>W1,8, W1,3</td>
</tr>
<tr>
<td>List</td>
<td>Companies</td>
<td>IBM, AT&amp;T</td>
</tr>
<tr>
<td></td>
<td>Persons</td>
<td>Smith, Michael</td>
</tr>
<tr>
<td></td>
<td>Keywords</td>
<td>Based in, said he</td>
</tr>
<tr>
<td>Template</td>
<td>Company</td>
<td>NNP CN, desc</td>
</tr>
<tr>
<td></td>
<td>Person</td>
<td>P_design NNP</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>NNP L, desig</td>
</tr>
<tr>
<td></td>
<td>Date</td>
<td>MM Num, Num</td>
</tr>
<tr>
<td></td>
<td>Proper Name</td>
<td>NNP NNP</td>
</tr>
<tr>
<td>Special</td>
<td>LCS</td>
<td>VW &lt; Volkswagen</td>
</tr>
<tr>
<td>purpose</td>
<td>Duplicated PNs</td>
<td>DUP_2</td>
</tr>
</tbody>
</table>

Hidden Markov Models

• Idea: Predict the class of the current word given:
  — the class of the preceding word (NC<sub>i</sub>)
  — the current word (w<sub>i</sub>)

Hidden Markov Models

• Formulas:
  \[ P(NC | NC_{i-1}, w_{i-1}) = \frac{c(NC, NC_{i-1}, w_{i-1})}{c(NC_{i-1}, w_{i-1})} \]
  \[ P(\langle w, f \rangle | \langle w, f \rangle_{i-1}, NC) = \frac{c(\langle w, f \rangle, \langle w, f \rangle_{i-1}, NC)}{c(\langle w, f \rangle_{i-1}, NC)} \]

• Where:
  — NC = current name class
  — NC<sub>x</sub> = name class x words back
  — c(W) = count number of times the event W appears in the training corpus
  — w = a word
  — f = a feature