Information extraction from text

Part 3

Learning of extraction rules

- IE systems depend on a domain-specific knowledge
  - acquiring and formulating the knowledge may require many person-hours of highly skilled people (usually both domain and the IE system expertize is needed)
  - the systems cannot be easily scaled up or ported to new domains
  - automating the dictionary construction is needed

2

Learning of extraction rules

- AutoSlog
- Crystal
- AutoSlog-TS
- Multi-level bootstrapping
- repeated mentions of events in different forms
- ExDisco

3

AutoSlog

- Ellen Riloff, University of Massachusetts
  - Automatically constructing a dictionary for information extraction tasks, 1993
- continues the work with CIRCUS

4

AutoSlog

- Automatically constructs a domain-specific dictionary for IE
- given a training corpus, AutoSlog proposes a set of dictionary entries that are capable of extracting the desired information from the training texts
- if the training corpus is representative of the target texts, the dictionary should work also with new texts

5

AutoSlog

- To extract information from text, CIRCUS relies on a domain-specific dictionary of concept node definitions
  - a concept node definition is a case frame that is triggered by a lexical item and activated in a specific linguistic context
  - each concept node definition contains a set of enabling conditions which are constraints that must be satisfied

6

Concept node definitions

- Each concept node definition contains a set of slots to extract information from the surrounding context
  - e.g., slots for perpetrators, victims, ...
  - each slot has
    - a syntactic expectation: where the filler is expected to be found in the linguistic context
    - a set of hard and soft constraints for its filler

7

Concept node definitions

- Given a sentence as input, CIRCUS generates a set of instantiated concept nodes as its output
- if multiple triggering words appear in sentence, then CIRCUS can generate multiple concept nodes for that sentence
  - if no triggering words are found in the sentence, no output is generated

8

Concept node dictionary

- Since concept nodes are CIRCUS' only output for a text, a good concept node dictionary is crucial
- the UMASS/MUC4 system used 2 dictionaries
  - a part-of-speech lexicon: 5436 lexical definitions, including semantic features for domain-specific words
  - a dictionary of 389 concept node definitions

9

Concept node dictionary

- For MUC4, the concept node dictionary was manually constructed by 2 graduate students: 1500 person-hours

10

AutoSlog

- Two central observations:
  - the most important facts about a news event are typically reported during the initial event description
    - the first reference to a major component of an event (e.g. a victim or perpetrator) usually occurs in a sentence that describes the event
    - the first reference to a targeted piece of information is most likely where the relationship between that information and the event is made explicit

11

AutoSlog

- The immediate linguistic context surrounding the targeted information usually contains the words or phrases that describe its role in the event
  - e.g. "A U.S. diplomat was kidnapped by FMLN guerillas"
  - the word 'kidnapped' is the key word that relates the victim (A U.S. diplomat) and the perpetrator (FMLN guerillas) to the kidnapping event
  - 'kidnapped' is the triggering word

12

Algorithm

- Given a set of training texts and their associated answer keys, AutoSlog proposes a set of concept node definitions that are capable of extracting the information in the answer keys from the texts

13

Algorithm

- Given a string from an answer key template
  - AutoSlog finds the first sentence in the text that contains the string
  - the sentence is handed over to CIRCUS which generates a conceptual analysis of the sentence
  - using the analysis, AutoSlog identifies the first clause in the sentence that contains the string

14

Algorithm

- A set of heuristics are applied to the clause to suggest a good conceptual anchor point for a concept node definition
- if none of the heuristics is satisfied then AutoSlog searches for the next sentence in the text and process is repeated

15

Conceptual anchor point heuristics

- A conceptual anchor point is a word that should activate a concept
- each heuristic looks for a specific linguistic pattern in the clause surrounding the targeted string
- if a heuristic identifies its pattern in the clause then it generates
  - a conceptual anchor point
  - a set of enabling conditions

16

Conceptual anchor point heuristics

- Suppose
  - the clause "the diplomat was kidnapped"
  - the targeted string "the diplomat"
- the string appears as the subject and is followed by a passive verb 'kidnapped'
- a heuristic that recognizes the pattern **<subject> passive-verb** is satisfied
  - returns the word 'kidnapped' as the conceptual anchor point, and
  - as enabling condition: a passive construction

17

Linguistic patterns

- <subj> passive-verb
- <subj> active-verb
- <subj> verb infinitive
- <subj> aux noun
- passive-verb <dobj>
- active-verb <dobj>
- infinitive <dobj>
- <victim> was murdered
- <perpetrator> bombed
- <perpetrator> attempted to kill
- <victim> was victim
- killed <victim>
- bombed <target>
- to kill <victim>

18

Linguistic patterns

- verb infinitive <dobj>
- gerund <dobj>
- noun aux <dobj>
- noun prep <np>
- active-verb prep <np>
- passive-verb prep <np>
- threatened to attack <target>
- killing <victim>
- fatality was <victim>
- bomb against <target>
- killed with <instrument>
- was aimed at <target>

19

Building concept node definitions

- The conceptual anchor point is used as the triggering word
- enabling conditions are included
- a slot to extract the information
  - a name of the slot comes from the answer key template
  - the syntactic constituent from the linguistic pattern, e.g. the filler is the subject of the clause

20

Building concept node definitions

- hard and soft constraints for the slot
  - e.g. constraints to specify a legitimate victim
- a type
  - e.g. the type of the event (bombing, kidnapping) from the answer key template
  - uses domain-specific mapping from template slots to the concept node types
    - not always the same: a concept node is only a part of the representation

21

Example

…, public buildings were bombed and a car-bomb was…

Slot filler in the answer key template: "public buildings"

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))

22

A bad definition

"they took 2-year-old gilberto molasco, son of patricio rodriguez, .."

CONCEPT NODE
Name: victim-active-verb-dobj-took
Trigger: took
Variable Slots: (victim (*DOBJ* 1))
Constraints: (class victim *DOBJ*)
Constant Slots: (type kidnapping)
Enabling Conditions: ((active))

23

A bad definition

- a concept node is triggered by the word "took" as an active verb
- this concept node definition is appropriate for this sentence, but in general we don't want to generate a kidnapping node every time we see the word "took"

24

Bad definitions

- AutoSlog generates bad definitions for many reasons
  - a sentence contains the targeted string but does not describe the event
  - a heuristic proposes the wrong conceptual anchor point
  - CIRCUS analyzes the sentence incorrectly
- Solution: human-in-the-loop

25

Empirical results

- Training data: 1500 texts (MUC-4) and their associated answer keys
  - 6 slots were chosen
  - 1258 answer keys contained 4780 string fillers
- result:
  - 1237 concept node definitions

26

Empirical results

- human-in-the-loop:
  - 450 definitions were kept
  - time spent: 5 hours (compare: 1500 hours for a hand-crafted dictionary)
- the resulting concept node dictionary was compared with a hand-crafted dictionary within the UMass/MUC-4 system
  - precision, recall, F-measure almost the same

27

CRYSTAL

- Soderland, Fisher, Aseltine, Lehnert (University of Massachusetts), CRYSTAL: Inducing a conceptual dictionary, 1995

28

Motivation

- CRYSTAL addresses some issues concerning AutoSlog:
  - the constraints on the extracted constituent are set in advance (in heuristic patterns and in answer keys)
  - no attempt to relax constraints, merge similar concept node definitions, or test proposed definitions on the training corpus
  - 70% of the definitions found by AutoSlog were discarded by the human

29

Medical domain

- Task is to analyze hospital reports and identify references to "diagnosis" and to "sign or symptom"
- subtypes of Diagnosis
  - confirmed, ruled out, suspected, pre-existing, past
- subtypes of Sign or Symptom:
  - present, absent, presumed, unknown, history

30

Example: concept node

- Concept node type: Sign or Symptom
- Subtype: absent
- Extract from Direct Object
- Active voice verb
- Subject constraints:
  - words include "PATIENT"
  - head class: <Patient or Disabled Group>
- Verb constraints: words include "DENIES"
- Direct object constraints: head class <Sign or Symptom>

31

Example: concept node

- This concept node definition would extract "any episodes of nausea" from the sentence "The patient denies any episodes of nausea"
- it fails to apply to the sentence "Patient denies a history of asthma", since asthma is of semantic class <Disease or Syndrome>, which is not a subclass of <Sign or Symptom>

32

Quality of concept node definitions

- Concept node type: Diagnosis
- Subtype: pre-existing
- Extract from "with"-PP
- Passive voice verb
- Verb constraints: words include "DIAGNOSED"
- PP constraints:
  - preposition = "WITH"
  - words include "RECURRENCE OF"
  - modifier class <Body Part or Organ>
  - head class <Disease or Syndrome>

33

Quality of concept node definitions

- This concept node definition identifies pre-existing diagnoses with a set of constraints that could be summarized as:
  - "... was diagnosed with recurrence of <body_part> <disease>"
  - e.g., "The patient was diagnosed with a recurrence of laryngeal cancer"
- is this definition a good one?

34

Quality of concept node definitions

- Will this concept node definition reliably identify only pre-existing diagnoses?
- Perhaps in some texts the recurrence of a disease is actually
  - a principal diagnosis of the current hospitalization and should be identified as "diagnosis, confirmed"
  - or a condition that no longer exists -> "past"
- in such cases: an extraction error occurs

35

Quality of concept node definitions

- On the other hand, this definition might be reliable, but miss some valid examples
  - the valid cases might be covered if the constraints were relaxed
- judgments about how tightly to constrain a concept node definition are difficult to make (manually)
- -> automatic generation of definitions with gradual relaxation of constraints

36

Creating initial concept node definitions

- Annotation of a set of training texts by a domain expert
  - each phrase that contains information to be extracted is bracketed with tags to mark the appropriate concept node type and subtype
- the annotated texts are segmented by the sentence analyzer to create a set of training instances

37

Creating initial concept node definitions

- Each instance is a text segment
  - some syntactic constituents may be tagged as positive instances of a particular concept node type and subtype

38

Creating initial concept node definitions

- Process begins with a dictionary of concept node definitions built from each instance that contains the type and subtype being learned
  - if a training instance has its subject tagged as "diagnosis" with subtype "pre-existing", an initial concept type definition is created that extracts the phrase in the subject as a pre-existing diagnosis
  - constraints derived from the words

39

Induction

- Before the induction process begins, CRYSTAL cannot predict which characteristics of an instance are essential to the concept node definitions
- all details are encoded as constraints
  - the exact sequence of words and the exact sets of semantic classes are required
- later CRYSTAL learns which constraints should be relaxed

40

Example

- "Unremarkable with the exception of mild shortness of breath and chronically swollen ankles"
- the domain expert has marked "shortness of breath" and "swollen ankles" with type "sign or symptom" and subtype "present"

41

Example: initial concept node definition

CN-type: Sign or Synptom
Subtype: Present
Extract from "WITH"-PP
Verb = <NULL>
Subject constraints: words include "UNREMARKABLE"
PP constraints:
preposition = "WITH"
words include "THE EXCEPTION OF MILD SHORTNESS OF BREATH AND CHRONICALLY SWOLLEN ANKLES"
modifier class <Sign or Sympton>
head class <Sign or Symptom>, <Body Location or Region>

42

Initial concept node definition

- It is unlikely that an initial concept node definition will ever apply to a sentence from a different text
  - too tightly constrained
- constraints have to be relaxed
  - semantic constraints: moving up the semantic hierarchy or dropping the constraint
  - word constraints: dropping some words

43

Inducing generalized concept node definitions

- The combinatorics on ways to relax constraints becomes overwhelming
  - in our example, there are over 57,000 possible generalizations of the initial concept node definitions
- useful generalizations are found by locating and comparing definitions that are highly similar

44

Inducing generalized concept node definitions

- Let D be the definition being generalized
- there is a definition D' which is very similar to D
  - according to a similarity metric that counts the number of relaxations required to unify two concept node definitions
- a new definition U is created with constraints relaxed just enough to unify D and D'

45

Inducing generalized concept node definitions

- The new definition U is tested against the training corpus
  - the definition U should not extract phrases that were not marked with the type and subtype being learned
- If U is a valid definition, all definitions covered by U are deleted from the dictionary
  - D and D' are deleted

46

Inducing generalized concept node definitions

- The definition U becomes the current definition and the process is repeated
  - a new definition similar to U is found etc.
- eventually a point is reached where further relaxation would produce a definition that exceeds some pre-specified error tolerance
  - the generalization process is begun on another initial concept node definition until all initial definitions have been considered for generalization

47

Algorithm

```
Initialize Dictionary and Training Instances Database
do until no more initial CN definitions in Dictionary
   D = an initial CN definition removed from the dictionary
   loop
      D' = the most similar CN definition to D
      if D' = NULL, exit loop
      U = the unification of D and D'
      Test the coverage of U in Training Instances
      if the error rate of U > Tolerance, exit loop
      Delete all CN definitions covered by U
      Set D = U
   Add D to the Dictionary
Return the Dictionary
```

48

Unification

- Two similar definitions are unified by finding the most restrictive constraints that cover both
- if word constraints from the two definitions have an intersecting string of words, the unified word constraint is that intersecting string
  - otherwise the word constraint is dropped

49

Unification

- Two class constraints may be unified by moving up the semantic hierarchy to find a common ancestor of classes
  - class constraints are dropped when they reach the root of the semantic hierarchy
  - if a constraint on a particular syntactic component is missing from one of the two definitions, that constraint is dropped

50

Examples of unification

- 1. Subject is <Sign or Symptom>
- 2. Subject is <Laboratory or Test Result>
- unified: <Finding> (the common parent in the semantic hierarchy)

- 1. A
- 2. A and B
- unified: A

51

CRYSTAL: conclusion

- Goal of CRYSTAL is
  - to find the minimum set of generalized concept node definitions that cover all of the positive training instances
  - to test each proposed definition against the training corpus to ensure that the error rate is within a predefined tolerance
- requirements
  - a sentence analyzer, a semantic lexicon, a set of annotated training texts

52

AutoSlog-TS

- Riloff (University of Utah): Automatically generating extraction patterns from untagged text, 1996

53

Extracting patterns from untagged text

- Both AutoSlog and CRYSTAL need manually tagged or annotated information to be able to extract patterns
- manual annotation is expensive, particularly for domain-specific applications like IE
  - may also need skilled people
  - ~8 hours to annotate 160 texts (AutoSlog)

54

Extracting patterns from untagged text

- The annotation task is complex
- e.g. for AutoSlog the user must annotate relevant noun phrases
  - What constitutes a relevant noun phrase?
  - Should modifiers be included or just a head noun?
  - All modifiers or just the relevant modifiers?
  - Determiners? Appositives?

55

Extracting patterns from untagged text

- The meaning of simple NP's may change substantially when a prepositional phrase is attached
  - "the Bank of Boston" vs. "the Bank of Toronto"
- Which references to tag?
  - Should the user tag all references to a person?

56

AutoSlog-TS

- Needs only a preclassified corpus of relevant and irrelevant texts
  - much easier to generate
  - relevant texts are available online for many applications
- generates an extraction pattern for every noun phrase in the training corpus
- the patterns are evaluated by processing the corpus and generating relevance statistics for each pattern

57

Process

- Stage 1:
  - the sentence analyzer produces a syntactic analysis for each sentence and identifies the noun phrases
  - for each noun phrase, the heuristic (AutoSlog) rules generate a pattern (a concept node) to extract the noun phrase
    - if more than one rule matches the context, multiple extraction patterns are generated
    - <subj> bombed, <subj> bombed embassy

58

Process

- Stage 2:
  - the training corpus is processed a second time using the new extraction patterns
  - the sentence analyzer activates all patterns that are applicable in each sentence
  - relevance statistics are computed for each pattern
  - the patterns are ranked in order of importance to the domain

59

Relevance statistics

- relevance rate: Pr (relevant text | text contains pattern i) = rfreq_i / totfreq_i
  - rfreq_i : the number of instances of pattern i that were activated in the relevant texts
  - totfreq_i: the total number of instances of pattern i in the training corpus
- domain-specific expressions appear substantially more often in relevant texts than in irrelevant texts

60

Ranking of patterns

- The extraction patterns are ranked according to the formula:
  - relevance rate * log (frequency)
  - or zero, if relevance rate < 0.5
    - in this case, the pattern is negatively correlated with the domain (assuming the corpus is 50% relevant)
- the formula promotes patterns that are
  - highly relevant or highly frequent

61

The top 25 extraction patterns

- <subj> exploded
- murder of <np>
- assassination of <np>
- <subj> was killed
- <subj> was kidnapped
- attack on <np>
- <subj> was injured
- exploded in <np>

62

The top 25 extraction patterns, continues

- death of <np>
- <subj> took place
- caused <dobj>
- claimed <dobj>
- <subj> was wounded
- <subj> occurred
- <subj> was located
- took_place on <np>

63

The top 25 extraction patterns, continues

- responsibility for <np>
- occurred on <np>
- was wounded in <np>
- destroyed <dobj>
- <subj> was murdered
- one of <np>
- <subj> kidnapped
- exploded on <np>
- <subj> died

64

Human-in-the-loop

- The ranked extraction patterns were presented to a user for manual review
- the user had to
  - decide whether a pattern should be accepted or rejected
  - label the accepted patterns
    - murder of <np> -> <np> means the victim

65

AutoSlog-TS: conclusion

- Empirical results comparable to AutoSlog
  - recall slightly worse, precision better
- the user needs to
  - provide sample texts (relevant and irrelevant)
  - spend some time filtering and labeling the resulting extraction patterns

66

Multi-level bootstrapping

- Riloff (Utah), Jones(CMU): Learning Dictionaries for Information Extraction by Multi-level Bootstrapping, 1999

67

Multi-level bootstrapping

- An algorithm that generates simultaneously
  - a semantic lexicon
  - extraction patterns
- input: unannotated training texts and a few seed words for each category of interest (e.g. location)

68

Multi-level bootstrapping

- Mutual bootstrapping technique
  - extraction patterns are learned from the seed words
  - the learned extraction patterns are exploited to identify more words that belong to the semantic category

69

Multi-level bootstrapping

- a second level of bootstrapping
  - only the most reliable lexicon entries are retained from the results of mutual bootstrapping
  - the process is restarted with the enhanced semantic lexicon
- the two-tiered bootstrapping process is less sensitive to noise than a single level bootstrapping

70

Mutual bootstrapping

- Observation: extraction patterns can generate new examples of a semantic category, which in turn can be used to identify new extraction patterns

71

Mutual bootstrapping

- Process begins with a text corpus and a few predefined seed words for a semantic category
  - text corpus: e.g. terrorist events texts, web pages
  - semantic category : (e.g.) location, weapon, company

72

Mutual bootstrapping

- AutoSlog is used in an exhaustive fashion to generate extraction patterns for every noun phrase in the corpus
- The extraction patterns are applied to the corpus and the extractions are recorded

73

Mutual bootstrapping

- Input for the next stage:
  - a set of extraction patterns, and for each pattern, the NPs it can extract from the training corpus
  - this set can be reduced by pruning the patterns that extract one NP only
    - general (enough) linguistic expressions are preferred

74

Mutual bootstrapping

- Using the data, the extraction pattern is identified that is most useful for extracting known category members
  - known category members in the beginning = the seed words
  - e.g. in the example, 10 seed words were used for the location category (in terrorist texts): bolivia, city, colombia, district, guatemala, honduras, neighborhood, nicaragua, region, town

75

Mutual bootstrapping

- The best extraction pattern found is then used to propose new NPs that belong to the category (= should be added to the semantic lexicon)
- in the following algorithm:
  - SemLex = semantic lexicon for the category
  - Cat_EPlist = the extraction patterns chosen for the category so far

76

Algorithm

- Generate all candidate extraction patterns from the training corpus using AutoSlog
- Apply the candidate extraction patterns to the training corpus and save the patterns with their extractions to EPdata
- SemLex = {seed_words}
- Cat_EPlist = {}

77

Algorithm, continues

- Mutual Bootstrapping Loop
  - 1. Score all extraction patterns in EPdata
  - 2. best_EP = the highest scoring extraction pattern not already in Cat_EPlist
  - 3. Add best_EP to Cat_EPlist
  - 4. Add best_EP's extractions to SemLex
  - 5. Go to step 1

78

Mutual bootstrapping

- At each iteration, the algorithm saves the best extraction pattern for the category to Cat_EPlist
- all of the extractions of this pattern are assumed to be category members and are added to the semantic lexicon

79

Mutual bootstrapping

- In the next iteration, the best pattern that is not already in Cat_EPlist is identified
  - based on both the original seed words + the new words that have been added to the lexicon
- the process repeats until some end condition is reached

80

Scoring

- Based on how many different lexicon entries a pattern extracts
- the metric rewards generality
  - a pattern that extracts a variety of category members will be scored higher than a pattern that extracts only one or two different category members, no matter how often

81

Scoring

- Head phrase matching:
  - X matches Y if X is the rightmost substring of Y
  - "New Zealand" matches "eastern New Zealand" and "the modern day New Zealand"
  - ... but not "the New Zealand coast" or "Zealand"
  - important for generality
- each NP was stripped of leading articles, common modifiers ("his", "other",...) and numbers before being saved to the lexicon

82

Scoring

- The same metric was used as in AutoSlog-TS
  - score(pattern_i) = R_i * log(F_i)
- F_i: the number of unique lexicon entries among the extractions produced by pattern_i
- N_i: the total number of unique NPs that pattern_i extracted
- R_i = F_i / N_i

83

Example

- 10 seed words were used for the location category (terrorist texts):
  - bolivia, city, colombia, district, guatemala, honduras, neighborhood, nicaragua, region, town
- the first five iterations...

84

Example

Best pattern ”headquartered in <x> (F=3, N=4)
Known locations nicaragua
New locations san miguel, chapare region, san miguel city

Best pattern ”gripped <x>” (F=2, N=2)
Known locations colombia, guatemala
New locations none

85

Example

Best pattern ”downed in <x>” (F=3, N=6)
Known locations nicaragua, san miguel*, city
New locations area, usulutan region, soyapango

Best pattern ”to occupy <x>” (F=4, N=6)
Known locations nicaragua, town
New locations small country, this northern area,
san sebastian neighborhood, private property

86

Example

Best pattern ”shot in <x>” (F=5, N=12)
Known locations city, soyapango*
New locations jauja, central square, head, clash, back,
central mountain region, air,
villa el_salvador district,
northwestern guatemala, left side

87

Strengths and weaknesses

- The extraction patterns have identified several new location phrases
  - jauja, san miguel, soyapango, this northern area
- but several non-location phrases have also been generated
  - private property, head, clash, back, air, left side
  - most mistakes due to ”shot in <x>”
- many of these patterns occur infrequently in the corpus

88

Multi-level bootstrapping

- The mutual bootstrapping algorithm works well but its performance can deteriorate rapidly when non-category words enter the semantic lexicon
- once an extraction pattern is chosen for the dictionary, all of its extractions are immediately added to the lexicon
  - few bad entries can quickly infect the dictionary

89

Multi-level bootstrapping

- For example, if a pattern extracts dates as well as locations, then the dates are added to the lexicon and subsequent patterns are rewarded for extracting these dates
- to make the algorithm more robust, a second level of bootstrapping is used

90

Multi-level bootstrapping

- The outer bootstrapping mechanism ("meta-bootstrapping")
  - compiles the results from the inner (mutual) bootstrapping process
  - identifies the five most reliable lexicon entries
  - these five NPs are retained for the permanent semantic lexicon
  - the entire mutual bootstrapping process is then restarted from scratch (with new lexicon)

91

Scoring for reliability

- To determine which NPs are most reliable, each NP is scored based on the number of different category patterns that extracted it
  - how many members in the Cat_EPlist?
- intuition: a NP extracted by e.g. three different category patterns is more likely to belong to the category than a NP extracted by only one pattern

92

Multi-level bootstrapping

- The main advantage of meta-bootstrapping comes from re-evaluating the extraction patterns after each mutual bootstrapping process
- for example, after the first mutual bootstrapping run, 5 new words are added to the permanent semantic lexicon

93

Multi-level bootstrapping

- the mutual bootstrapping is restarted with the original seed words + the 5 new words
- now, the best pattern selected might be different from the best pattern selected last time -> a snowball effect
- in practice, the ordering of patterns changes: more general patterns float to the top as the semantic lexicon grows

94

Multi-level bootstrapping: conclusion

- Both a semantic lexicon and a dictionary of extraction patterns are acquired simultaneously
- resources needed:
  - corpus of (unannotated) training texts
  - a small set of words for a category

95

Repeated mentions of events in different forms

- Brin 1998, Agichtein&Gravano 2000
- in many cases we can obtain documents from multiple information sources, which will include descriptions of the same relation or event in different forms
- if several descriptions mention the same names participants, there is a good chance that they are instances of the same relation

96

Repeated mentions of events in different forms

- Suppose that we are seeking patterns corresponding to the relation HQ between a company and the location of its headquarters
- we are initially given one such pattern: "C, headquartered in L" => HQ(C,L)

97

Repeated mentions of events in different forms

- We can search for instances of this pattern in the corpus in order to collect pairs of invididuals in the relation HQ
  - for instance, "IBM, headquartered in Armonk" => HQ("IBM","Armonk")
- if we find other examples in the text which connect these pairs, e.g. "Armonk-based IBM", we might guess that the associated pattern "L-based C" is also indicator of HQ

98

ExDisco

- Yangarber, Grishman, Tapanainen, Huttunen
  - Automatic acquisition of domain knowledge for information extraction, 2000
  - Unsupervised discovery of scenario-level patterns for information extraction, 2000

99

Motivation: previous work

- A user interface which supports rapid customization of the extraction system to a new scenario
  - allows the user to provide examples of relevant events, which are automatically converted into the appropriate patterns and generalized to cover syntactic variants (passive, relative clause,...)
  - the user can also generalize the patterns

100

Motivation

- Although the user interface makes adapting the extraction system quite rapid, the burden is still on the user to find the appropriate set of examples

101

Basic idea

- Look for linguistic patterns which appear with 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
  - one of the best indications of the relevance of the documents is the presence of good patterns -> circularity -> acquired in tandem

102

Preprocessing

- Name recognition marks all instances of names of people, companies, and locations -> replaced with the class name
- a parser is used to extract all the clauses from each document
  - for each clause, a tuple is built, consisting of the basic syntactic constituents
  - different clause structures (passive...) are normalized

103

Preprocessing

- Because tuples may not repeat with sufficient frequency, each tuple is reduced to a set of pairs, e.g.
  - verb-object
  - subject-object
- each pair is used as a generalized pattern
- once relevant pairs have been identified, they can be used to gather the set of words for the missing roles

104

Discovery procedure

- Unsupervised procedure
  - the training corpus does not need to be annotated, not even classified
  - the user must provide 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

105

Discovery procedure

- 1. The pattern set is used to divide the corpus U into a set of relevant documents, R, and a set of non-relevant documents U - R
- 2. Search for new candidate patterns:
  - automatically convert each document in the corpus into a set of candidate patterns, one for each clause
  - rank patterns by the degree to which their distribution is correlated with document relevance

106

Discovery procedure

- 3. Add the highest ranking pattern to the pattern set
  - optionally present the pattern to the user for review
- 4. Use 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

107

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}

108

ExDisco: conclusion

- Resources needed:
  - unannotated, unclassified corpus
  - a set of seed patterns
- produces complete, multi-slot event patterns

109