A Context Pattern Induction Method for Named Entity Extraction

Partha Pratim Talukdar
Computer & Information Science Department
University of Pennsylvania, Philadelphia
partha@cis.upenn.edu

Joint work with Thorsten Brants (Google), Mark Liberman (Penn)
and Fernando Pereira (Penn).
Named Entity Extraction

Recognition and classification of entity names e.g. people names, organization names, place names etc.

We have identified a transcriptional repressor, \textit{Nrg1}, in a genetic screen designed to reveal negative factors involved in the expression of \textit{STA1}.

\[\text{We have identified a transcriptional repressor, } \textit{Nrg1}, \text{ in a genetic screen designed to reveal negative factors involved in the expression of } \textit{STA1}.\]
Can anything be done by combining unlabeled data with partial entity lists?
Objective

To Capture Redundancy in Expression.

Unlabeled Data

Seed
Morgan-Stanley
Google

Context Pattern Inducer & Entity Extractor

Morgan Stanley
Google
Goldman-Sachs
Sun

analyst at <ENT>
companies such as <ENT>
joint venture between <ENT>

Penn
**One automaton induced for each trigger word.**
Preparing for Grammar Induction

Type of grammar: regular or context free?
Where do we start: ideally patterns should be variable length.
What about starting from a token which is specific to the context of entities: *Trigger words.*

*an increased expression of **adenosine deaminase** in vad mic e expression of a murine **adenosine deaminase** gene in rhesus monkey contrast the expression of **apolipoprotein e** mrna was greater than*
Objective:

*Automatically find out tokens which are specific to extracted entity contexts and which can indicate occurrence of entities in its neighbourhood.*

- What about frequent tokens in entire corpus?
- What about frequent tokens in extracted context?
  - These tokens can be common everywhere.
- What about those with high term weights?
  - Noise and very specific words can fill top slots.
Trigger Words: Dominating Words

• Assign term weight $W_t$ to each token in context.

• From each context segment $C_j$, find dominating word ($DW_j$), the token with highest term weight:

$$DW_j = \arg\max_t W_t, \forall t \in C_j$$

• Exactly one dominating word is selected from each context. Compute frequency (multiplicity) of these dominating words.

• Consider top $n$ as trigger words.
showed an increased expression of <ENT> in vad mice colon vivo expression of a murine <ENT> gene in rhesus monkey hematopoietic plasmodium falciparum expression of the <ENT> gene in mouse l cells in contrast the expression of <ENT> mrna was greater than that

<table>
<thead>
<tr>
<th>Token</th>
<th>Dominating Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>2</td>
</tr>
<tr>
<td>murine</td>
<td>1</td>
</tr>
<tr>
<td>falciparum</td>
<td>1</td>
</tr>
<tr>
<td>n = 1</td>
<td></td>
</tr>
</tbody>
</table>
Automata Induction

- One automaton induced for each trigger word.
- Given a token, we can uniquely identify the single state it points to: \textit{1-reversible}.
- Captures bi-gram statistics and helps combine evidence.
- Cycles are allowed.
- Induced automaton is to be used as an acceptor and not as generator.
Automaton Pruning

- Posterior score of each transition is computed using forward-backward algorithm.

- A transition is pruned if its posterior score is significantly lower than the best outgoing transition.
Automaton as Extractor

• Induced automata are used as extractors.
• Tokens that fit patterns’ slots are *candidate* entities.
• But can we directly consider candidate entity tokens as part of valid entity names?
  - No. But simple heuristics work very well.
• Only candidates who together satisfy $K [D K]^* K$ are retained *e.g.:*
  
  *physicist at the University of Pennsylvania and*

  \[
  D \quad K \quad D \quad K
  \]

  Pattern: *physicist at <ENT> and*

  Extracted Entity: *University of Pennsylvania*
Pattern Ranking

• All induced patterns are not equally good.

Positive Seed (ORG)  Negative Seed (PER)  Negative Seed (LOC)

ORG Pattern to be Ranked

Score: 5 3

• Easier when working with multiple ambiguous classes at the same time.
• Finally select top ranking $n$ patterns.
An extracted entity gets a higher score if more number of *good patterns* (ranked as shown previously) extract it.

- Good Pattern 1
- Good Pattern 2
- Good Pattern 3
- Good Pattern 4
- Good Pattern 5
  ...
  ...
- Good Pattern \( n \)

\[ \text{Entity}_60 \quad \downarrow \]
\[ \text{Entity}_8 \quad \uparrow \]
Experimental Results
Experiment with Watch Brand Names

- gold -ENT- watch
- diamond -ENT- watch
- fake -ENT- watches
- bought -ENT- watch
- encrusted -ENT- watch
- stole -ENT- watch
- Richemont AG, -ENT- watches
- Rolex and -ENT- watches
- buy -ENT- watches
- Cartier and -ENT- watches

Rolex
Cartier
Swiss
Movado
Seiko
Gucci
Patek
Piaget
Omega
Citizen
...

Penn
English Organization Name Experiment

- analyst at -ENT-.  
- companies such as -ENT-.  
- analyst with -ENT- in  
- series against the -ENT-tonight  
- Today's Schaeffer's Option Activity Watch features -ENT- (  
  - Cardinals and -ENT-,  
  - sweep of the -ENT- with  
  - joint venture with -ENT- (  
  - rivals -ENT- Inc.  
- Friday night's game against -ENT-.  

Boston Red Sox  
St. Louis Cardinals  
Chicago Cubs  
Florida Marlins  
Montreal Expos  
San Francisco Giants  
Red Sox  
Cleveland Indians  
Chicago White Sox  
Atlanta Braves  
...
English Person Name Experiment

- compatriot -ENT-.
- compatriot -ENT- in Rep. -ENT-.
- Actor -ENT- is Sir -ENT-.
- Actor -ENT-.
- Tiger Woods, -ENT- and movie starring -ENT-.
- Andre Agassi
- Lleyton Hewitt
- Ernie Els
- Serena Williams
- Andy Roddick
- Retief Goosen
- Vijay Singh
- Jennifer Capriati
- Roger Federer

More examples in the paper.
Entity List Extension Results

<table>
<thead>
<tr>
<th>Category</th>
<th>Seed Size</th>
<th>Extended Size</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>379</td>
<td>3001</td>
<td>70%</td>
</tr>
<tr>
<td>ORG</td>
<td>1597</td>
<td>33369</td>
<td>85%</td>
</tr>
<tr>
<td>PER</td>
<td>3616</td>
<td>86265</td>
<td>88%</td>
</tr>
</tbody>
</table>

- Precision is based on random evaluation of 100 entities.

- The method also works for very small seed list: watch brand name experiment with seed set size of 17.

- It is the **quality of the seed entities** (their unambiguous nature) that is more important than their number.
## Influence on Supervised CRF Tagger

### PER, LOC, ORG

<table>
<thead>
<tr>
<th>Training Data (Tokens)</th>
<th>Test-a</th>
<th>Test-b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No List</td>
<td>Seed List</td>
</tr>
<tr>
<td>9268</td>
<td>68.16</td>
<td>70.91</td>
</tr>
<tr>
<td>23385</td>
<td>78.36</td>
<td>79.21</td>
</tr>
<tr>
<td>46816</td>
<td>82.08</td>
<td>80.79</td>
</tr>
<tr>
<td>92921</td>
<td>85.34</td>
<td>83.03</td>
</tr>
<tr>
<td>203621</td>
<td>89.71</td>
<td>84.50</td>
</tr>
</tbody>
</table>

### PER, LOC, ORG, MISC

<table>
<thead>
<tr>
<th>Training Data (Tokens)</th>
<th>Test-a</th>
<th>Test-b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No List</td>
<td>Seed List</td>
</tr>
<tr>
<td>9229</td>
<td>68.27</td>
<td>70.93</td>
</tr>
<tr>
<td>204657</td>
<td>89.52</td>
<td>84.30</td>
</tr>
</tbody>
</table>

*Test Data Sizes: Test-a 51362 tokens, Test-b 46435 tokens*
Related Work

• Most of the previous methods ([Riloff & Jones ’99], generic extractor in [Etzioni et.al. ‘05]) are language dependent (e.g. need chunking information) but current method is completely language independent.

• Successfully used features derived from unlabeled data (token membership in extended lists) to improve a high-performing CRF tagger.

• We report effectiveness of the algorithm on relatively large dataset of 18 billion tokens.
Future Work

• Empirical comparison with other methods.

• Better pattern and entity ranking.

• Compare to see whether features derived in this paper can complement other recent methods that also generate features from unlabeled data.

• Experiment with other languages and domains.
Thanks
Automaton Pruning (contd.)

- Which transitions to prune (remove)?
- How about taking pruning decision locally?

- There is possibility of transition (42, 41) getting pruned in some threshold based scheme when decision is taken locally.
Pruning

• For numerical stability, log probabilities are used which are processed as per following log-semiring definition:
  
  Set: [-inf, inf]
  Plus: log(exp(x) + exp(y))
  Zero: -inf
  Times: +
  One: 0

• After pruning, automata are trimmed.
• Automata are stored in AT&T FSM format.
German ORG & PER Experiment

**Organization Patterns**

Tageszeitung "-<ENT>-"  
Zeitung -<ENT>- Å»  
Aktie von -<ENT>- mit  
laut "-<ENT>-"  
Laut "-<ENT>-"  
Heimspiel gegen -<ENT>-  
empfehlen die Aktie von -<ENT>-: (vwd) - Die -<ENT>- Inc  
Bei -<ENT>- geht  
Bericht der -<ENT>- Å»  
Wie die -<ENT>- Å»  
Airlines , -<ENT>-  
berichtete die -<ENT>- Å»  
berichtet die -<ENT>- Å»  
Analysen von -<ENT>- .  
Laut -<ENT>- Å»  
Analysten von -<ENT>- stufen  
Analysten von -<ENT>- die  
Marktführer -<ENT>- .  
Klubs -<ENT>- und

**Person Patterns**

s. -<ENT>- (  
Landsmann -<ENT>- .  
Nachfolger -<ENT>- ,  
Wer -<ENT>- ?  
Landsmann -<ENT>- (  
Seite von -<ENT>- in  
Seite von -<ENT>- und  
Superstars -<ENT>- und  
7:5 , -<ENT>- (  
Kollege -<ENT>- .  
Prominente wie -<ENT>- ,  
Hollywoodstar -<ENT>- (  
Schauspielerin -<ENT>- ,  
Weltstars wie -<ENT>- ,  
Schauspieler -<ENT>- und  
Nationalspieler -<ENT>- (  
6:1 , -<ENT>- (  
Angeles (dpa) - -<ENT>- (  
verletzen -<ENT>- und  
Schauspieler -<ENT>- (  
......
Influence on Supervised Tagger

• Conditional Random Field (CRF) based tagger trained on CoNLL-2003 English data for LOC, ORG and PER names.

• Tested with and without automatically generated entity lists as additional features.

• Tested with varying amount of training data to test the hypothesis that the tagger benefits most from using unsupervised generated list when there is less training data.
Automata Induction

• All entity names are replaced by token “<ENT>”
• Only one token to the right of “<ENT>” considered.
• Cycles are allowed.
• Induced automaton is to be used as an acceptor and not as generator.
• Each transition is initially scored as follows:

\[
Score(a_i, a_j) = \frac{TransCount(a_i, a_j)}{\sum_k TransCount(a_i, a_k)}
\]