WHISK: Learning IE Rules for Semi-structured and Free Text

Roadmap

- Information Extraction
- WHISK Rule Representation
- The WHISK Algorithm
- Interactive Preparation of Training
- Empirical Results

Information Extraction System

- IE System can serve
  - as a front end for high precision information retrieval and text routing
  - as a forst step in knowledge discovery systems
  - as input to an intelligent agent
- IE Systems have been developed for writing styles ranging from structured text with tabular information to free text such as news stories
- A key element of such systems is a set of text extraction rules

IE System and Text

- For structured text
  - Specify a fixed order of relevant information and the labels or HTML tags that delimit strings to be extracted
- For free text
  - Need several steps: syntactic analysis, semantic tagging, recognizer for domain objects such as person and company names, and discourse processinf
- Semi-structured text falls between these extremes

Semi-structured Text

- Ungrammatical, Telegraphic in style, No rigid format
- Capitol Hill – 1 br twnhme. Fplc D/W/W/D. Undrgnd pkg incl $675. 3 BR, upper flr of turn of ctry HOME. Incl gar, N. Hill Loc $995. (206) 999-9999
 (This ad last ran on 08/03/97.)<hr>
- Rental:
  - Neighborhood: Capitol Hill
  - Bedrooms: 1
  - Price: 675

Rental:

- Neighborhood: Capitol Hill
- Bedrooms: 3
- Price: 995

Free Text

- Input text:
  - C. Vincent Protho, chairman and chief exedcutive officer of this maker of semiconductors, was named to the additional post of president,m succeeding John W. Smith, who resigned to pursue other interests.
- Succession event
  - PersonIn: C. Vincent Protho
  - PersonOut: John W. Smith
  - Post: President
- Mr. Adams, former president of X Corp., was named CEO of Y Inc.

Roadmap

- Information Extraction
- WHISK Rule Representation
- The WHISK Algorithm
- Interactive Preparation of Training
- Empirical Results

Rules for structured and semi-structured text

- WHISK rules are based on a form of regular expression patterns

ID:: 1
Pattern:: * (*Digit*) 'BR' * ,$' (*Number*)
OutPut:: Rental {Bedrooms $1} {Price $2}

- The rule is re-applied starting from the last character matched by the prior application of the rule

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

- WHISK rules allow a form of disjunction

Bdrm = (brs|br|bds|bdrm|bd|bedrooms|bedroom|bed)

ID:: 2
Pattern:: * (*Nghbr*) * (*Digit*) ' ' *Bdrm* * '$' (*Number*)
Output:: Rental {Neighborhood $1} {Bedrooms $2} {Price $}

Extensions of the Rules for grammatical Text

- Needs
  - Syntactic analyzer
  - Entity recognizer

@S[
{SUBJ @PN[C. Vincent Protho]PN , @PS[chairman and chief excutive officer]
of this maker of semiconductots, }
{VB @Passive was named @nam }
{PP to the additional post of @PS[president]PS , }
{REL_V succeeding @succeed @PN[John W. Smith]PN ,
who resigned @resign to pursue @pursu other interests. }
]@S 8910130051-1

ID:: 3
Pattern:: * (Person) * ‘@Passive‘ *F ‘named‘ * {PP *F (Position) * ‘@succeed , (Person)
Output:: Succession {PersonIn $1} {Post $2} {PersonOut $3}

Roadmap

- Information Extraction
- WHISK Rule Representation
- The WHISK Algorithm
- Interactive Preparation of Training
- Empirical Results

The WHISK Algorithm

- The WHISK Algorithm
  - Is a Supervised Learning Algorithm
  - Requires a set of hand-tagged training instances
  - Presents user with a batch of instances to tag
  - Induces a set of rules from the expanded training set
- WHISK begins with a reservoir of untagged instances and an empty training set of tagged instances
- At each iteration of WHISK a set of untagged instances are selected from reservoir and presented to the user to annotate
- The user adds a tag for each case frame to be extracted from the instance

```
@S[
      Capitol Hill – 1 br twnhne. 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 <br>
      <i> <font size=-2> (This ad last ran on 08/03/97.) </font> </i> <hr>
]@S 5
@@TAGS Rental {Neighborhood Capitol Hill} {Bedrooms 1} {Price 675}
@@TAGS Rental {Neighborhood Capitol Hill} {Bedrooms 3} {Price 995}
```

```
WHISK(Resorvoir)
  RuleSet = NULL
  Training = NULL
  Repeat at user's request
    Select a batch 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)
  Prune RuleSet
```

Anchoring the Extraction Slots

Empty Rule: “ * (*) * (*) * (*) * “

Anchoring Slot 1:
- Base_1: * (*Nghbr*)
- Base_2: ‘@start‘ (*) ‘ -‘

Anchoring Slot 2:
- Base_1: * (*Nghbr*) * (*Digit*)
- Base_2: * (*Nghbr*) * ‘- ‘ (*) ‘ br‘

Anchoring Slot 3:
- Base_1: * (*Nghbr*) * (*Digit*) * (*Number*)
- Base_2: * (*Nghbr*) * (*Digit*) * ‘$‘ (*) ‘.‘

Adding Terms to a Proposed Rule

- WHISK tries adding either the term itself or its semantic class to the rule
  - Each word, number, punctuation, HTML tag
  - Line breaks, line beginning with indentation, line followed by colon, blanklines
  - WHISK prefers terms near extraction boundaries
  - WHISK can be given a window size of *k* tokens and only consider termswithin *k* of an extraction slot

```
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)

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
```

Laplacian = (e +1) / (n +1), where e is the number of errors and n is the number of extractions made on the training set

```
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
```

Example: Error tolerance threshold is set to 0.10,
a rule that applies 20 times with 1 error (L = 0.095) will be accepted unless an extension is found that covers 10 or more with 0 errors (L = 0.091). If the best extension has coverage of only 5 with 0 errors (L=0.167) this is not considered a more reliable rule ans WHISK keeps the rule with coverage 20 instead.

Roadmap

- Information Extraction
- WHISK Rule Representation
- The WHISK Algorithm
- Interactive Preparation of Training
- Empirical Results

Interactive Preparation of Training

- Selecting informative instances
  - In each iteration of WHISK, a batch of instances is selected from the reservoir of untagged instances, presented to the user for tagging, and then added to the training set

  Instances covered by an existing rule

  Instances that are near misses of a rule

  Instances not covered by any rule
- When to stop tagging?

Roadmap

- Information Extraction
- WHISK Rule Representation
- The WHISK Algorithm
- Interactive Preparation of Training
- Empirical Results

Test Domains

- Structured texts:
  - CNN weather forecast wb pages
  - BigBook seachable telephone directory
- Semi-structured texts:
  - Rental Ads
  - Seminar Announcements
  - Software Jobs
- Free texts:
  - Management Succession from Wall Street Journal articles

Methods and Metrics

- Recall = TP / (TP + FN)
- Precision = TP / (TP + FP)
- Accuracy = (TP + TN) / (TP + TN + FP + FN)

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

Results for Structured Texts

- Structured Text: 100 % in Recall and Precision

<TD NOWRAP><FONT SIZE=+1> Thursday </FONT>
<IMG SRC="/WEATHER/images/pcloudy.jpg" ALT="partly cloudy" WIDTH=64 HEIGHT=64>
 <FONT SIZE=-1> partly cloudy </FONT>
 <FONT SIZE=-1> High: </FONT> <B> 29 C / 84 F </B>
 <FONT SIZE=-1> Low: </FONT> <B> 13 C / 56 F </B> </TD>

Results for Semi-structured Texts

	Unpruned		Pruned	
Slot	R	P	R	P
Start Time	100.0	86.2	100.0	96.2
End Time	87.2	85.0	87.2	89.5
Speaker	11.1	52.6	0.0	0.0
Location	55.4	83.6	36.1	93.8

Results for Free Texts

	Unpruned		Pruned	
Training	R	P	R	P
100	51.5	24.1	9.6	45.6
200	49.9	31.5	13.9	62.1
400	53.5	36.0	19.3	70.5
800	56.3	42.9	31.0	70.6
6,900	61.0	48.5	46.4	68.9