Empirical Methods in Information Extraction

By Claire Cardie

Presentation by Dusty Sargent

Background

- Domain-specific task differs from more general problems studied so far
- Summarizes important points in a text with respect to a target topic
- Structures information for storage into database

Event:	tornado
Date:	4/3/97
Time:	19:15
Location:	Farmers Branch : "northwest of Dallas" : TX : USA
Damage:	"mobile homes" (2)
	"Texaco station" (1)
Estimated Losses:	$350,000
Injuries:	none

Background (cont'd)

- MUC (Message Understanding Conference) evaluates systems
- Provides answer keys and texts for particular topic
- Recall = (# correct slot fillers in output template) / (# of slot-fillers in answer key)
- Precision = (# correct slot fillers in output template) / (# of slot-fillers in output template)
- Has been used in practical applications

Applications

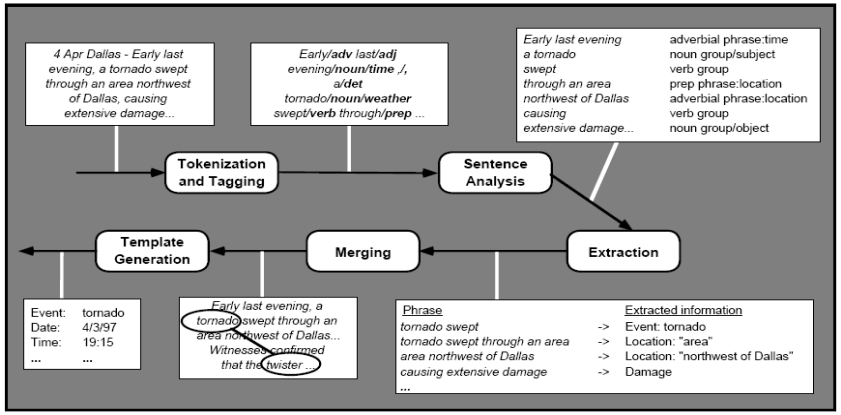
- Summarize medical records (test results, diagnoses, symptoms, etc.)
- Extract information about terrorist activities from radio or television broadcasts
- Keep records of corporate mergers and acquisitions
- Build knowledge bases from information found in websites
- Create job listings from web-based classified ads, job-search sites and newsgroups

Performance

- State of the art systems reach 50% recall and 70% precision on complicated extraction problems
- Can reach 90% precision and recall on the easiest extraction tasks
- Human error rate also high for information extraction
- Best systems have only twice error rate of human experts trained for same task
- Still a lot of room for improvement
- Time consuming development phase and cause of errors difficult to determine

Architecture

- Traditional NLP approach with full syntactic and semantic analysis of input text
- Less common simple approach with keyword matching and little linguistic analysis



Architecture (cont'd)

- Tagging and tokenization: divide input into sentences and words, part-of-speech tag and disambiguation word senses
- Sentence analysis: partial parse and tag with respect to semantic roles
- Extraction: identify relevant entities and relations between them, specific to the domain
- Merging: coreference resolution between extracted entities and events
- Template generation: map extracted information into domain specific output format

Corpus-based Learning

- Used for the underlying tasks of information extraction
- Can apply to preliminary stages of the architecture
- Difficulty in finding enough training data for all the levels of analysis required
- Expensive to retrain the system for each domain to which it must be applied
- Standard NLP learning techniques difficult to apply to later stages: learning extraction patterns, coreference resolution, template generation
- New training corpus needed for each task; difficult to learn general patterns from answer keys

Learning Extraction Patterns

- Use general pattern matching techniques for extraction phase
- Acquire good extraction patterns from training corpus with empirical methods
- Similar to Candidate Elimination Algorithm
- Extraction patterns ordered from general to specific, need balance between the two
- Need general patterns to apply to more than one case
- Patterns must be specific enough that they do not apply in the wrong context

AutoSlog

- One of earliest systems for learning extraction patters, by Lehnert and Riloff (1992 – 1993)
- Learns "concept nodes", domain-specific semantic frames, maximum of one slot per frame
- Concept nodes used with CIRCUS parser for the final extraction task

Sentence Two: "Witnesses confirm that the twister occurred without warning at approximately 7:15 p.m and *destroyed two mobile homes.*"

Concept Node Definition:
Concept = Damaged-Object
Trigger = "destroyed"
Position = direct-object
Constraints = ((physical-object))
Enabling Conditions = ((active-voice))

Instantiated Concept Node
Damaged-Object = "two mobile homes"

Concept Node Definition

- Concept: the concept to be extracted, e.g. Damaged-Object
- Trigger: word that activates pattern
- Position: syntactic position where the concept is likely to be found in the sentence
- Constraints: constraints on argument at "Position" necessary for extraction to occur; can be hard or soft
- Enabling Conditions: constraints on linguistic context of trigger word

Example Application

- Example: "...the twister occurred at approximately 7:15pm and destroyed two mobile homes."
- Concept is Damaged-Object
- Concept node is activated by trigger word "destroyed"
- Enabling Condition: "destroyed" occurs in active voice
- Position: direct-object of verb "destroyed"
- Constraints: direct-object of "destroyed" must be a physical object
- Result: "two mobile homes" is extracted to fill the Damaged-Object slot of the concept node

Concept Node Algorithm

- Concept nodes applied during partial parsing phase of the extraction system
- When trigger word encountered, check for enabling conditions
- If met, extract phrase in appropriate position
- Test phrase for constraints
- If constraints met, label phrase as instance of the concept type

Learning Concept Nodes

- Learning algorithm specific to domain
- Requires training text with noun phrases annotated with concept type, or uses answer keys
- Uses partial parse and small set of linguistic patterns to help learn concept nodes
- New version, AutoSlog-TS, only needs to be given texts marked as relevant or irrelevant to the domain of the extraction task

Learning Algorithm

- Find sentence in which target noun phrase occurs in training data
- Parse the sentence with partial parser
- Apply the list of linguistic patterns in order
- If a pattern linguistic pattern applies to the sentence, create a concept node definition from the appropriate elements of the sentence

Learning Example

- "Witnesses confirm that the twister occurred without warning at approximately 7:15pm and destroyed *two mobile homes*(Damaged_Object)".
- Target noun phrase is "two mobile homes", marked in training corpus as an instance of the concept Damaged_Object, or found in the Damaged_Object field in the answer key
- Step 1: find the above sentence in the training corpus, in which the target noun phrase occurs
- Step 2: parser determines that "two mobile homes" was the direct object of active verb "destroyed" in the third clause
- Step 3: match third clause to the following linguistic pattern: <active-voice-verb> followed by <target-np> = <direct-object>
- Step 4: generate the concept node seen previously from matched constituents, context, concept type, and semantic class

AutoSlog-TS

- Improved version needs only relevant and irrelevant texts as training data
- Adapts AutoSlog to use statistical techniques
- Nearly matches performance of AutoSlog on MUC 4 extraction task, using a fraction of the human effort
- Scans corpus once and generates an extraction pattern for every noun phrase
- Scans again and ranks extraction patterns according to some ranking function

PALKA

- Learns extraction patterns similar to concept nodes using a different method
- Uses a concept hierarchy, predefined set of trigger words, and semantic class lexicon
- Concept hierarchy contains generic semantic case frames for each concept
- Looks for sentences in corpus containing keywords, and fills case frame slots using semantic class information

CRYSTAL

- Uses more complex patterns in the form of semantic case frames
- Triggers are detailed descriptions of linguistic context of target noun phrase
- Can test for specific sequences of words or types of related constituents
- Learns patterns by generalizing input examples until an error threshold is reached
- Begin by generating most specific possible patterns and gradually relax constraints

Other Systems

- LIEP recognizes relationships between two target noun phrases that fill slots in the output template
- RAPIER generalizes from an input set of patterns, but operates at word level, unlike CRYSTAL
- Existing methods only work well at extracting noun phrases
- Few methods have been evaluated under similar conditions
- Difficult to compare and determine advantages of different approaches

Coreference Resolution

- The most difficult pre-processing task for information extraction systems
- Less research for coreference resolution and template generation than for learning extraction patterns

[Motor Vehicles International Corp.] announced a major management shake-up ... [MVI] said the chief executive officer has resigned ... [The Big 10 auto maker] is attempting to regain market share. ... [It] will announce significant losses for the fourth quarter ... A [company] spokesman said [they] are moving [their] operations to Mexico in a cost-saving effort. ... [MVI, [the first company to announce such a move since the passage of the new international trade agreement],] is facing increasing demands from unionized workers. ... [Motor Vehicles International] is [the biggest American auto exporter to Latin America].

Coreference Resolution

- Different methods needed to handle each linguistic type of reference
- Coreference resolution is the major weak point of most modern information extraction systems
- Many systems use heuristics, but difficult to cover all possible cases
- Often, heuristics require detailed parses, which most extraction systems do not provide
- Accumuated errors from earlier parsing and variety of domains adds to difficulty

Empirical Methods

- Do not need to make specific learning methods for this task as with learning extraction patterns
- Can cast coreference resolution as a classification problem and use existing inductive learning methods
- Given two noun phrases and their contexts, classify as positive if they refer to the same object, negative if they do not
- Use inductive learning to automatically derive coreference resolution heuristics

General Approach

- Step 1: link all coreferential phrases via annotations in the training corpus
- Step 2: create positive and negative training examples from all possible pairs in the corpus
- Step 3: annotate examples with features relating to their contexts and classes
- Step 4: use learning algorithm to derive a classifier based on the examples, often use decision trees for this purpose
- Systems have been compared at the MUC coreference competition

MLR (Machine Learning based Resolver)

- Uses C4.5 decision tree learning algorithm
- Derives feature set from earlier parsing stages
- Uses a data set derived automatically by its information extraction system
- Instances are described in terms of domain-independent linguistic features
- Tested on MUC-6 coreference resolution tasks using Japanese business joint ventures corpus
- Scored recall of 67-70% and precision of 83-88% on MUC-6 coreference resolution tasks

Resolve

- Also uses C4.5 decision tree learning algorithm and derives features from earlier parsing stages
- Has advantage because it uses manually annotated training data
- Features are very domain specific
- Tested on the English version of the business joint ventures corpus, contains 74% negative examples
- Scored 80-85% recall and 87-92% precision at MUC-6 conference
- Less labor intensive than manually coded coreference algorithms

Results/Research

- Possible to make automatically trained systems that approach performance of manually coded systems
- No need to develop specific algorithms for coreference resolution
- Need to test different feature sets, hopefully domain-independent features
- Need to determine the effect of using domain specific information and test outside of the information extraction domain
- Determine effects of errors in earlier stages

Future Directions

- Information extraction is a relatively new sub field of natural language processing
- Use statistical methods to avoid the need for large amounts of domain-specific training data
- Develop domain-independent systems that do not need to be retrained for new each extraction task
- Many algorithms exist, but there is little training data available and it is expensive to produce a new corpus for each task

Future Directions (cont'd)

- Partial solution to making domain-independent systems: build systems that end user can train by themselves for new tasks
- For this goal, need algorithms that can fully specify an extraction system using just answer keys
- Demand by industry, military, etc. for practical systems increases with the amount of online text
- To meet demand, must eventually make systems that work autonomously and can handle any domain without tuning