A Multi-Domain Web-Based Algorithm for POS Tagging of Unknown Words

Shulamit Umansky-Pesin, Roi Reichart, Ari Rappoport

The Hebrew University of Jerusalem
Outline

- Introduction
- Algorithm
- Experimental results
- Conclusions
Part-Of-Speech tagging

- The POS tagging problem
  - Determine the POS tag for a particular instance of a word

- Supervised taggers perform well:
  - Toutanova et al., 2003: 97.24% overall accuracy on WSJ corpus
  - But only 89.04% accuracy on unknown words
Domain adaptation

• The training and test corpora are from different domains
• Number of unknown words increases
• The total and unknown words accuracy suffers:
  – Tagging GENIA: 80.12% accuracy on unknown words
  – Tagging BNC: 68.71% accuracy on unknown words
Previous approaches

• Unknown words treatment:
  – Orthographical data (capital letters, digits, hyphens)
  – Prefixes and suffixes
  – Language-specific hand-crafted morphological and syntactic features
  – External data (lexicons etc.)
Previous approaches

• Domain adaptation:
  – Daume III, 2007 – manually labeled corpus from target domain
  – Blitzer et al., 2006 – unlabeled corpus from target domain

• Target domain is not always well-defined (for example, web)

• Preparing a corpus is time-consuming, labeling it is much more so.
Outline

• Introduction
• Algorithm
• Experimental results
• Conclusions
Web search and context

• “You shall know a word by the company it keeps” (John Rupert Firth, 1957)

• Retrieve the “company” from the web
  – Who else “keeps the same company” (replacement)
  – The “company” on one side given the word and “company” on the other side (left-side and right-side contexts)
UV irradiation and H2O2 treatment of T lymphocytes ...

irradiation and * treatment of

Replacement in context

Wouldn’t work alone!
Web search and context

“UV irradiation and H2O2 treatment of T lymphocytes …”

Left-side context

“* * H2O2 treatment of”

Right-side context

"irradiation and H2O2 * *"

by an indicated that enhanced by familiar with observed after

H2O2 treatment of

on comparison on Fe treatment by cause an does not
POS tagger

• Maximum Entropy tagger - reimplementation of MxPOST (Ratnaparkhi, 1996)
• Training phase left unchanged
• Original (Ratnaparkhi, 1996) features used
• POS tag is determined by 2-words context
MaxEnt features

<table>
<thead>
<tr>
<th>Condition</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_i$ is not rare</td>
<td>$w_i = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td>$w_i$ is rare</td>
<td>$X$ is prefix of $w_i$, $</td>
</tr>
<tr>
<td></td>
<td>$X$ is suffix of $w_i$, $</td>
</tr>
<tr>
<td></td>
<td>$w_i$ contains number &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_i$ contains uppercase character &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_i$ contains hyphen &amp; $t_i = T$</td>
</tr>
<tr>
<td>$\forall w_i$</td>
<td>$t_{i-1} = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$t_{i-2}t_{i-1} = XY$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_{i-1} = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_{i-2} = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_{i+1} = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_{i+2} = X$ &amp; $t_i = T$</td>
</tr>
</tbody>
</table>

Table 1: Features on the current history $h_i$
MaxEnt tagger - reminder

Context $h_i$ -> Tagging algorithm -> Tag probability $P(t_i|h_i)$

<table>
<thead>
<tr>
<th>UV</th>
<th>irradiation and H2O2 treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNP</td>
<td>NN</td>
</tr>
<tr>
<td>IN</td>
<td>NN</td>
</tr>
</tbody>
</table>

CC 0.995 NNP 0.005
MaxEnt tagger - reminder

Tagging algorithm

Context $h_i$ → Context → Features $f_j(h,t)$ → Features’ weights $\alpha_j$ → Tag probability $P(t_i|h_i)$ → Training data → $p(h,t) = Z \prod_{j=1}^{k} \alpha_j f_j^{(h,t)}$
MaxEnt - reminder

- At each step maintain a list N of tag sequences:
  - `UV_NNP irradiation_NN`
  - `UV_NNP irradiation_NNP`
- For each candidate sequence of tags
  - Extract features for the new word ("and")
  - For each possible* tag
    - Calculate tag conditional probability $P(t_i|h_i)$ using the features parameters learned in training
    - Calculate sequence conditional probability $P(t_1.. t_i|h_1...h_i)$
- Select N top-scoring sequences
- Repeat

*possible tags:
• All tags for unknown words
• Only tags seen in training for known words
Unknown words & web search

- irradiation
- and
- H2O2 treatment
- of
- by
- an
- H2O2 treatment
- of
- indicated that
- H2O2 treatment
- of
- irradiation and
- H2O2 on comparison
- irradiation and
- H2O2 treatment by

Tagging algorithm

Original context

Web contexts

P(t_i|h_i^{original})
P(t_i|h_1^{i})
P(t_i|h_2^{i})
P(t_i|h_3^{i})
P(t_i|h_4^{i})

Average

Final tag probability

P(t_i|h_i)
• Additional steps:
  – Collect left- and right-side contexts and replacements from the web and create new words sequences
  – For each new words sequence $h'_i$
    • For each tag
      – Calculate tag conditional probability $P(t_i|h'_i)$ using the features from the new context
    – Calculate final tag probability as the average between all $P(t_i|h'_i)$ and the original $P(t_i|h_i)$
Outline

• Introduction
• Algorithm
• Experimental results
• Conclusions
Experimental setup

- Unknown words threshold: 5
- Baseline: MxPOST tagger
## Experimental setup - English

<table>
<thead>
<tr>
<th>Name</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ</td>
<td>WSJ 2-21</td>
<td>WSJ 23</td>
</tr>
<tr>
<td>GENIA (domain adaptation)</td>
<td>WSJ 2-21</td>
<td>2000 sentences sample from GENIA</td>
</tr>
<tr>
<td>BNC (domain adaptation)</td>
<td>WSJ 2-21</td>
<td>2000 sentences sample from BNC</td>
</tr>
</tbody>
</table>
## Results - English

### Unknown words accuracy

<table>
<thead>
<tr>
<th></th>
<th>WSJ</th>
<th>GENIA</th>
<th>BNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>88.79%</td>
<td>80.12%</td>
<td>68.71%</td>
</tr>
<tr>
<td>Web-assisted</td>
<td>89.86%</td>
<td>83.00%</td>
<td>72.12%</td>
</tr>
<tr>
<td>Improvement</td>
<td>1.07%</td>
<td>2.88%</td>
<td>3.41%</td>
</tr>
<tr>
<td>Error reduction</td>
<td>9.54%</td>
<td>14.48%</td>
<td>10.89%</td>
</tr>
</tbody>
</table>
## Experimental setup - German

<table>
<thead>
<tr>
<th>Name</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negra</td>
<td>15689 NEGRA sentences</td>
<td>2096 NEGRA sentences</td>
</tr>
<tr>
<td>Tiger (domain adaptation)</td>
<td>15689 NEGRA sentences</td>
<td>2000 TIGER sentences</td>
</tr>
<tr>
<td>Negra (domain adaptation)</td>
<td>15689 TIGER sentences</td>
<td>2096 NEGRA sentences</td>
</tr>
</tbody>
</table>
## Results - German

### Unknown words accuracy

<table>
<thead>
<tr>
<th></th>
<th>Negra domain adaptation</th>
<th>Tiger domain adaptation</th>
<th>Negra domain adaptation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td>91.06%</td>
<td>87.88%</td>
<td>87.86%</td>
</tr>
<tr>
<td><strong>Web-assisted</strong></td>
<td>91.95%</td>
<td>89.01%</td>
<td>89.84%</td>
</tr>
<tr>
<td><strong>Improvement</strong></td>
<td>0.89%</td>
<td>1.13%</td>
<td>1.98%</td>
</tr>
<tr>
<td><strong>Error reduction</strong></td>
<td>9.95%</td>
<td>9.32%</td>
<td>16.3%</td>
</tr>
</tbody>
</table>
# Experimental setup - Chinese

<table>
<thead>
<tr>
<th>Name</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTB</td>
<td>14903 CTB sentences</td>
<td>1945 CTB sentences</td>
</tr>
</tbody>
</table>
### Results - Chinese

**Unknown words accuracy**

<table>
<thead>
<tr>
<th></th>
<th>CTB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td>78.03%</td>
</tr>
<tr>
<td><strong>Web-assisted</strong></td>
<td>80.75%</td>
</tr>
<tr>
<td><strong>Improvement</strong></td>
<td>2.72%</td>
</tr>
<tr>
<td><strong>Error reduction</strong></td>
<td>12.28%</td>
</tr>
</tbody>
</table>
Outline

• Introduction
• Algorithm
• Experimental results
• Conclusions
Conclusions

• No preprocessing steps!
• Train once, tag anything – no knowledge about domain is required
• Language-independent
• Can be adapted to suit other taggers
What about Hebrew?
What about Hebrew?

- Some additional segmentation of web matches is required
- Other than that… should work!
Thank you