Information extraction

- **Introduction**
  - Task definition
  - Evaluation
  - IE system architecture
- **Acquiring extraction patterns**
  - Manually defined patterns
  - Learning approaches
    - Semi-automatic methods for extraction from unstructured text
    - Fully automatic methods for extraction from structured text
  - Semi-structured text
- **Named entity detection**
- **Sequence-tagging methods for IE**

Why?

- Provide intuition for useful ***features*** for the machine learning approaches

Learning IE patterns from examples

- **Goal**
  - Given a training set of ***annotated*** documents
    - answer keys / gold standard
  - Learn extraction patterns for each slot type using an appropriate machine learning algorithm.

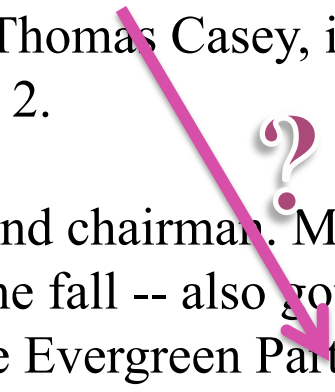
Changes in Management

Evergreen Information said Barry Nelsen, who had a heart-bypass operation last week, resigned as president and chief executive. The board formally accepted the resignation of Thomas Casey, its former chairman, who stepped down effective Feb. 2.

Martin Bell was named president, CEO, and chairman. Mr. Bell -- who has been chief financial officer since the fall -- also got voting control of 970,000 shares held by the Evergreen Partnership, a vehicle for the company's three co-founder

Excluding these shares, Evergreen Inform million shares or exercisable warrants outsta spokeswoman.

The computer products and services conce fewer than 10 employees from about 35, and managers' salaries. In a press release, it said company is still viable.



In-out-event

Type:	OUT
Person:	“Barry Nelsen”
Position:	PRESIDENT, CHIEF EXECUTIVE
Company:	“Evergreen Information”

Natural disasters

The twister occurred without warning at approximately 7:15p.m. and destroyed ***two mobile homes***.

Natural disaster	
Type:	TORNADO
Damaged-obj:	“two mobile homes”

Syntactico-semantic patterns

The twister occurred without warning at approximately 7:15p.m. and destroyed ***two mobile homes***.

Pattern:

Trigger: "destroyed"

condition: active voice verb?

Slot: Damaged-Object

Position: direct-object

condition: DO is a physical-object?

Learning IE patterns from examples

- **Goal**
  - Given a training set of ***annotated*** documents
    - Answer keys
    - Annotated text spans
  - Learn extraction patterns for each slot type using an appropriate machine learning algorithm.

Learning IE patterns

- **Methods vary with respect to**
  - The **class of pattern** learned (e.g. lexically-based regular expression, syntactic-semantic pattern)
  - **Training corpus** requirements
  - Amount and type of **human feedback** required
  - Degree of **pre-processing** necessary
  - **Other resources**/knowledge bases presumed

Information extraction

- **Introduction**
  - Task definition
  - Evaluation
  - IE system architecture
- **Acquiring extraction patterns**
  - Manually defined patterns
  - Learning approaches
    - Semi-automatic methods for extraction from unstructured text
    - Fully automatic methods for extraction from structured text
  - Semi-structured text
- **Named entity detection**
- **Sequence-tagging methods for IE**

Learning syntactico-semantic patterns

The twister occurred without warning at approximately 7:15p.m. and destroyed ***two mobile homes***.

Pattern:

Trigger: "destroyed"

condition: active voice verb?

Slot: Damaged-Object

Position: direct-object

condition: DO is a physical-object?

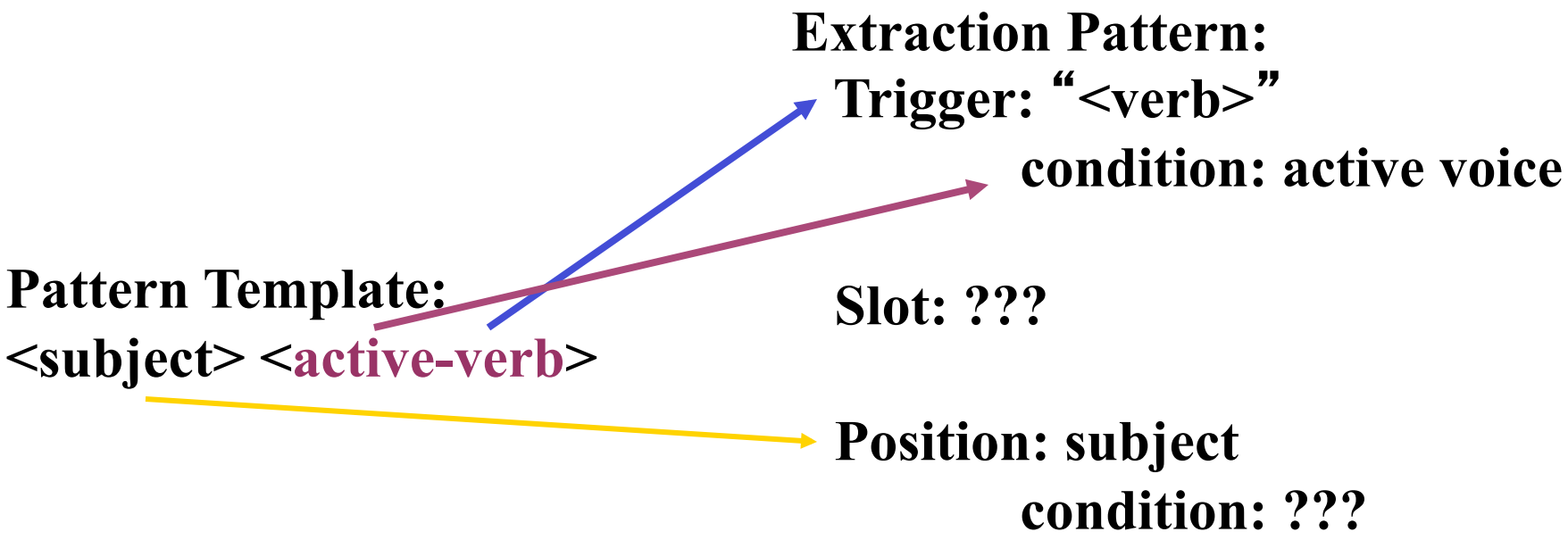
Autoslog

Pattern templates

NP extraction; NPs in prominent grammatical roles

<subject> <passive-verb>	<victim> was **murdered**
<subject> <active-verb>	<perpetrator> **bombed**
<subject> <infinitival-verb>	<perpetrator> attempted to **kill**
<subject> <auxiliary-verb>+<noun>	<victim> was **victim**
*<passive-verb> <dobj>	**killed** <victim>
<active-verb> <dobj>	**bombed** <target>
<infinitive> <dobj>	**to kill** <victim>
<verb>+<infinitive> <dobj>	threatened to **attack** <target>
<gerund> <obj>	**killing** <victim>
<noun>+ <auxiliary> <dobj>	**fatality** was <victim>
<noun>+<prep> <np>	**bomb** against <target>
<active-verb>+<prep> <np>	**killed** with <instrument>
<passive-verb>+<prep> <np>	was **aimed** at <target>

Template → Extraction Pattern



Semantic restrictions table

[domain-specific role]

[semantic constraint]

- **Perpetrator**
  - Person, government, terrorist organization
- **Target (damaged-object)**
  - Building, vehicle, physical-object
- **Victim**
  - Person
- **Location**
  - Location
- **Date**
  - Date
- **Instrument**
  - Weapon

Template → Extraction Pattern

The twister occurred without warning at approximately 7:15p.m. and ***destroyed two mobile homes***.

damaged-object

Pattern:

Trigger: "<verb>"

condition: active voice

Slot: <slot-type> of <target-np>

Position: direct-object

condition: <<semantic class> for <slot-type>>

Semantic restrictions table

Template → Extraction Pattern

The twister occurred without warning at approximately 7:15p.m. and ***destroyed two mobile homes***.

damaged-object

Pattern Template:

Trigger: "<verb>"

condition: active voice

Slot: <slot-type> of <target-np>

Position: direct-object

condition: DO is <<semantic class> of <slot-type>>

Extraction Pattern:

Trigger: "destroyed"

condition: active voice verb?

Slot: Damaged-Object

Position: direct-object

condition: DO is a physical-object?

Autoslog algorithm

- **For each annotated "string fill", *s*, in the training data**
  - Parse the sentence that contains *s.* Also obtain NE and semantic class information for all of its NPs.
  - Apply the syntactic pattern templates in order. Execute the first one that applies to determine:
    - the *trigger* word
    - the triggering *constraints* (syntactic)
    - the *position* of phrase to be extracted (grammatical role)
  - Determine *slot type*
    - The annotated slot type for *s* in the training corpus
  - Determine the *semantic constraints*
    - Defined a priori based on typical semantic class of fillers
      - Semantic restrictions table
  - Create and save the extraction pattern

Applying the patterns

The bombs destroyed and completely leveled ***two mobile homes***.

Extraction pattern:

Trigger: "destroyed"

condition: active voice verb?

Slot: Damaged-Object

Position: direct-object

condition: DO is a physical-object?

Extracts:

Slot: Damaged-Object

Position: "two mobile homes"

Autoslog algorithm characteristics

- Domain-independent pattern templates
  - So require little/no modification when switching domains
- Requires (minimally) a partial parser
- Assumes semantic category(ies) for each slot are known, and all potential slot fillers can be tested w.r.t. them
- Produces very **high-precision** IE system

Learned terrorism patterns

- <victim> was murdered
- <perpetrator> bombed
- <perpetrator> attempted to kill
- was aimed at <target>

Bad patterns are possible

- took <victim>

They took 2-year-old Gilberto Molasco, son of Patricio Rodriquez, and 17-year-old Andres Argueta, son of Ernesto Argueta.

Natural disasters patterns

- <subject> = disaster-event (earthquake) registered (active)
- registered (active) <direct obj> = magnitude
  - Yesterday's earthquake registered 6.9 on the Richter scale.

- measuring (gerund) <direct obj> = magnitude
  - measuring 6.9 …

- aid (noun)…in/to/for (prep) <obj> = disaster-event-location/ victim
  - …sending medical aid to Afghanistan…
  - …sending medical aid to earthquake victims

Advantages and Disadvantages

- Learns bad patterns as well as good patterns
  - Too general (e.g. triggered by "is" or "are" or by verbs not tied to the domain)
  - Too specific
  - Just plain wrong
    - Parsing errors
    - Target NPs occur in a prepositional phrase and Autoslog can't determine the trigger (e.g. is it the preceding verb or the preceding NP?)
- Requires that a person review the proposed extraction patterns, discarding bad ones
- No computational linguist needed (?)
- Reduced human effort from 1200-1500 hours to ~4.5 hours

Results

- 1500 texts, 1258 answer keys
- 4780 slots (6 types)
- Autoslog generated 1237 patterns
- After human filtering: 450 patterns
- Compare to manually built patterns

System/Data Set	Recall	Precision	F-measure
Manual/TST3	46	56	50.51
Autoslog/TST3	43	56	48.65
Manual/TST4	44	40	41.90
Autoslog/TST4	39	45	41.79

Autoslog-TS

- Largely unsupervised
- Two sets of documents: relevant, not relevant
- Apply pattern templates to extract every NP in the texts
- Compute *relevance rate* for each pattern *i* :

Pr (relevant text | text contains i) =
freq of *i* in relevant texts / frequency of *i* in corpus

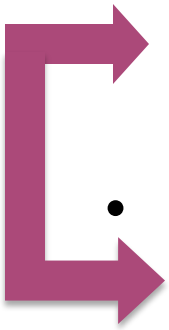
- Sort patterns according to relevance rate and frequency
relevance rate * log (freq)

Autoslog-TS

- Human review of learned patterns is still required
- Also requires, for each pattern, the manual labeling of the semantic category of the extracted slot filler

Information extraction

- **Introduction**
  - Task definition
  - Evaluation
  - IE system architecture
- **Acquiring extraction patterns**
  - Manually defined patterns
  - Learning approaches
    - Semi-automatic methods for extraction from unstructured text
    - Fully automatic methods for extraction from structured text
  - Semi-structured text
- **Named entity detection**
- **Sequence-tagging methods for IE**



Information extraction

- **Introduction**
  - Task definition
  - Evaluation
  - IE system architecture
- **Acquiring extraction patterns**
  - Manually defined patterns
  - Learning approaches
    - Semi-automatic methods for extraction from unstructured text
    - Fully automatic methods for extraction from structured text
  - ~~Semi-structured text~~
- **Named entity detection**
- **Sequence-tagging methods for IE**