Information extraction from text

Spring 2002, Part 1
Helena Ahonen-Myka

Course organization

- Lectures: 28.1., 25.2., 26.2., 18.3.
  - 10-12, 13-15 (Helena Ahonen-Myka)
- Exercise sessions: 25.2., 26.2., 18.3.
  - 15.00-17 (Reeta Kuuskoski)
- exercises given each week
  - everybody tells a URL, where the solutions appear
  - deadline each week on Thursday midnight

Course organization

- Requirements
  - lectures and exercise sessions are voluntary
  - from the weekly exercises, one needs to get 50% of the points
    - each exercise gives 1-2 points
    - 2 exercises/week
- Exam 27.3. (16-20 Auditorio)
- Exam: max 30 pts; exercises: max 30 pts

Overview

- 1. General architecture of information extraction (IE) systems
- 2. Building blocks of IE systems
- 3. Learning approaches
- 4. Other related applications and approaches: IE on the web, question answering systems, (news) event detection and tracking

1. General architecture of IE systems

- What is our task?
- IE compared to other related fields
- General architecture and process
- More detailed view of the stages (example)
- Some design issues

Reference

- The following is largely based on
  - Ralph Grishman: Information extraction: Techniques and Challenges. In Information Extraction, a multidisciplinary approach to an emerging information technology. Lecture Notes in AI, Springer-Verlag, 1997.

Task

- "Information extraction involves the creation of a structured representation (such as a database) of selected information drawn from the text"

7

Example: terrorist events

19 March - A bomb went off this morning near a power tower in San Salvador leaving a large part of the city without energy, but no casualties have been reported. According to unofficial sources, the bomb - allegedly detonated by urban guerrilla commandos - blew up a power tower in the northwestern part of San Salvador at 0650 (1250 GMT).

8

Example: terrorist events

Incident type	bombing
Date	March 19
Location	El Salvador: San Salvador (city)
Perpetrator	urban guerilla commandos
Physical target	power tower
Human target	-
Effect on physical target	destroyed
Effect on human target	no injury or death
Instrument	bomb

9

Example: terrorist events

- A document collection is given
- For each document, decide if the document is about terrorist event
- For each terrorist event, determine
  - type of attack
  - date
  - location, etc.
- = fill in a template (~database record)

10

Other examples

- International joint ventures
  - arguments to be found: partners, the new venture, its product or service, etc.
- executive succession
  - who was hired/fired by which company for which position

11

Message understanding conferences (MUC)

- The development of IE systems has been shaped by a series of evaluations, the MUC conferences
- MUCs have provided IE tasks and sets of training and test data + evaluation procedures and measures
- participating projects have competed with each other but also shared ideas

12

Message understanding conferences (MUC)

- MUC-1 (1987): tactical naval operations reports (12 for training, 2 for testing)
  - 6 systems participated
- MUC-2 (1989): the same domain (105 messages for training, 25 for training)
  - 8 systems participated

13

Message understanding conferences (MUC)

- MUC-3 (1991); domain was newswire stories about terrorist attacks in nine Latin American countries
  - 1300 development texts were supplied
  - three test sets of 100 texts each
  - 15 systems participated
- MUC-4 (1992); the domain was the same
  - different task definition and corpus etc.
  - 17 systems participated

14

Message understanding conferences (MUC)

- MUC-5 (1993)
  - 2 domains: joint ventures in financial newswire stories and microelectronics products announcements
  - 2 languages (English and Japanese)
  - 17 systems participated (14 American, 1 British, 1 Canadian, 1 Japanese)
  - larger corpora

15

Message understanding conferences (MUC)

- MUC-6 (1995); domain was management succession events in financial news stories
  - several subtasks
  - 17 systems participated
- MUC-7 (1998); domain was air vehicle (airplane, satellite,...) launch reports

16

IE vs. information retrieval

- Information retrieval (IR)
  - given a user query, an IR system selects a (hopefully) relevant subset of documents from a larger set
  - the user then browses the selected documents in order to fulfil his or her information need
- IE extracts relevant information from documents -> IR and IE are complementary technologies

17

IE vs full text understanding

- In IE
  - generally only a fraction of the text is relevant
  - information is mapped into a predefined, relatively simple, rigid target representation
  - the subtle nuances of meaning and the writer's goals in writing the text are of at best secondary interest

18

IE vs full text understanding

- In text understanding
  - the aim is to make sense of the entire text
  - the target representation must accommodate the full complexities of language
  - one wants to recognize the nuances of meaning and the writer's goals

19

General architecture

- Rough view of the process:
  - the system extracts individual "facts" from the text of a document through local text analysis
  - the system integrates these facts, producing larger facts or new facts (through inference)
  - the facts are integrated and the facts are translated into the required output format

20

Architecture: more detailed view

- The individual facts are extracted by creating a set of patterns to match the possible linguistic realizations of the facts
  - it is not practical to describe these patterns directly as word sequences
  - the input is structured; various levels of constituents and relations are identified
  - the patterns are stated in terms of these constituents and relations

21

Architecture: more detailed view

- Possible stages
  - lexical analysis
    - assigning part-of-speech and other features to words/phrases through morphological analysis and dictionary lookup
  - name recognition
    - identifying names and other special lexical structures such as dates, currency expressions, etc.

22

Architecture: more detailed view

- full syntactic analysis or some form of partial parsing
  - partial parsing: e.g. identify noun groups, verb groups, head-complement structures
- task-specific patterns are used to identify the facts of interest

23

Architecture: more detailed view

- The integration phase examines and combines facts from the entire document or discourse
  - resolves relations of coreference
    - use of pronouns, multiple descriptions of the same event
  - may draw inferences from the explicitly stated facts in the document

24

Some terminology

- domain
  - general topical area (e.g. financial news)
- scenario
  - specification of the particular events or relations to be extracted (e.g. joint ventures)
- template
  - final, tabular (record) output format of IE
- template slot, argument (of a template)

25

Pattern matching and structure building

- lexical analysis
- name recognition
- syntactic structure
- scenario pattern matching
- coreference analysis
- inferencing and event merging

26

Running example

- "Sam Schwartz retired as executive vice president of the famous hot dog manufacturer, Hupplewhite Inc. He will be succeeded by Harry Himmelfarb."

27

Target templates

Event	leave job
Person	Sam Schwartz
Position	executive vice president
Company	Hupplewhite Inc.

Event	start job
Person	Harry Himmelfarb
Position	executive vice president
Company	Hupplewhite Inc

28

Lexical analysis

- The text is divided into sentences and into tokens
- each token is looked up in the dictionary to determine its possible parts-of-speech and features
  - general-purpose dictionaries
  - special dictionaries
    - major place names, major companies, common first names, company suffixes ("Inc.")

29

Name recognition

- Various types of proper names and other special forms, such as dates and currency amounts, are identified
- names appear frequently in many types of texts: identifying and classifying them simplifies further processing
- names are also important as argument values for many extraction tasks

30

Name recognition

- Names are identified by a set of patterns (regular expressions) which are stated in terms of parts-of-speech, syntactic features, and orthographic features (e.g. capitalization)

31

Name recognition

- Personal names might be identified
  - by a preceding title: Mr. Herrington Smith
  - by a common first name: Fred Smith
  - by a suffix: Snippety Smith Jr.
  - by a middle initial: Humble T. Hopp

32

Name recognition

- Company names can usually be identified by their final token(s), such as
  - Hepplewhite Inc.
  - Hepplewhite Corporation
  - Hepplewhite Associates
  - First Hepplewhite Bank
- however, some major company names ("General Motors") are problematic
  - dictionary of major companies is needed

33

Name recognition

- <name type="person"> Sam Schwartz </name> retired as executive vice president of the famous hot dog manufacturer, <name type="company"> Hupplewhite Inc.</name> He will be succeeded by <name type="person">Harry Himmelfarb</name>.

34

Name recognition

- Subproblem: identify the aliases of a name (name coreference)
  - Larry Liggett = Mr. Liggett
  - Hewlett-Packard Corp. = HP
- alias identification may also help name classification
  - "Humble Hopp reported..." (person or company?)
  - subsequent reference: "Mr. Hopp" (-> person)

35

Syntactic structure

- Identifying some aspects of syntactic structure simplifies the subsequent phase of fact extraction
  - the arguments to be extracted often correspond to noun phrases
  - the relationships often correspond to grammatical functional relations
- but: identification of the complete syntactic structure of a sentence is difficult

36

Syntactic structure

- Problems e.g. with prepositional phrases to the right of a noun
  - "I saw the man in the park with a telescope."

37

Syntactic structure

- In extraction systems, there is a great variation in the amount of syntactic structure which is explicitly identified
  - some systems do not have any separate phase of syntactic analysis
  - others attempt to build a complete parse of a sentence
  - most systems fall in between and build a series of parse fragments

38

Syntactic structure

- Systems that do partial parsing
  - build structures about which they can be quite certain, either from syntactic or semantic evidence
    - for instance, structures for noun groups (a noun + its left modifiers) and for verb groups (a verb with its auxiliaries)
    - both can be built using just local syntactic information
    - in addition, larger structures can be built if there is enough semantic information

39

Syntactic structure

- The first set of patterns labels all the basic noun groups as noun phrases (NP)
- the second set of patterns labels the verb groups

40

Syntactic structure

- <np entity="e1"> Sam Schwartz </np> <vg>retired</vg> as <np entity="e2"> executive vice president</np> of <np entity="e3">the famous hot dog manufacturer</np>, <np entity="e4"> Hupplewhite Inc.</np> <np entity="e5">He</np> <vg>will be succeeded</vg> by <np entity="e6">Harry Himmelfarb</np>.

41

Syntactic structure

- Associated with each constituent are certain features which can be tested by patterns in subsequent stages
  - for verb groups: tense (past/present/future), voice (active/passive), baseform/stem
  - for noun phrases: baseform/stem, is this phrase a name?, number (singular/plural)

42

Syntactic structure

- For each NP, the system creates a semantic entity

entity e1	type: person	name: "Sam Schwartz"
entity e2	type: position	value: "executive vice president"
entity e3	type: manufacturer	
entity e4	type: company	name:"Hupplewhite Inc."
entity e5	type: person	
entity e6	type: person	name: "Harry Himmelfarb"

43

Syntactic structure

- Semantic constraints
  - the next set of patterns build up larger noun phrase structures by attaching right modifiers
  - because of the syntactic ambiguity of right modifiers, these patterns incorporate some semantic constraints (domain specific)

44

Syntactic structure

- In our example, two patterns will recognize the appositive construction:
  - *company-description, company-name*,
- and the prepositional phrase construction:
  - *position* of *company*
- in the second pattern:
  - *position* matches any NP whose entity is of type "position"
  - *company* respectively

45

Syntactic structure

- the system includes a small semantic type hierarchy (*is-a* hierarchy)
  - the pattern matching uses the *is-a* relation, so any subtype of company (such as manufacturer) will be matched
- in the first pattern
  - *company-name*: NP of type "company" whose head is a name
  - *company-description*: NP of type "company" whose head is a common noun

46

Syntactic structure

- <np entity="e1"> Sam Schwartz </np> <vg>retired</vg> as <np entity="e2"> executive vice president of the famous hot dog manufacturer, Hupplewhite Inc.</np> <np entity="e5">He</np> <vg>will be succeeded</vg> by <np entity="e6"> Harry Himmelfarb</np>.

47

Syntactic structure

- Entities are updated as follows:

entity e1	type: person	name: "Sam Schwartz"
entity e2	type: position	value: "executive vice president"
		company: e3
entity e3	type: manufacturer	name: "Hupplewhite Inc."
entity e5	type: person	
entity e6	type: person	name: "Harry Himmelfarb"

48

Scenario pattern matching

- Role of scenario patterns is to extract the events or relationships relevant to the scenario
- in our example, there will be 2 patterns
  - *person* retires as *position*
  - *person* is succeeded by *person*
- *person* and *position* are pattern elements which match NP's with the associated type
- "retires" and "is succeeded" are pattern elements which match active and passive verb groups, respectively

49

Scenario pattern matching

- Text labeled with 2 clauses
  - clauses point to an event structure
  - event structures point to the entities
- <clause event="e7"> Sam Schwartz retired as executive vice president of the famous hot dog manufacturer, Hupplewhite Inc.</clause> <clause event="e8">He will be succeeded by Harry Himmelfarb</clause>.

50

Scenario pattern matching

entity e1	type: person	name: "Sam Schwartz"
entity e2	type: position	value: "executive vice president"
		company: e3
entity e3	type: manufacturer	name:"Hupplewhite Inc."
entity e5	type: person	
entity e6	type: person	name: "Harry Himmelfarb"
event e7	type: leave-job	person: e1 position: e2
event e8	type: succeed	person1: e6 person2: e5

51

Coreference analysis

- Task of resolving anaphoric references by pronouns and definite noun phrases
  - in our example: "he" (entity e5)
  - coreference analysis will look for the most recent previously mentioned entity of type person, and will find entity e1
  - references to e5 are changed to refer to e1 instead
- also the *is-a* hierarchy is used

52

Coreference analysis

entity e1	type: person	name: "Sam Schwartz"
entity e2	type: position value	value: "executive vice president"
		company: e3
entity e3	type: manufacturer	name:"Hupplewhite Inc."
entity e6	type: person	name: "Harry Himmelfarb"
event e7	type: leave-job	person: e1 position: e2
event e8	type: succeed	person1: e6 person2: e1

53

Inferencing and event merging

- Partial information about an event may be spread over several sentences
  - this information needs to be combined before a template can be generated
- some of the information may also be implicit
  - this information needs to be made explicit through an inference process

54

Inferencing and event merging

- In our example, we need to determine what the "succeed" predicate implies, e.g.
- "Sam was president. He was succeeded by Harry."
  - -> Harry will become president
- "Sam will be president; he succeeds Harry"
  - -> Harry was president.

55

Inferencing and event merging

- Such inferences can be implemented by production system rules:
  - leave-job(X-person,Y-job) & succeed(Z-person,X-person) => start-job(Z-person,Y-job)
  - start-job(X-person,Y-job) & succeed(X-person,Z-person) => leave-job(Z-person,Y-job)

56

Inferencing and event merging

entity e1	type: person	name: "Sam Schwartz"
entity e2	type: position value	value: "executive vice president"
		company: e3
entity e3	type: manufacturer	name:"Hupplewhite Inc."
entity e6	type: person	name: "Harry Himmelfarb"
event e7	type: leave-job	person: e1 position: e2
event e8	type: succeed	person1: e6 person2: e1
event e9	type: start-job	person: e6 position:e2

57

Target templates

Event	leave job
Person	Sam Schwartz
Position	executive vice president
Company	Hupplewhite Inc.
Event	start job
Person	Harry Himmelfarb
Position	executive vice president
Company	Hupplewhite Inc

58

Inferencing and event merging

- Our simple scenario did not require us to take account of the time of each event
- for many scenarios, time is important
  - explicit times must be reported, or
  - the sequence of events is significant
- time information may be derived from many sources

59

Inferencing and event merging

- Sources of time information
  - absolute dates and times ("on April 6, 1995")
  - relative dates and times ("last week")
  - verb tenses
  - knowledge about inherent sequence of events
- since time analysis may interact with other inferences, it will normally be performed as part of the inference stage of processing

60

(MUC) Evaluation

- Participants are initially given
  - a detailed description of the scenario (the information to be extracted)
  - a set of documents and the templates to be extracted from these documents (the training corpus)
- systems developers then get some time (1-6 months) to adapt their system to the new scenario

61

(MUC) Evaluation

- After this time, each participant
  - gets a new set of documents (the test corpus)
  - uses their system to extract information from these documents
  - returns the extracted templates to the conference organizer
- the organizer has manually filled a set of templates (the answer key) from the test corpus

62

(MUC) Evaluation

- Each system is assigned a variety of scores by comparing the system response to the answer key
- the primary scores are precision and recall

63

(MUC) Evaluation

- N_key = total number of filled slots in the answer key
- N_response = total number of filled slots in the system response
- N_correct = number of correctly filled slots in the system response (= the number which match the answer key)

64

(MUC) Evaluation

- precision = N_correct / N_response
- recall = N_correct / N_key
- F score is a combined recall-precision score:
  - F = (2 x precision x recall) / (precision + recall)

65

Some design issues

- the amount of syntactic analysis
- portability

66

To parse or not to parse

- One of the most evident differences among extraction systems involves the amount of syntactic analysis which is performed
- the benefits of syntax analysis are clear
  - we may want to extract the subject and object of verbs like "hire" and "fire"
  - if syntactic relations were already correctly marked, the scenario patterns would probably be simpler and more accurate

67

To parse or not to parse

- Many early extraction systems performed full syntactic analysis
- however, building a complete syntactic structure is not easy
  - parsers may end up making poor local decisions about structures when they try to create a parse spanning the entire sentence
  - large search space -> parsing may be slow
  - in IE, irrelevant parts should not be processed

68

To parse or not to parse

- Most IE systems create only partial syntactic structures
  - syntactic structures which can be created with high confidence and using local information
  - e.g. bracketing noun and verb groups
  - + some larger noun groups if there is semantic evidence to confirm the correctness of the attachment

69

To parse or not to parse

- Full syntactic analysis has also the role of regularizing syntactic structure
  - different clausal forms, such as active and passive forms, relative clauses, reduced relatives etc., are mapped into essentially the same structure
  - this regularization simplifies the scenario pattern matching -> fewer forms of each scenario pattern

70

To parse or not to parse

- In a partial parsing approach which does not perform such regularization, there must be separate scenario patterns for each syntactic form
  - -> multiplies the number of patterns to be written by a factor of 5 to 10

71

To parse or not to parse

- For instance, we would need separate patterns for
  - IBM hired Harry (active)
  - Harry was hired by IBM (passive)
  - IBM, which hired Harry,... (relative clause)
  - Harry, who was hired by IBM, ...
  - Harry, hired by IBM, ... (reduced relative)
  - etc.

72

To parse or not to parse

- To handle this some systems haved introduced metarules or rule schemata
  - methods for writing a single basic pattern and having it transformed into the patterns needed for the various syntactic forms of a clause

73

To parse or not to parse

- Basic pattern:
  - Subject=company verb=hired object=person
- the system would generate:
  - company hired person
  - person was hired by company
  - company, which hired person,
  - person, who was hired by company
  - person, hired by company
  - etc.

74

To parse or not to parse

- Also modifiers which can intervene between the sentence elements can be inserted into the appropriate positions in each clausal pattern
  - IBM yesterday promoted Mr. Smith to executive vice president.
  - GE, which was founded in 1880, promoted Mr. Smith to president

75

To parse or not to parse

- Situation may change (or has already changed?) as the full-sentence parsing technology is improving
  - speed -> parsers are becoming faster
  - robustness -> parsers can also handle 'ungrammatical' sentences, so they do not introduce more errors than they fix

76

Portability

- One of the barriers to making IE a practical technology is the cost of adapting an extraction system to a new scenario
- in general, each application of extraction will involve a different scenario
- implementing a scenario should not require too much time and not the skills of the extraction system designers

77

Portability

- The basic question in developing a customization tool is the form and level of the information to be obtained from the user
- goal: the customization is performed directly by the user (rather than by an expert system developer)

78

Portability

- if we are using a pattern matching system, most work will probably be focused on the development of the set of patterns
- also changes
  - to the semantic hierarchy
  - to the set of inference rules
  - to the rules for creating the output templates

79

Portability

- We cannot expect the user to have experience with writing patterns (regular expressions with associated actions) and familiarity with formal syntactic structure
- one possibility is to provide a graphical representation of the patterns but still too many details of the patterns are shown
- possible solution: learning from examples

80

Portability

- Learning of patterns
  - information is obtained from examples of sentences of interest and the information to be extracted
- for instance, in a system "AutoSlog" patterns are created semiautomatically from the templates of the training corpus

81

Portability

- In AutoSlog
  - given a template slot which is filled with words from the text (e.g. a name), the program would search for these words in the text and would hypothesize a pattern based on the immediate context of these words
  - the patterns are presented to a system developer, who can accept or reject the pattern

82

Portability

- The earlier MUC conferences involved large training corpora (over 1000 documents and their templates)
- however, the preparation of large, consistent training corpora is expensive
  - large corpora would not be available for most real tasks
  - users are willing to prepare a few examples (20-30?) only

83

Next time...

- We will talk about the ways to automatize the phases of the IE process, i.e. the ways to make systems more portable and faster to implement

84