Information extraction from text

Part 4

In this part

- IE from (semi-)structured text
- other applications
  - multilingual IE
  - question answering systems
  - news event tracking and detection
- closing of the course

2

WHISK

- Soderland: Learning information extraction rules for semi-structured and free text, Machine Learning, 1999

3

Semi-structured text (online rental ad)

Capitol Hill - 1 br twnhme. Fplc D/W W/D. Undrgrnd Pkg incl $675. 3 BR, upper flr of turn of ctry HOME. incl gar, grt N. Hill loc $995. (206) 999-9999
  (This ad last ran on 08/03/97.)
 <hr>

4

2 case frames extracted:

- Rental:
  - Neighborhood: Capitol Hill
  - Bedrooms: 1
  - Price: 675
- Rental:
  - Neighborhood: Capitol Hill
  - Bedrooms: 3
  - Price: 995

5

Semi-structured text

- The sample text (rental ad) is not grammatical nor has a rigid structure
  - we cannot use a natural language parser as we did before
  - simple rules that might work for structured text do not work here

6

Rule representation

- WHISK rules are based on a form of regular expression patterns that identify
  - the context of relevant phrases
  - the exact delimiters of the phrases

7

Rule for number of bedrooms and associated price

- ID:: 1
- Pattern:: *(Digit) 'BR' * '$' (Number)
- Output:: Rental {Bedrooms $1}{Price $2}
  - * : skip any number of characters until the next occurrence of the following term in the pattern (here e.g. : the next digit)
  - single quotes: literal -> exact (case insensitive) match
  - Digit: a single digit; Number: possibly multi-digit

8

Rule for number of bedrooms and associated price

- parentheses (unless within single quotes) indicate a phrase to be extracted
  - the phrase within the first set of parentheses (here: Digit) is bound to the variable $1 in the output portion of the rule
- if the entire pattern matches, a case frame is created with slots filled as labeled in the output portion
- if part of the input remains, the rule is re-applied starting from the last character matched before

9

2 case frames extracted:

- Rental:
  - Bedrooms: 1
  - Price: 675
- Rental:
  - Bedrooms: 3
  - Price: 995

10

Disjunction

- The user may define a semantic class
  - a set of terms that are considered to be equivalent
  - Digit and Number are special semantic classes (built-in in WHISK)
  - e.g. Bdrm = (brs|br|bds|bdrm|bd|bedrooms|bedroom|bed)
  - a set does not have to be complete or perfectly correct: still it may help WHISK to generalize rules

11

Rule for neighborhood, number of bedrooms and associated price

- ID:: 2
- Pattern:: *(Nghbr) *(Digit) ' ' Bdrm * '$' (Number)
- Output:: Rental {Neighborhood $1} {Bedrooms $2}{Price $3}
  - assuming the semantic classes Nghbr (neighborhood names for the city) and Bdrm

12

Algorithm for learning rules automatically

- A supervised learning algorithm: a set of hand-tagged training instances is needed
- the tagging process is interleaved with learning stages
- in each iteration, WHISK
  - presents the user a set of instances to tag
  - learns a set of rules from the expanded training set

13

Creating hand-tagged training instances

- It depends on the domain what constitutes an instance and what preprocessing is done
  - an entire text may constitute an instance
  - a text may be broken into multiple instances based on HTML tags or other regular expressions
  - semantic tags may be added in preprocessing etc.

14

Creating hand-tagged training instances

- The user adds a tag for each case frame to be extracted from the instance
  - if the case frame has multiple slots, the tag will be multi-slot
  - some of the "tagged" instances will have no tags, if the user has determined that the instance contains no relevant information

15

Tagged instance

@S[

Capitol Hill - 1 br twnhme. Fplc D/W W/D. Undrgrnd Pkg

incl $675. 3 BR, upper flr of turn of ctry HOME. incl gar,

grt N. Hill loc $995. (206) 999-9999

  (This ad last ran on 08/03/97.)

 <hr>]@S

@@TAGS Rental {Neighborhood Capitol Hill}{Bedrooms 1} {Price 675}

@@TAGS Rental {Neighborhood Capitol Hill}{Bedrooms 3} {Price 995}

16

Creating a rule from a seed instance

- untagged instances, training instances
- if a rule is applied successfully to an instance, the instance is considered to be **covered** by the rule
- if the extracted phrases exactly match a tag associated with the instance, it is considered a correct extraction, otherwise an error

17

Algorithm

```
WHISK (Reservoir)
    RuleSet = NULL
    Training = NULL
    Repeat at user's request
        Select a set of NewInst from Reservoir
        (User tags the NewInst)
        Add NewInst to Training
        Discard rules with errors on NewInst
        For each Inst in Training
            For each Tag of Inst
                If Tag is not covered by RuleSet
                    Rule = GROW_RULE(Inst, Tag, Training)
```

18

Algorithm

- At each iteration, the user tags a set of instances from the Reservoir of untagged instances
- some of these new training instances may be counterexamples to existing rules
  - -> the rule is discarded so that a new rule may be grown

19

Algorithm

- WHISK then selects an instance-tag pair for which the slot fills of the tag are not extracted from the instance by any rule in RuleSet
  - the instance-tag pair becomes a seed to grow a new rule that covers the seed
- WHISK induces rules top-down,
  - first finding the most general rule that covers the seed
  - then extending the rule by adding terms one at a time

20

Algorithm

- The metric used to select a new term is the Laplacian expected error of the rule:
  - Laplacian = (e+1)/(n+1)
- n: the number of extractions made on the training set
- e: the number of errors among those extractions

21

Algorithm

- This metric gives an estimate of the true error of a rule that is sensitive to the amount of support it has in the training set
- for alternate rules with the same number of training errors, the metric will be lowest for the rule with highest coverage
- a rule that covers a single tag with no error will get an expected error rate of 0.5

22

Anchoring the extraction slots

- WHISK grows a rule from a seed, by starting with an empty rule and anchoring the extraction boundaries one slot at a time
- to anchor an extraction, WHISK considers
  - a rule with terms added just within the extraction boundary (Base_1), and
  - a rule with terms added just outside the extraction (Base_2)

23

Tagged instance

@S[

Capitol Hill - 1 br twnhme. Fplc D/W W/D. Undrgrnd Pkg

incl $675. 3 BR, upper flr of turn of ctry HOME. incl gar,

grt N. Hill loc $995. (206) 999-9999

  (This ad last ran on 08/03/97.)

 <hr>]@S

@@TAGS Rental {Neighborhood Capitol Hill}{Bedrooms 1} {Price 675}

@@TAGS Rental {Neighborhood Capitol Hill}{Bedrooms 3} {Price 995}

24

Anchoring the extraction slots

- Anchoring Slot 1:
  - Base_1: * (Nghbr)
  - Base_2: '@start'(*) ' -'
- the semantic class Nghbr matches the first and only term of slot 1 -> Base_1
- the terms just outside the first slot are the start of the sentence and the space and hyphen -> Base_2
- assume: Base_1 applies correctly to more instances in the training set -> anchor for slot 1

25

two alternatives for extending the rule to cover slot 2

- Base_1: * (Nghbr) * (Digit)
- Base_2: * (Nghbr) * '- ' (*) ' br'

- each operates correctly on the seed instance
- Base_1 looks for the 1st digit after the first neighborhood
- Base_2 for the 1st '- ' after the 1st neighborhood and extracts all characters up to the next ' br' as the number of Bedrooms

26

Anchoring Slot 3

- Base_1: * (Nghbr) * (Digit) * (Number)
- Base_2: * (Nghbr) * (Digit) * '$' (*) '.'

- assume Base_1 was chosen for slot 2
- the process is continued for slot 3
- the final anchored rule operates correctly on the seed instance, but may make some extraction errors on other training instances
- WHISK continues adding terms

27

Adding terms to a proposed rule

- WHISK extends a rule
  - by considering each term that could be added
  - and testing the performance of each proposed extension on the hand-tagged training set
- the new rule must apply to the seed instance -> only terms from this instance need to be considered in growing the rule

28

Adding terms to a proposed rule

- If a term from the instance belongs to a user-defined semantic class, WHISK tries adding either the term itself or its semantic class to the rule
- each word, number, punctuation, or HTML tag in the instance is considered a term

29

GROW-RULE

```
GROW_RULE (Inst, Tag, Training)
    Rule = empty rule (terms replaced by wildcards)
    For i = 1 to number of slots in Tag
        ANCHOR (Rule, Inst, Tag, Training, i)
    Do until Rule makes no errors on Training or
        no improvement in Laplacian
            EXTEND_RULE(Rule, Inst, Tag, Training)
```

30

ANCHOR

```
ANCHOR (Rule, Inst, Tag, Training, i)
    Base_1 = Rule + terms just within extraction i
    Test first i slots of Base_1 on Training
    While Base_1 does not cover Tag
        EXTEND_RULE(Base_1, Inst, Tag, Training)
    Base_2 = Rule + terms just outside extraction i
    Test first i slots of Base_2 on Training
    While Base_2 does not cover Tag
        EXTEND_RULE(Base_2, Inst, Tag, Training)
    Rule = Base_1
    If Base_2 covers more of Training than Base_1
        Rule = Base_2
```

31

Extending a rule

- Each proposed extension of a rule is tested on the training set
- the proposed rule with lowest Laplacian expected error is selected as the next version of the rule
  - until the rule either makes no errors or
  - until the Laplacian is below a threshold and none of the extensions reduce the Laplacian

32

Extending a rule

- If several proposed rules have the same Laplacian, WHISK uses heuristics that prefer the semantic class over a word
- rationale: to prefer the least restrictive rule that fits the data -> rules probably operate better on unseen data
- also terms near extraction boundaries are preferred

33

EXTEND-RULE

```
EXTEND_RULE (Rule, Inst, Tag, Training)
    Best_Rule = NULL;  Best_L = 1.0
    If Laplacian of Rule within error tolerance
        Best_Rule = Rule
        Best_L = Laplacian of Rule
    For each Term in Inst
        Proposed = Rule + Term
        Test Proposed on Training
        If Laplacian of Proposed < Best_L
            Best_Rule = Proposed
            Best_L = Laplacian of Proposed
    Rule = Best_Rule
```

34

Rule set may not be optimal

- WHISK cannot guarantee that the rules it grows are optimal
  - optimal = the lowest Laplacian expected error on the hand-tagged training instances
- terms are added and evaluated one at a time
  - it may happen that adding two terms would create a reliable rule with high coverage and few errors, but adding either term alone does not constrain the rule from making extraction errors

35

Rule set may not be optimal

- If WHISK makes a "wrong" choice of terms to add, it may miss a reliable, high-coverage rule, but will continue adding terms until the rule operates reliably on the training set
  - such a rule will be more restrictive than the optimal rule and will tend to have lower coverage on unseen instances

36

Structured text

- When the text is rigidly structured, text extraction rules can be learned easily from only a few examples
  - e.g., structured text on the web is often created automatically by a formatting tool that delimits the variable information with exactly the same HTML tags or other labels

37

Structured text

<td nowrap> Thursday
<img src="/WEATHER/images/pcloudy.jpg" alt="partly cloudy" width=64 height=64>
 partly cloudy
 High: 29 C / 84 F
 Low: 13 C / 56 F</td>

ID:: 4
Pattern:: *(Day)' '* '1> '(*) '
* ' ' (*) ' ' * ' ' (*) '
Output:: Forecast {Day $1}{Conditions $2}{High $3}{Low $4}

38

Structured text

- WHISK can learn the previous rule from two training instances (as long as the variable information is not accidentally identical)
- in experiments, this rule gave recall 100% at precision 100%

39

Evaluation

- perfect recall and precision can generally be obtained for structured text in which each set of slots is delimited with some unique sequence of HTML tags or labels
- for less structured text, the performance varies with the difficulty of the extraction task (50% - 100%)
  - for the rental ad domain, one rule covers 70% of the recall (with precision 97%)

40

Other applications using IE

- multilingual IE
- question answering systems
- (news) event detection and tracking

41

Multilingual IE

- Assume we have documents in two languages (English/French), and the user requires templates to be filled in one of the languages (English) from documents in either language
  - "Gianluigi Ferrero a assisté à la réunion annuelle de Vercom Corp à Londres."
  - "Gianluigi Ferrero attended the annual meeting of Vercom Corp in London."

42

Both texts should produce the same template fill:

- <meeting-event-01> :=
  - organisation: 'Vercom Corp'
  - location: 'London'
  - type: 'annual meeting'
  - present: <person-01>
- <person-01> :=
  - name: 'Gianluigi Ferrero'
  - organisation: UNCLEAR

43

Multilingual IE: three ways of addressing the problem

- 1. solution
  - A full French-English machine translation (MT) system translates all the French texts to English
  - an English IE system then processes both the translated and the English texts to extract English template structures
  - the solution requires a separate full IE system for each target language and a full MT system for each language pair

44

Multilingual IE: three ways of addressing the problem

- 2. solution
  - Separate IE systems process the French and English texts, producing templates in the original source language
  - a 'mini' French-English MT system then translates the lexical items occurring in the French templates
  - the solution requires a separate full IE system for each language and a mini-MT system for each language pair

45

Multilingual IE: three ways of addressing the problem

- 3. solution
  - a general IE system, with separate French and English front ends
  - the IE system uses a language-independent domain model in which 'concepts' are related via bi-directional mappings to lexical items in multiple language-specific lexicons
  - this domain model is used to produce a language-independent representation of the input text - a discourse model

46

Multilingual IE: three ways of addressing the problem

- 3. solution continues…
  - the required information is extracted from the discourse model and the mappings from concepts to the English lexicon are used to produce templates with English lexical items
  - the solution requires a separate syntactic/semantic analyser for each language, and the construction of mappings between the domain model and a lexicon for each language

47

Multilingual IE

- Which parts of the IE process/systems are language-specific?
- Which parts of the IE process are domain-specific?

48

Question answering systems: TREC (8-10)

- Participants were given a large corpus of newspaper/newswire documents and a test set of questions (open domain)
- a restricted class of types for questions
- each question was guaranteed to have at least one document in the collection that explicitly answered it
- the answer was guaranteed to be no more than 50 characters long

49

Example questions from TREC-9

- How much folic acid should an expectant mother get daily?
- Who invented the paper clip?
- What university was Woodrow Wilson president of?
- Where is Rider College located?
- Name a film in which Jude Law acted.
- Where do lobsters like to live?

50

More complex questions

- What is epilepsy?
- What is an annuity?
- What is Wimbledon?
- Who is Jane Goodall?
- What is the Statue of Liberty made of?
- Why is the sun yellow?

51

TREC

- Participants returned a ranked list of five [document-id, answer-string] pairs per question
- all processing was required to be strictly automatic
- part of the questions were syntactic variants of some original question

52

Variants of the same question

- What is the tallest mountain?
- What is the world's highest peak?
- What is the highest mountain in the world?
- Name the highest mountain.
- What is the name of the tallest mountain in the world?

53

Examples of answers

- What is a meerkat?
  - The meerkat, a type of mongoose, thrives in…
- What is the population of Bahamas?
  - Mr. Ingraham's charges of 'impropriety' are unlikely to excite the 245,000 people of the Bahamas
- Where do lobsters like to live?
  - The water is cooler, and lobsters prefer that

54

TREC

- Scoring
  - if the correct answer is found in the first pair, the question gets a score 1
  - if the correct answer is found in the kth pair, the score is 1/k (max k = 5)
  - if the correct answer is not found, the score is 0
  - total score for a system: an average of the scores for the questions

55

FALCON, QAS

- Harabagiu et al: FALCON: Boosting knowledge for answer engines, 2000
- Harabagiu et al: Answering complex, list and context questions with LCC's Question-Answering Server, 2001

56

FALCON

- NLP methods are used to derive the question semantics: what is the type of the answer?
- IR methods (a search engine) are used to find all text paragraphs that may contain the answer
- incorrect answers are filtered out from the answer candidates (NLP methods)

57

FALCON

- Knowledge sources:
  - old questions (different variants) and the corresponding answers
  - Wordnet: alternatives for keywords

58

FALCON: system

- Question processing
  - named entity recognition, phrases -> semantic form for the question
  - one of the words/phrases in the question may indicate the type of the answer
  - this word is mapped to the answer taxonomy (uses Wordnet) to find a semantic class for the answer type (e.g. quantity, food)
  - other words/phrases are used as keywords for a query

59

FALCON: system

- Paragraph processing
  - the question keywords are structured into a query that is passed to a search engine
  - only the text paragraphs defined by the presence of the query keywords within a window of pre-defined size (e.g. 10 lines) are retrieved
  - if too few or too many paragraphs are returned, the query is reformulated (by dropping or adding keywords)

60

FALCON: system

- Answer processing
  - each paragraph is parsed and transformed into a semantic form
  - if unifications between the question and answer semantic forms are not possible for any of the answer paragraphs, alternations of the question keywords (synonyms, morphological derivations) are considered and sent to the retrieval engine
  - the new paragraphs are evaluated

61

FALCON: system

- Logical justification
  - an answer is extracted only, if a logical justification of its correctness can be provided
  - the semantic forms of questions and answers are translated into logical forms
  - inference rules model, e.g. coreferences and some general world knowledge (WordNet)
  - if the answers cannot be justified, semantic alternations are considered and a reformulated query is sent to the search engine

62

Definition questions

- A special case of answer type is associated with questions that inquire about definitions
- there are questions having a syntactic format that indicates that the question asks for the definition of a certain concept
  - such questions are easily identified as they are matched by a set of patterns

63

Definition questions

- Some question patterns:
  - What {is|are} <phrase_to_define>?
  - What is the definition of <phrase_to_define>?
  - Who {is|was|are|were} <person_name(s)>
- Some answer patterns:
  - <phrase_to_define> {is|are}
  - <phrase_to_define>, {a|an|the}
  - <phrase_to_define> -

64

FALCON

- Clearly the best system in TREC-9 (2000)
- 692 questions
- score: 60 %
- the work has been continued within LCC as the Question-Answering Server (QAS), which participated in TREC-10

65

Insight Q/A system

- Soubbotin: Patterns of potential answer expressions as clues to the right answers, 2001
- basic idea: for each question type, there is a set of predefined patterns
  - each such an indicator pattern has a score for each (relevant) question type

66

Insight Q/A system

- first, answer candidates are retrieved (for query, the most specific words of the question are used)
- answer candidates are checked for the presence of the indicator patterns
  - candidates containing the highest-scored indicators are chosen as final answers

67

Insight Q/A system

- Preconditions for the use of the method:
  - detailed categorization of question types ("Who-Post"; "Who-Author",...)
  - a large variety of patterns for each type (e.g. for "Who-Author"-type 23 patterns)
  - a sufficiently large number of candidate answers for each question
- TREC-10 results: 68% (the best?)

68

New challenges: TREC-10

- What if the existence of an answer is not guaranteed?
  - It is not easy to recognize that an answer is not available
  - in real life applications, an incorrect answer may be worse than not returning an answer at all

69

New challenges

- each question may require information from more than one document
  - Name 10 countries that banned beef imports from Britain in the 1990s.
- follow-up questions
  - Which museum in Florence was damaged by a major bomb explosion in 1993?
  - On what day did this happen?

70

Question-answering in a closed domain

- In TREC competitions, the types of questions belonged to some closed class, but the topics did not belong to any specific domain (open-domain)
- in practice, a question-answering system may be particularly helpful in some closed well-known domain, like within some company

71

Question-answering in a closed domain

- special features
  - the questions can have any type, and they may have errors and spoken-language expressions
  - the same questions (variants) probably occur regularly -> extensive use of old questions
  - closed domain: extensive use of domain-knowledge feasible
    - ontologies, thesauri, inference rules

72

QA vs IE

- open domain, closed domain?
- IE: static task definition, QA: question defines (dynamically) the task
- IE: structured answer ("database record"), QA: answer is a fragment of text
  - in future: also in QA more exact answers
- many similar modules can be used
  - language analysis, general semantics (WordNet)

73

News event detection and tracking

- We would like to have a system that
  - reads news streams (e.g. from news agencies)
  - detects significant events
  - presents the contents of the events to the user as compactly as possible
  - alerts if new events occur
  - gives the user the possibility to follow the development of some user-selected events
    - the system alerts if follow-up news appear

74

Event detection and tracking

- What is an event?
  - something that happens in some place at some time
  - e.g. elections in Zimbabwe in 2002
- an event usually represents some topic
  - e.g. elections
- the definition of event is not always clear
  - an event may later split to several subtopics

75

Event detection and tracking

- For each new text:
  - decision: is this text about a new event?
  - if not, to which existing event chain does it belong?
- methods: text categorization, clustering
- similarity metrics
- also: language analysis
  - name recognition: proper names, locations, time expressions

76

Event detection and tracking vs IE vs QA

- No query, no task definition
  - the user may choose some event chain to follow, but the system has to be prepared to follow any chain
- open domain, WordNet could be used for measuring the similarity of two texts
- analysis of news stories
  - name recognition: proper names, time expressions, locations etc. important in all

77

Closing

- What did we study:
  - stages of an IE process
  - learning domain-specific knowledge (extraction rules, semantic classes)
  - IE from (semi)structured text
  - some related approaches/applications

78

Closing

- Exam: next week on Wednesday 27.3. at 16-20 (Auditorio)
  - alternative: on Tuesday 26.3. at 16-20 (Auditorio)
- some model answers for exercises will appear soon
- remember Course feedback / Kurssikysely!

79