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

**Example of IE from MUC/FASTUS (1993): it isn't easy!**

**Example of IE: FASTUS (1993)**

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

**Example of IE: FASTUS (1993)**

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

**Example of IE: FASTUS (1993)**

**Example of Existing IE Systems**

- Systems to summarize medical patient records by extracting diagnoses, symptoms, physical findings, test results, and therapeutic treatments.
- Gathering earnings, profits, board members, etc. from company reports.
- Verification of construction industry specifications documents (are the quantities correct/reasonable?)
- Classified advertisements: jobs, real estate, etc.
- Building museum database of artefacts from descriptions.
- Extraction of company take-over events.
- Extracting gene/protein interactions from biomedical texts.
Three generations of IE systems

- **Hand-Built Systems - Knowledge Engineering [1980s -]**
  - Rules written by hand
  - Require experts who understand both the systems and the domain
  - Iterative guess-test-reject cycle
- **Automatic, Trainable Rule-Extraction Systems [1990s -]**
  - Rules discovered automatically using predefined templates, using methods like ILP
  - Require huge, labeled corpora (effort is just moved)
- **Statistical Generative Models [1997 -]**
  - Greedy decode the statistical model to find which bits of the text were relevant, using HMMs or statistical parsers
  - Learning usually supervised; may be partially unsupervised

**Example of IE: FASTUS (1993)**

- **FASTUS**
  - Based on finite state automata (FSA) transductions
  - **1. Complex Words:** Recognition of multi-words and proper names
  - **2. Basic Phrases:** Simple noun groups, verb groups and particles
  - **3. Complex Phrases:** Complex noun groups and verb groups
  - **4. Domain Events:** Patterns for events of interest to the application
  - **5. Merging Structures:** Templates from different parts of the text are merged if they provide information about the same entity or event

**Example of IE: FASTUS (1993)**

**Pattern-matching**

- **(PN ‘s (ADJ)+ (P (ADJ)+ N)+)**

**Example of IE: FASTUS (1993)**

- **Attachment Ambiguities are not made explicit**
Rule-based Extraction Examples

Determining which person holds what office in what organization:
- [person], [office] of [org]
  - Vuk Draskovic, leader of the Serbian Renewal Movement
- [org] (named, appointed, etc.) [person] [office]
  - NATO appointed Wesley Clark as Commander in Chief

Determining where an organization is located:
- [org] [loc]
  - NATO headquarters in Brussels
- [org] [loc] (division, branch, headquarters, etc.)
  - KFOR Kosovo headquarters

Automatic Frame Learning Systems

Pros:
- Portable across domains
- Tend to have broad coverage
- Robust in the face of degraded input
- Automatically finds appropriate statistical patterns
- System knowledge not needed by those who supply the domain knowledge.

Cons:
- Annotated training data, not lots of it, is needed.
- Isn't necessarily better or cheaper than hand-built sol'n
- Classic e.g.: Riloff et al., AutoSlog (UMass) learns lexico-syntactic extraction patterns from templates

Autoslog (Riloff)

Annotated Texts

Sentence Analyzer

R : A bomb

VP: exploded

PP: on the plane

PP: over the ocean.

Extraction Patterns

detonator exploded

exploded on target(s)

AutoSlog Heuristics

Rapier [Califf & Mooney, AAAI-99]

- Rapier learns three regex-style patterns for each slot:
  - Pre-filler pattern
  - Filler pattern
  - Post-filler pattern

- One of several recent trainable IE systems that incorporate linguistic constraints, see also: SIF (Miller et al., MUC-7) SRV [Freitag, AAI-92], Whisk [Switzerland, ML-93]

- "...paid $11M for the company...

- "...sold to the bank for an undisclosed amount...

- "...paid Honeywell an undisclosed price...

Pre-filler: 1) tag: (nn,mp) 2) word: undisclosed 1) sem: price
Post-filler: 2) list: length 2 1) tag: jj

RAPIER rules for extracting "transaction price"

Part-of-speech tags & Semantic classes

- Part of speech: syntactic role of a specific word
  - noun (nn), proper noun (npi), adjective (JJ), adverb (RB),
    determiner (DT), verb (VB, "'t", "'s")...
  - NLP: Well-known algorithms for automatically assigning POS
tags to English, French, Japanese, ... (<95% accuracy)

- Semantic Classes: Synonyms or other related words
  - "Price" class: price, cost, amount, ...
  - "Month" class: January, February, March, ..., December
  - "US State" class: Alaska, Alabama, ..., Washington, Wyoming
  - WordNet: large on-line thesaurus containing (among other things) semantic classes

Rapier rule matching example

- "...sold to the bank for an undisclosed amount...

POS: SClass:
    vb pr det nn pr det jj mn

Pre-filler: 1) tag: (nn,mp) 2) word: undisclosed 1) sem: price
Post-filler: 2) list: length 2 1) tag: jj

- "...paid Honeywell an undisclosed price...

POS: SClass:
    vb mp det jj mn

Pre-filler: 1) tag: (nn,mp) 2) word: undisclosed 1) sem: price
Rapier Rules: Details

- Rapier rule -
  - pre-fill pattern
  - fill pattern
  - post-fill pattern
- pattern = subpattern +
- subpattern = constraint +
- constraint:
  - Word: exact word that must be present
  - Tag: matched word must have given POS tag
  - Class: semantic class of matched word
  - Can specify disjunction with "|"
  - List length N: between 0 and N words satisfying other constraints

Rapier’s Learning Algorithm

- Input: set of training examples (list of documents annotated with “extract this substring”)
- Output: set of rules
- Rule: Rules = a rule that exactly matches each training example
- Repeat several times:
  - Seed: Select M examples randomly and generate the K most-accurate maximally-general filter-only rules (prefiller = postfiller = “true”).
  - Grow: Repeat for N = 1, 2, 3,...
    - Try to improve K best rules by adding N context words of prefiller or postfiller context
  - Keep: Rules = Rules ∪ the best of the K rules - subsumed rules

Information Extraction: Learning Lexico-Syntactic Patterns (Autoslog-TS, Riloff)

- Reduces greatly the needed resources: just classifications
- start with known relevant articles; extract noun phrases
- Create huge set of patterns of form:
  - <slot> + passive-verb <agent> was murdered
  - active-verb <agent> bombed <targets>
- Use: key lexical items, syntactic frames, semantic sorts
- Compute relevance rate:
  - Prevalence [text has pattern] = m/r
  - m = freq, r = total freq,
  - Rank patterns in order of importance
    - How, relevance rate * log2(freq), rr > 0.5
  - Human judge reviewed top patterns
  - Apply all patterns to each text. Worked almost as well

Statistical generative models

- Pros:
  - Well-understood underlying statistical model makes it easy to use wide range of tools from statistical decision theory
  - Portable across domains
  - Tend to have broad coverage, robustness, good recall
- Drawbacks
  - Range of features and patterns available may be limited
  - Not necessarily as good for complex mult-slot patterns
- Good current e.g.: Freitag & McCallum (CMU, JustSystems, Whizbang! Labs) statistically-learned models using Bayes classifiers, HMMs, etc.

Freitag and McCallum details

- Partly fixed structure, partly hidden (constrained EM using remote supervision)
- Parameter tying and shrinkage smoothing techniques
- Better just to use a good unknown model?
- Structure learning of transition structure
- Why not just plain EM?
- Results good on semi-structured data
- Still rather modest on free form text
  - Need richer model class?