Event Information Extraction

Li fang

Dept.of Computer Science & Engineering

Contents

- Event and its representation
- Template based Event Information Extraction
- Sentence-based event extraction
- Temporal expression extraction

What is an event?

- Event: who did what to whom when and where.
  - ✓ An event is a specific occurrence involving participants.
  - ✓ An event is something that happens.
  - ✓ An event can frequently be described as a change of state.
- Event Representations
  - ✓ Template in MUC
  - ✓ Verb in ACE
  - ✓ Keyword in reality

Aspects of Event Extraction

- Source Granularity:

1. Sentence-based
2. Single-document
3. Multi-document

- Document Assumption:

1. single-event
2. multi-events

Event Extraction

- Template-based event information extraction.
- ✓ One document represent one event.

- Verb represent one event
- ✓ One sentence represent one event. One document has many events.

Template & slot

- Event: described as a template
- Template: set of slots
- Slot: labeled to indicate the kind of information about the event:

a) An attribute of an entity,
b) A relationship between two or more entities,
c) An event

For example

- Vehicle Launch Events

Content:<launch_event>

Vehicle_info: <vehicle_info>

Payload_info: <payload_info>

Launch_date: <time>

Launch_site: <location>

Mission_type: *{military, civilian}*

Mission-function:*{test,deploy,retrieval*

Mission-status: *{succeeded,failed, in_progree, scheduled}*

…..

A rocket carrying a television satellite exploded seconds after launch Wednesday, dealing a potential blow to Rupert Murdoch's ambitions to offer satellite programming in Latin America.

…

Slot filling

slots

Filling Template

- Four kinds of slots in the template:

1. **Set fill**: by selection from a prespecified list of categories defined in the fill rules for a given slot.
2. **String fill**:with *an exact copy of a text string* from the input.
3. **Normalized fill**:with *a text string* that is converted to *a canonical form* in accordance with the filled rules for a given slot.
4. **Index fill** (pointer): with the index of an object <>

Slots and Patterns

- **Specify** an item to extract for a slot using a regular expression pattern.
  - Price pattern: "\b\$\d+(\.\d{2})?\b"
- **Require** preceding (pre-filler) and succeeding (post-filler) pattern to identify proper context.
  - Amazon list price:
    - Pre-filler pattern: "List Price: "
    - Filler pattern: ".+"
    - Post-filler pattern: ""

Slots and Patterns （cont.）

- NLP helps
  - Part-of-speech (POS) tagging
    - Mark each word as a noun, verb, preposition, etc.
  - Syntactic parsing
    - Identify phrases: NP, VP, PP
  - Semantic word categories (e.g. from WordNet)
    - **KILL**: kill, murder, assassinate, strangle, suffocate
- Extraction patterns can use POS or phrase tags.
  - Crime victim:
    - Prefiller: [POS: V, Hypernym: KILL]
    - Filler: [Phrase: NP]

How to define patterns to fill in the slots

- Supervised method
- Semi-supervised method
- Unsupervised method

Supervised method

- Input: a training text
- Output: a pattern or a rule

Training text: ... public buildings were bombed and a car-bomb was…

Filler of the slot 'Phys_Target' in the answer key template:"public buildings"

Pattern:

Name: target-subject-passive-verb-bombed

Trigger: bombed

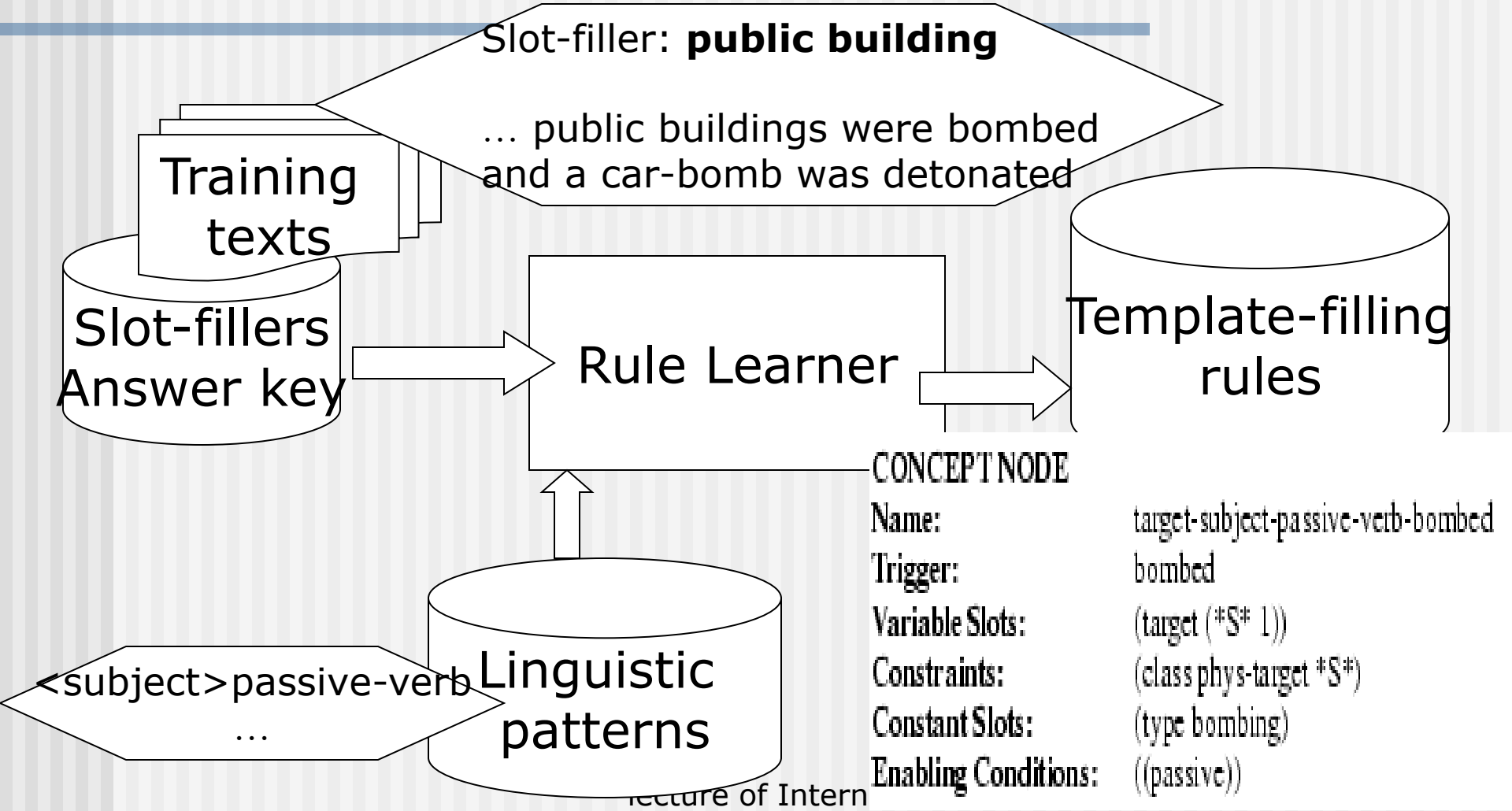
Slot: Phys_Target *Subject*

Slot-constraints: class phys-target *Subject*

Constant-slots: type bombing

Enabled-by: passive

Workflow


Slot-filler: **public building**
… public buildings were bombed and a car-bomb was detonated
Training texts
Slot-fillers Answer key
Rule Learner
Template-filling rules
CONCEPT NODE
Name: target-subject-passive-verb-bombed
Trigger: bombed
Variable Slots: (target (*S* 1))
Constraints: (class phys-target *S*)
Constant Slots: (type bombing)
Enabling Conditions: ((passive))
<subject>passive-verb
…
Linguistic patterns

- The bracketed item is a slot name
- The underlined word is the triggering word

Linguistic Pattern	Exa
<subject> passive-verb	<victim> was murdered
<subject> active-verb	<perpetrator> bombed
<subject> verb infinitive	<perpetrator> attempted to kill
<subject> auxiliary noun	<victim> was victim
passive-verb <dobj>[1]	killed <victim>
active-verb <dobj>	bombed <target>
infinitive <dobj>	to kill <victim>
verb infinitive <dobj>	threatened to attack <target>
gerund <dobj>	killing <victim>
noun auxiliary <dobj>	fatality was <victim>
noun prep <np>	bomb against <target>
active-verb prep <np>	killed with <instrument>

Unsupervised approaches

- Bootstrapping
- Duality/Density principle for validation of each iteration
- Input:
- ✓ Unclassified and unannotated corpus
- ✓ Seed patterns about an event e.g.
- ✓ subject(company)-verb(appoint)-object(person)

Preprocessing

- Full parser for detecting:
- *Subject*: a semantic subject
- *Verb*
- *Object*
- *A phrase* which refers to the object or the subject, e.g: *object complement* like, *company named John Smith* president.

- NE recognition

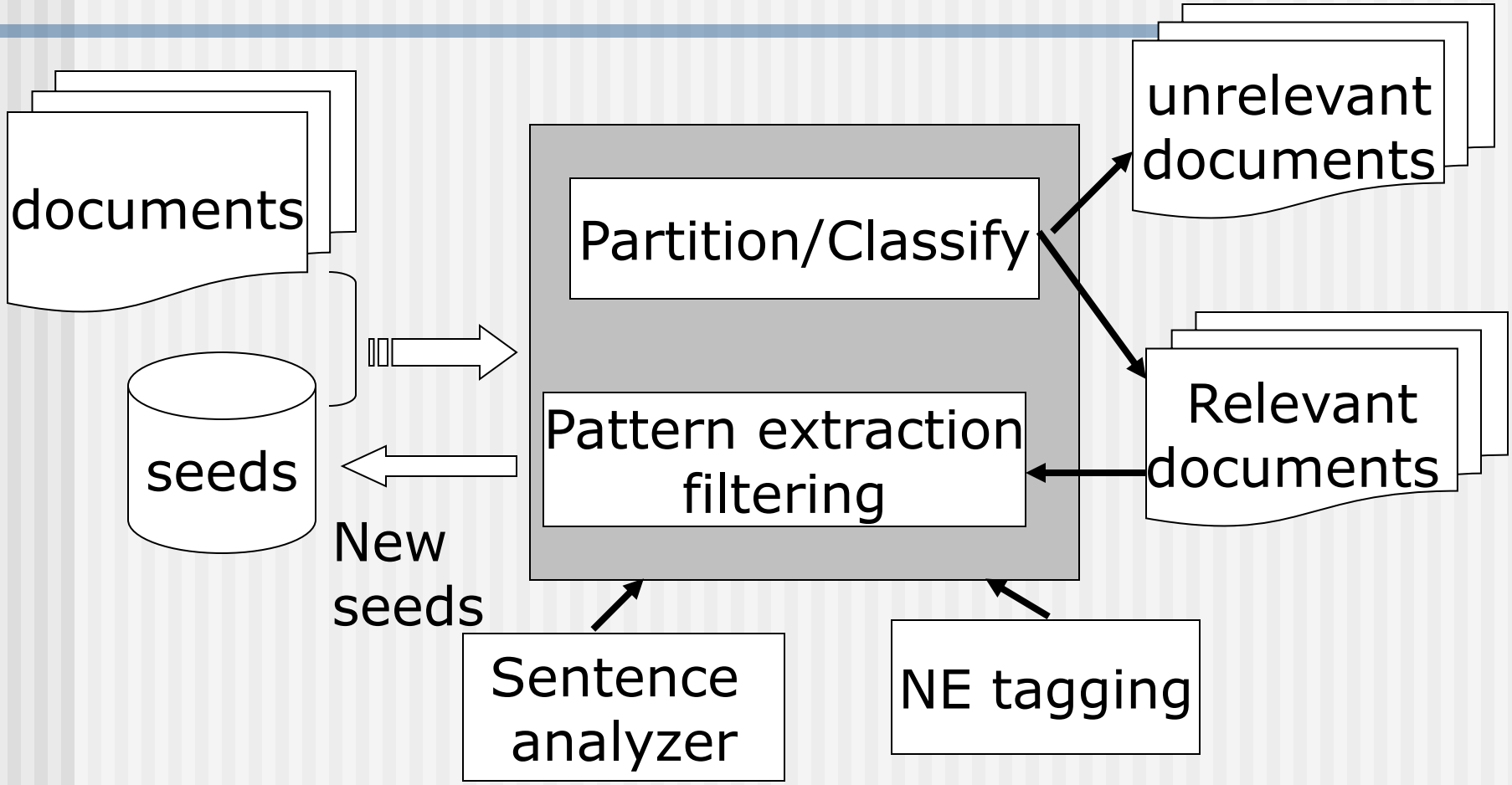
Duality/Density Principle (definition)

- Density:
- Relevant documents contain more relevant patterns.
- Duality:
- Documents that are relevant to the scenario will necessarily contain good patterns.
- Good patterns are indicators of relevant documents.

Unsupervised Learning (Duality)

- Aim: Look for linguistic patterns which appear with a relatively high frequency in relevant documents.
- the set of relevant documents is not known, they have to be found as part of the discovery process
  - Relevant documents include good patterns.
  - Good patterns indicate relevant documents.
  - Both-> circularity -> acquired in tandem

WorkFlow of the system



Preprocessing (1)

- **Name recognition** marks all instances of names of people, companies, and locations -> replaced with the class name (C-Person, C-Company,...)
- a parser extracts all the clauses:
  - Build a tuple, consisting of the basic syntactic constituents (**subject, verb, object**)
  - different clause structures (passive…) are normalized.

Preprocessing (2)

- Each tuple is reduced to a set of pairs, e.g.
  - verb-object
  - subject-object
- Each pair is used as a generalized pattern
- Relevant pairs be used to gather the set of words for the missing roles
  - e.g. verbs that occur with a relevant subject-object pair: ”company {hire/fire/expel/...} person”

Discovery procedure (1)

- Input:
  - the training corpus (not annotated, not even classified)
  - a small set of seed patterns (regarding the scenario)
- starting with this seed, the system automatically performs a *repeated*, *automatic* expansion of the pattern set.

Discovery procedure (2)

1. Pattern set → divide the corpus U into relevant document set R, and a non-relevant documents U – R
   - a document is relevant, if it contains at least one instance of one of the patterns
2. Search for new candidate patterns:
   - automatically generate a set of candidate patterns, one for each clause.
   - **rank patterns** by the degree to which their distribution is correlated with document relevance

Discovery procedure (3)

3. Add the highest ranking pattern to the pattern set
   - optionally present the pattern to the user for review
4. The new pattern set → to induce a new split of the corpus into relevant and non-relevant documents.
5. Repeat the procedure (from step 1) until some iteration limit is reached.

Example

Management succession scenario

- two initial seed patterns
  - C-Company C-Appoint C-Person
  - C-Person C-Resign
- C-Company, C-Person: semantic classes.
- C-Appoint = {appoint, elect, promote, name, nominate}
- C-Resign = {resign, depart, quit}

Pattern Selection: 满足密度要求

- Pattern (p) selection: consider those candidates patterns, p meets **Density criterion**:

$$\frac{|H \cap R|}{|H \cap U|} \gg \frac{|R|}{|U|}$$

where $H = H(p)$ is the set of documents where p hits.

A pattern appear more frequent in R, less frequent in U. the pattern is good.

U: universe documents

R: relevant documents

P: candidate patterns

Pattern Discovery

Relevant docu
contains gooc
patterns

- Computer the score of each p:

$$L(p) = P_c(p) \cdot \log |H \cap R|$$

where R denotes the relevant subset, and $H = H(p)$ the documents matching p, as above, and $P_c(p) = \frac{|H \cap R|}{|H|}$ is the conditional probability of relevance.

- Filter uninformative and raw patterns:

$|H \cap U| > \alpha |U|$ as uninformative, $|H \cap R| < \beta$ as noise.

Bootstrapping for Patterns and Relevant Documents

$$Prec^{i+1}(p) = \frac{1}{|H(p)|} \cdot \sum_{d \in H(p)} Rel^{i}(d)$$

On iteration number i+1, each pattern p is assigned a precision measure, based on the relevance of the documents it matches

$$Prec^{i+1}(K) = \frac{1}{|H(K)|} \cdot \sum_{d \in H(K)} Rel^{i}(d)$$

If k is a classifier consisting of a set of patterns. H(k): as the set of documents where all of patterns p k match, and the cumulative precision of K is:

$$Rel^{i+1}(d) = \max\left(Rel^{i}(d), Prec^{i+1}(K_d)\right)$$

For each document d, which is matched by a subset of currently accepted patterns Kd, New relevance score of the document d as:

Conclusion for the unsupervised method

- Advantages
  - Unsupervised (unannotated, unclassified corpus)
  - Multi-slot template filler rules
- Disadvantages
- only subject-verb-object patterns, local patterns are ignored
- no generalization of pattern rules
- collocations are not taken into account.

Sentence based Event Extraction (concepts)

- The event extent and event trigger.

  Event extent is *a sentence* within which a taggable event is described. Event trigger is *the word* that most clearly expresses its occurrence.
- The *event participants*: *entities* that are involved in that event.
- Other entities and values within the scope of an event that are not properly participants, but should be understood as part of the event, such *entities* are regarded as *attributes* of the event.
- Event arguments include event *participants* and event *attributes*.

For example

A bomb exploded yesterday in a marketplace in Lahore. The attack killed 7 and injured 20.

Event: Trigger	Type	Subtype	Modality	Arguments
V1: *exploded* V1: *attack*	Conflict	Attack	T: Past P: Positive G: Specific M: Asserted	Instrument: *bomb* Time-Holds: *yesterday* Target: *marketplace* Place: *Lahore* Target: *20* Target: *7*
V2: *killed*	Life	Die	T: Past P: Positive S: Specific M: Asserted	Victim: *7*
V3: *injured*	Life	Injure	T: Past P: Positive S: Specific M: Asserted	Victim: *20*

Single word that best expresses the event

Event mention extent is entire sentence containing trigger

Tense

Polarity

Genericity

Modality

Examples of ACE Event types and subtypes

Life	*Movement*	*Transaction*	*Personnel*	*Conflict*	*Contact*	*Business*
Be Born	Transport Person	Transfer Ownership	Start Position	Attack	Meet	Start-Org
Marry	Transport Artifact	Transfer Money	End Position	Demonstrate	Communicate	End-Org
Divorce			Nominate			Declare Bankruptcy
Injure			Elect			Merge Org
Die						

Justice					
Arrest	Sentence	Indict	Extradite	Charge	Execute
Release	Jail	Try	Acquit	Parole	Pardon
Hold Hearing	Fine	Sue	Convict	Appeal	

Steps of Event Extraction

- **Anchor identification**: finding event triggers in text and assigning them an event type.
- **Argument identification**: determining entity mentions, timexes and values.
- **Attribute assignment**: determining the value of the modality, polarity and so on.
- **Event coreference**: determining which event mentions refer to the same event.

Event Extraction (example)

TEXT: *Three **murders** occurred in France today, including the senseless **slaying** of Bob Cole and the **assassination** of Joe Westbrook. Bob was on his way home when he was **attacked**…*

- Finding event triggers and assign their types.

Event Extraction (example)

- name identification, entity mention classification and co-reference

Entity(Timex) mention	head word	Entity ID	Entity type
0001-1-1	France	0001-1	GPE
0001-T1-1	Today	0001-T1	Timex
0001-2-1	Bob Cole	0001-2	PER
0001-3-1	Joe Westbrook	0001-3	PER
0001-2-2	Bob	0001-2	PER
0001-2-3	He	0001-2	PER

Event Extraction (example)

- **Argument identification**: There are three Die events, which share the same Place and Time roles, with different Victim roles. There is one Attack event sharing the same Place and Time roles with the Die events.

TEXT: *Three **murders** occurred in **France today**, including the senseless **slaying** of **Bob Cole** and the **assassination** of **Joe Westbrook**. **Bob** was on his way home when **he** was **attacked**…*

Event type	Trigger	Role		
		Place	Victim	Time
Die	murder	0001-1-1		0001-T1-1
Die	death	0001-1-1	0001-2-1	0001-T1-1
Die	killing	0001-1-1	0001-3-1	0001-T1-1
Event type	**Trigger**	**Role**		
		Place	**Target**	**Time**
Attack	attack	0001-1-1	0001-2-3	0001-T1-1

Methods of event extraction

Reference: David Ahn. The stages of event extraction

- Anchor identification: two stages

1. a binary classifier: a word is a trigger word or not.
2. A multi-class classifier: the event type

- Argument identification

1. a single multi-class classification task
2. a separate multi-class classifier for each event type.

- Attribute assignment

1. binary classification for genericity, modality,polarity
2. tense is a multi-class task.

Features for event anchors (cont.)

- Lexical features
- WordNet features
- Left context (3 words): lowercase, POS tag
- Right context (3 words): lowercase, POS tag
- Dependency features
- Related entity features

Features for Argument identification (cont.)

- Anchor word of event mention: full, lowercase, POS tag, depth in parse tree
- Event type of event mention
- Constituent head word of entity mention: full, lowercase, POS tag, depth in parse tree
- Determiner of entity mention, if any
- Entity type and mention type(name, pronoun, other NP) of entity mention
- Dependency path between anchor word and constituent head word of entity mention, expressed as a sequence of labels, of words and POS tags.

Summarization on Event Extraction

- Events are predefined.
- ✓ Event as a Template to extract from a document.
- ✓ Verb as an event mostly use classification methods to extract from sentences.
- Events are not predefined
- ✓ Clustering, keywords extraction

Temporal Expressions Extraction

- Absolute temporal expressions
- Relative temporal expressions
- Durations

Absolute	Relative	Durations
April 24, 1916	yesterday	four hours
The summer of '77	next semester	three weeks
10:15 AM	two weeks from yesterday	six days
The 3rd quarter of 2006	last quarter	the last three quarters

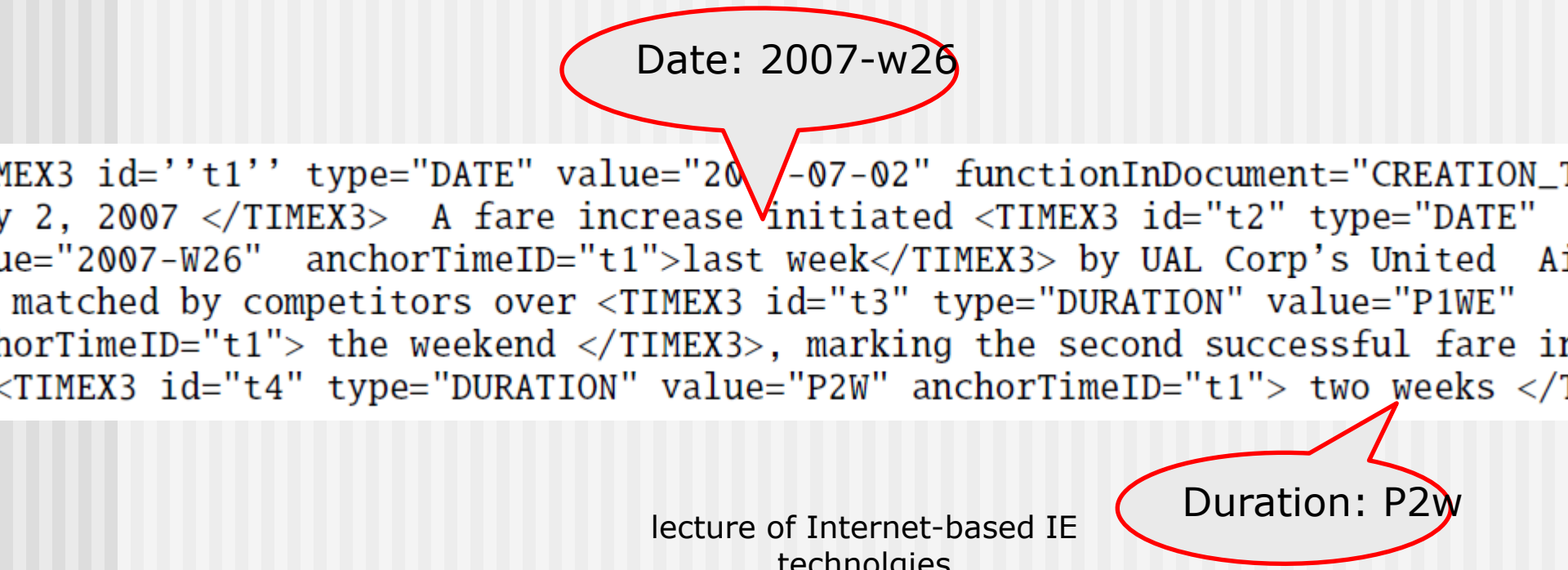
Lexical Triggers

- Nouns, proper nouns, adjectives and adverbs.

Category	Examples
Noun	*morning, noon, night, winter, dusk, dawn*
Proper Noun	*January, Monday, Ides, Easter, Rosh Hashana, Ramadan, Tet*
Adjective	*recent, past, annual, former*
Adverb	*hourly, daily, monthly, yearly*

Temporal Normalization

- The process of mapping a temporal expression to either a specific point in time or a duration.



Temporal Expression Extraction

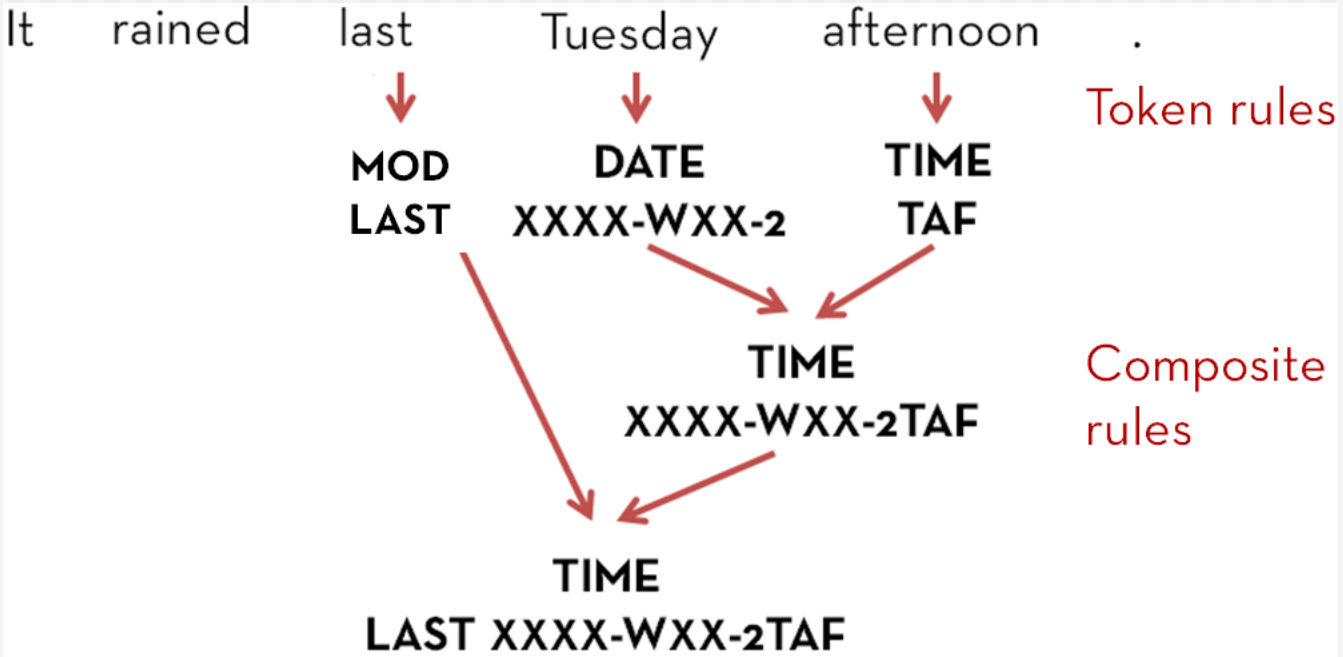
- Sequence-labeling approaches

A fare increase initiated last week by UAL Corp's...
O O O O B I O O O

Feature	Explanation
Token	The target token to be labeled
Tokens in window	Bag of tokens in the window around a target
Shape	Character shape features
POS	Parts of speech of target and window words
Chunk tags	Base-phrase chunk tag for target and words in a window
Lexical triggers	Presence in a list of temporal terms

Temporal Expression Extraction (cont.)

- Rule based method



Summarization

- Template_based event extraction
- Sentence based event extraction
- Temporal Expression Extraction

References

- Automatic Content Extraction 2008 Evaluation Plan
- David Ahn, " the stages of event extraction"
- Shasha Liao, Ralph Grishman,"Using Document level Cross-event inference to improve event extraction"

Discussion Topics

- If the event is not predefined, how to implement an event extraction system?

Chambers, N. and Jurafsky, D. Template-based information extraction without the templates. In ACL 2011.