Information extraction from text

Part 2

Course organization

- Conversion of exercise points to the final points:
  - an exercise point = 1.5 final points
  - 20 exercise points give the maximum of 30 final points
- Tomorrow, lectures and exercises in A318?

2

Examples of IE systems

- FASTUS (Finite State Automata-based Text Understanding System), SRI International
- CIRCUS, University of Massachusetts, Amherst

3

FASTUS

4

Lexical analysis

- John Smith, 47, was named president of ABC Corp. He replaces Mike Jones.
- Lexical analysis (using dictionary etc.):
  - John: proper name (known first name -> person)
  - Smith: unknown capitalized word
  - 47: number
  - was: auxiliary verb
  - named: verb
  - president: noun

5

Name recognition

- Name recognition
  - John Smith: person
  - ABC Corp: company
  - Mike Jones: person
- also other special forms
  - dates
  - currencies, prices
  - distancies, measurements

6

Triggering

- Trigger words are searched for
- sentences containing trigger words are relevant
- at least one trigger word for each pattern of interest
  - the least frequent words required by the pattern, e.g. in
    - take <HumanTarget> hostage
  - "hostage" rather than "take" is the trigger word

7

Triggering

- person names are trigger words for the rest of the text
  - Gilda Flores was assassinated yesterday.
  - Gilda Flores was a member of the PSD party of Guatemala.
- full names are searched for
  - subsequent references to surnames can be linked to corresponding full names

8

Basic phrases

- Basic syntactic analysis
  - John Smith: person (also noun group)
  - 47: number
  - was named: verb group
  - president: noun group
  - of: preposition
  - ABC Corp: company

9

Identifying noun groups

- Noun groups are recognized by a 37-state nondeterministic finite state automaton
- examples:
  - approximately 5 kg
  - more than 30 peasants
  - the newly elected president
  - the largest leftist political force
  - a government and military reaction

10

Identifying verb groups

- Verb groups are recognized by an 18-state nondeterministic finite state automaton
- verb groups are tagged as
  - Active, Passive, Active/Passive, Gerund, Infinitive
  - Active/Passive, if local ambiguity
    - Several men kidnapped the mayor today.
    - Several men kidnapped yesterday were released today.

11

Other constituents

- Certain relevant predicate adjectives ("dead", "responsible") and adverbs are recognized
- most adverbs and predicate adjectives and many other classes of words are ignored
- unknown words are ignored unless they occur in a context that could indicate they are surnames

12

Complex phrases

- "advanced" syntactic analysis
  - John Smith, 47: noun group
  - was named: verb group
  - president of ABC Corp: noun group

13

Complex phrases

- complex noun groups and verb groups are recognized
  - only phrases that can be recognized reliably using domain-independent syntactic information
- e.g.
  - attachment of appositives to their head noun group
  - attachment of "of" and "for"
  - noun group conjunction

14

Domain event patterns

- Domain phase
  - John Smith, 47, was named president of ABC Corp : domain event
  - one or more template objects created

15

Domain event patterns

- The input to domain event recognition phase is a list of basic and complex phrases in the order in which they occur
  - anything that is not included in a basic or complex phrase is ignored

16

Domain event patterns

- patterns for events of interest are encoded as finite-state machines
  - state transitions are effected by <head_word, phrase_type> pairs
  - "mayor-NounGroup", "kidnapped-PassiveVerbGroup", "killing-NounGroup"

17

Domain event patterns

- 95 patterns for the MUC-4 application
  - killing of <HumanTarget>
  - <GovtOfficial> accused <PerpOrg>
  - bomb was placed by <Perp> on <PhysicalTarget>
  - <Perp> attacked <HumanTarget>'s <PhysicalTarget> with <Device>
  - <HumanTarget> was injured

18

"Pseudo-syntax" analysis

- The material between the end of the subject noun group and the beginning of the main verb group must be read over
  - Subject (Preposition NounGroup)* VerbGroup
    - here (Preposition NounGroup)* does not produce anything
  - Subject Relpro (NounGroup | Other)* VerbGroup (NounGroup | Other)* VerbGroup

19

"Pseudo-syntax" analysis

- There is another pattern for capturing the content encoded in relative clauses
  - Subject Relpro (NounGroup | Other)* VerbGroup
- since the finite-state mechanism is nondeterministic, the full content can be extracted from the sentence
  - "The mayor, who was kidnapped yesterday, was found dead today."

20

Domain event patterns

- Domain phase
  - <Person> was named <Position> of <Organization>
  - John Smith, 47, was named president of ABC Corp : domain event
  - one or more templates created

21

Template created for the transition event

START	Person	---
	Position	president
	Organization	ABC Corp
END	Person	John Smith
	Position	president
	Organization	ABC Corp

22

Domain event patterns

- The sentence "He replaces Mike Jones." is analyzed respectively
- the coreference phase identifies "John Smith" as the referent of "he"
- a second template is formed

23

A second template created

START	Person	Mike Jones
	Position	----
	Organization	----
END	Person	John Smith
	Position	----
	Organization	----

24

Merging

- The two templates do not appropriately summarize the information in the text
- a discourse-level relationship has to be captured -> merging phase
- when a new template is created, the merger attempts to unify it with templates that precede it

25

Merging

START	Person	Mike Jones
	Position	president
	Organization	ABC Corp
END	Person	John Smith
	Position	president
	Organization	ABC Corp

26

FASTUS

- advantages
  - conceptually simple: a set of cascaded finite-state automata
  - the basic system is relatively small
    - dictionary is potentially very large
  - effective
    - in MUC-4: recall 44%, precision 55%

27

CIRCUS

28

Syntax processing in CIRCUS

- stack-oriented syntax analysis
- no parse tree is produced
- uses local syntactic knowledge to recognize noun phrases, prepositional phrases and verb phrases
- the constituents are stored in global buffers that track the subject, verb, direct object, indirect object and prepositional phrases of the sentence

29

Syntax processing

- To process the sentence that begins
  - "John brought..."
- CIRCUS scans the sentence from left to right and
- uses syntactic predictions to assign words and phrases to syntactic constituents
- initially, the stack contains a single prediction: the hypothesis for a subject of a sentence

30

Syntax processing

- when CIRCUS sees the word "John", it
  - accesses its part-of-speech lexicon, finds that "John" is a proper noun
  - loads the standard set of syntactic predictions associated with proper nouns onto the stack
  - recognizes "John" as a noun phrase
  - because the presence of a NP satisfies the initial prediction for a subject, CIRCUS places "John" in the subject buffer (*S*) and pops the satisfied syntactic prediction from the stack

31

Syntax processing

- Next, CIRCUS processes the word "brought", finds that it is a verb, and assigns it to the verb buffer (*V*)
- in addition, the current stack contains the syntactic expectations associated with "brought": (the following constituent is…)
  - a direct object
  - a direct object followed by a "to" PP
  - a "to" PP followed by a direct object
  - an indirect object followed by a direct object

32

For instance,

- John brought a cake.
- John brought a cake to the party.
- John brought to the party a cake.
  - this is actually ungrammatical, but it has a meaning…
- John brought Mary a cake.

33

Syntactic expectations associated with "brought"

- 1. if NP, NP -> *DO*;
  - predict: if EndOfSentence, NIL -> *IO*
- 2. if NP, NP -> *DO*;
  - predict: if PP(to), PP -> *PP*, NIL -> *IO*
- 3. if PP(to), PP -> *PP*;
  - predict: if NP, NP -> *DO*
- 4. if NP, NP -> *IO*;
  - predict: if NP, NP -> *DO*

34

Filling template slots

- As soon as CIRCUS recognizes a syntactic constituent, that constituent is made available to the mechanisms performing slot-filling (semantics)
- whenever a syntactic constituent becomes available in one of the global buffers, any active concept node that expects a slot filler from that buffer is examined

35

Filling template slots

- The slot is filled if the constituent satisfies the slot's semantic constraints
- both hard and soft constraints
  - a hard constraint must be satisfied
  - a soft constraint defines a preference for a slot filler

36

Filling template slots

- e.g. a concept node PTRANS
- sentence: John brought Mary to Manhattan
- PTRANS
  - Actor = "John"
  - Object = "Mary"
  - Destination = "Manhattan"

37

Filling template slots

- The concept node definition indicates the mapping between surface constituents and concept node slots:
  - subject -> Actor
  - direct object -> Object
  - prepositional phrase or indirect object -> Destination

38

Filling template slots

- A set of enabling conditions: describe the linguistic context in which the concept node should be triggered
  - PTRANS concept node should be triggered by "brought" only when the verb occurs in an active construction
  - a different concept node would be needed to handle a passive sentence construction

39

Hard and soft constraints

- soft constraints
  - the Actor should be animate
  - the Object should be a physical object
  - the Destination should be a location
- hard constraint
  - the prepositional phrase filling the Destination slot must begin with the preposition "to"

40

Filling template slots

- After "John brought", the Actor slot is filled by "John"
  - "John" is the subject of the sentence
  - the entry of "John" in the lexicon indicates that "John" is animate
- when a concept node satisfies certain instantiation criteria, it is freezed with its assigned slot fillers -> it becomes part of the semantic presentation of the sentence

41

Handling embedded clauses

- When sentences become more complicated, CIRCUS has to partition the stack processing in a way that recognizes embedded syntactic structures as well as conceptual dependencies

42

Handling embedded clauses

- John asked Bill to eat the leftovers.
  - "Bill" is the subject of "eat"
- That's the gentleman that the woman invited to go to the show.
  - "gentleman" is the direct object of "invited" and the subject of "go"
- That's the gentleman that the woman declined to go to the show with.

43

Handling embedded clauses

- We view the stack of syntactic predictions as a single control kernel whose expectations and binding instructions change in response to specific lexical items as we move through the sentence
- when we come to a subordinate clause, the top-level kernel creates a subkernel that takes over to process the inferior clause -> a new parsing environment

44

Knowledge needed for analysis

- Syntactic processing
  - for each part of speech: a set of syntactic predictions
  - for each word in the lexicon: which parts of speech are associated with the word
  - disambiguation routines to handle part-of-speech ambiguities

45

Knowledge needed for analysis

- Semantic processing
  - a set of semantic concept node definitions to extract information from a sentence
    - enabling conditions
    - a mapping from syntactic buffers to slots
    - hard slot constraints
    - soft slot constraints in the form of semantic features

46

Knowledge needed for analysis

- concept node definitions have to be explicitly linked to the lexical items that trigger the concept node
- each noun and adjective in the lexicon has to be described in terms of one or more semantic features
  - it is possible to test whether the word satisfies a slot's constraints
- disambiguation routines for word sense disambiguation

47

Concept node classes

- Concept node definitions can be categorized into the following taxonomy of concept node types
  - verb-triggered (active, passive, active-or-passive)
  - noun-triggered
  - adjective-triggered
  - gerund-triggered
  - threat and attempt concept nodes

48

Active-verb triggered concept nodes

- A concept node triggered by a specific verb in an active voice
- typically a prediction for finding the ACTOR in *S* and the VICTIM or PHYSICAL-TARGET in *DO*
- for all verbs important to the domain
  - kidnap, kill, murder, bomb, detonate, massacre, ...

49

Concept node definition for kidnapping verbs

- Concept node
  - name: $KIDNAP$
  - slot-constraints:
    - class organization *S*
    - class terrorist *S*
    - class proper-name *S*
    - class human *S*
    - class human *DO*
    - class proper-name *DO*

50

Concept node definition for kidnapping verbs, cont.

- variable-slots
  - ACTOR *S*
  - VICTIM *DO*
- constant-slots:
  - type kidnapping
- enabled-by:
  - active
  - not in reduced-relative

51

Is the verb active?

- Function **active** tests
  - the verb is in past tense
  - any auxiliary preceding the verb is of the correct form (indicating active, not passive)
  - the verb is not in the infinitive form
  - the verb is not preceding by "being"
  - the sentence is not describing threat or attempt
  - no negation, no future

52

Passive verb-triggered concept nodes

- Almost every verb that has a concept node definition for its active form should also have a concept node definition for its passive form
- these typically predict for finding the ACTOR in a by-*PP* and the VICTIM or PHYSICAL-TARGET in *S*

53

Concept node definition for killing verbs in passive

- Concept node
  - name $KILL-PASS-1$
  - slot-constraints:
    - class organization *PP*
    - class terrorist *PP*
    - class proper-name *PP*
    - class human *PP*
    - class human *S*
    - class proper-name *S*

54

Concept node definition for killing verbs in passive

- variable-slots:
  - ACTOR *PP* is-preposition "by"?
  - VICTIM *S*
- constant-slots:
  - type murder
- enabled-by:
  - passive
  - subject is not "no one"

55

Fillers for several slots

- "Castellar was killed by ELN guerillas with a knife"
- a separate concept node for each PP
- Concept node
  - name $KILL-PASS-2$
  - slot-constraints:
    - class human *S*
    - class proper-name *S*
    - class weapon *PP*

56

Fillers for several slots

- variable-slots:
  - INSTR *PP* is-preposition "by" and "with"?
  - VICTIM *S*
- constant-slots:
  - type murder
- enabled-by:
  - passive
  - subject is not "no one"

57

Noun-triggered concept nodes

- The following concept node definition is triggered by nouns
  - massacre, murder, death, murderer, assassination, killing, and burial
- it looks for the Victim in an of-PP

58

Concept node definition for murder nouns

- Concept node
  - name $MURDER$
  - slot-constraints:
    - class human *PP*
    - class proper-name *PP*
  - variable-slots:
    - VICTIM *PP*, preposition "of" follows triggering word?
  - constant-slots: type murder
  - enabled-by: noun-triggered, not-threat

59

Adjective-triggered concept nodes

- Sometimes a verb is too general to make a good trigger
  - "Castellar was found dead."
- it may be easier to use an adjective to trigger a concept node and check for the presence of specific verbs (in ENABLED-BY)

60

Other concept nodes

- Gerund-triggered concept nodes
  - for important gerunds
    - killing, destroying, damaging,…
- Threat and attempt concept nodes
  - require enabling conditions that check both the specific event (e.g. murder, attack, kidnapping) and indications that the event is a threat or attempt
  - "The terrorists intended to storm the embassy."

61

Defining new concept nodes

- 3 steps to defining a concept node for a new example
  - 1. Look for an existing concept node that extracts slots from the correct buffers and has enabling conditions that will be satisfied by the current example.
  - 2. If one exists, add the name of the existing concept node to the definition of the triggering word.

62

Defining new concept nodes

- 3. Otherwise, create a new concept node definition by modifying an existing one to handle the new example
  - usually specializing an existing, more general concept

63