CSEP 517
Natural Language Processing
Autumn 2013

Parts of Speech and Feature Rich Sequence Models

Luke Zettlemoyer - University of Washington

[Many slides from Dan Klein]
Overview

- POS Tagging
- Feature Rich Techniques
  - Maximum Entropy Markov Models (MEMMs)
  - Structured Perceptron
  - Conditional Random Fields (CRFs)
One basic kind of linguistic structure: syntactic word classes

- **Open class (lexical) words**
  - **Nouns**
    - Proper: *IBM, Italy*
    - Common: *cat / cats, snow*
  - **Verbs**
    - Main: *see, registered*
  - **Adjectives**
    - *yellow*
  - **Adverbs**
    - *slowly*
  - **Numbers**
    - *122,312, one*
  - **Prepositions**
    - *to with*
  - **Particles**
    - *off up*

- **Closed class (functional)**
  - **Determiners**
    - *the, some*
  - **Conjunctions**
    - *and or*
  - **Pronouns**
    - *he, its*
Penn Treebank POS: 36 possible tags, 34 pages of tagging guidelines.

<table>
<thead>
<tr>
<th>POS</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>conjunction, coordinating</td>
<td>and both but either or mid-1890 nine-thirty 0.5 one</td>
</tr>
<tr>
<td>CD</td>
<td>numeral, cardinal</td>
<td>a all an every no that the there</td>
</tr>
<tr>
<td>DT</td>
<td>determiner</td>
<td>gemeinschaft hund ich jeux among whether out on by if third ill-mannered</td>
</tr>
<tr>
<td>EX</td>
<td>existential</td>
<td>regrettable braver cheaper taller bravest cheapest tallest can may might</td>
</tr>
<tr>
<td>FW</td>
<td>foreign word</td>
<td>will would cabbage thermostat investment subhumanity</td>
</tr>
<tr>
<td>IN</td>
<td>preposition or conjunction, subordinating</td>
<td>Motown Cougar Yvette Liverpool Americans Materials States</td>
</tr>
<tr>
<td>JJ</td>
<td>adjective or numeral, ordinal</td>
<td>undergraduates bric-a-brac averages 's hers himself it we them</td>
</tr>
<tr>
<td>JJR</td>
<td>adjective, comparative</td>
<td>her his mine my our ours their thy your occasionally maddeningly</td>
</tr>
<tr>
<td>JJS</td>
<td>adjective, superlative</td>
<td>adventurously further gloomier heavier less-perfectly best biggest</td>
</tr>
<tr>
<td>MD</td>
<td>modal auxiliary</td>
<td>nearest worst aboard away back by on open through to huh howdy uh</td>
</tr>
<tr>
<td>NN</td>
<td>noun, common, singular or mass</td>
<td>whammo shucks heck ask bring fire see take pleaded swiped registered</td>
</tr>
<tr>
<td>NNP</td>
<td>noun, proper, singular</td>
<td>saw stirred focusing approaching erasing dilapidated imitated reunifed</td>
</tr>
<tr>
<td>NNS</td>
<td>noun, common, plural</td>
<td>unsettled twist appear comprise mold postpone bases reconstructs marks</td>
</tr>
<tr>
<td>POS</td>
<td>genitive marker</td>
<td>uses that what whatever which whichever that what whatever which whom</td>
</tr>
<tr>
<td>PRP</td>
<td>pronoun, personal</td>
<td>whose however whenever where why</td>
</tr>
<tr>
<td>PRP$</td>
<td>pronoun, possessive</td>
<td></td>
</tr>
<tr>
<td>RB</td>
<td>adverb</td>
<td></td>
</tr>
<tr>
<td>RBR</td>
<td>adverb, comparative</td>
<td></td>
</tr>
<tr>
<td>RBS</td>
<td>adverb, superlative</td>
<td></td>
</tr>
<tr>
<td>RP</td>
<td>particle</td>
<td></td>
</tr>
<tr>
<td>TO</td>
<td>&quot;to&quot; as preposition or infinitive marker</td>
<td></td>
</tr>
<tr>
<td>UH</td>
<td>interjection</td>
<td></td>
</tr>
<tr>
<td>VB</td>
<td>verb, base form</td>
<td></td>
</tr>
<tr>
<td>VBD</td>
<td>verb, past tense</td>
<td></td>
</tr>
<tr>
<td>VBG</td>
<td>verb, present participle or gerund</td>
<td></td>
</tr>
<tr>
<td>VBN</td>
<td>verb, past participle</td>
<td></td>
</tr>
<tr>
<td>VBP</td>
<td>verb, present tense, not 3rd person singular</td>
<td></td>
</tr>
<tr>
<td>VBZ</td>
<td>verb, present tense, 3rd person singular</td>
<td></td>
</tr>
<tr>
<td>WDT</td>
<td>WH-determiner</td>
<td></td>
</tr>
<tr>
<td>WP</td>
<td>WH-pronoun</td>
<td></td>
</tr>
<tr>
<td>WP$</td>
<td>WH-pronoun, possessive</td>
<td></td>
</tr>
<tr>
<td>WRB</td>
<td>Wh-adverb</td>
<td></td>
</tr>
</tbody>
</table>

Part-of-Speech Ambiguity

- Words can have multiple parts of speech

Fed raises interest rates 0.5 percent

Mrs./NNP Shaefer/NNP never/RB got/VBD around/RP to/TO joining/VBG
All/DT we/PRP gotta/VBN do/VB is/VBZ go/VB around/IN the/DT corner/NN
Chateau/NNP Petrus/NNP costs/VBZ around/RB 250/CD

- Two basic sources of constraint:
  - Grammatical environment
  - Identity of the current word

- Many more possible features:
  - Suffixes, capitalization, name databases (gazetteers), etc…
Why POS Tagging?

- Useful in and of itself (more than you’d think)
  - Text-to-speech: record, lead
  - Lemmatization: saw[v] → see, saw[n] → saw
  - Quick-and-dirty NP-chunk detection: grep {JJ | NN}* {NN | NNS}

- Useful as a pre-processing step for parsing
  - Less tag ambiguity means fewer parses
  - However, some tag choices are better decided by parsers

```
IN
DT NNP NN VBD VBN RP NN NNS
The Georgia branch had taken on loan commitments …

VDN
DT NN IN NN VBD NNS VBD
The average of interbank offered rates plummeted …
```
Baselines and Upper Bounds

- **Choose the most common tag**
  - 90.3% with a bad unknown word model
  - 93.7% with a good one

- **Noise in the data**
  - Many errors in the training and test corpora
  - Probably about 2% guaranteed error from noise (on this data)
Overview: Accuracies

- Roadmap of (known / unknown) accuracies:
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%

- TnT (Brants, 2000):
  - A carefully smoothed trigram tagger
  - Suffix trees for emissions
  - 96.7% on WSJ text (SOA is ~97.5%)

- Upper bound: ~98%
## Common Errors

- **Common errors [from Toutanova & Manning 00]**

<table>
<thead>
<tr>
<th>JJ</th>
<th>NN</th>
<th>NNP</th>
<th>NNPS</th>
<th>RB</th>
<th>RP</th>
<th>IN</th>
<th>VB</th>
<th>VBD</th>
<th>VBN</th>
<th>VBP</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>177</td>
<td>56</td>
<td>0</td>
<td>61</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>108</td>
<td>0</td>
<td>488</td>
</tr>
<tr>
<td>244</td>
<td>0</td>
<td>103</td>
<td>0</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>29</td>
<td>5</td>
<td>6</td>
<td>19</td>
<td>525</td>
</tr>
<tr>
<td>107</td>
<td>106</td>
<td>0</td>
<td>132</td>
<td>5</td>
<td>0</td>
<td>7</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>427</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>142</td>
</tr>
<tr>
<td>72</td>
<td>21</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>138</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>295</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>39</td>
<td>0</td>
<td>65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>104</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>169</td>
<td>103</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>323</td>
</tr>
<tr>
<td>17</td>
<td>64</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>85</td>
<td>189</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>143</td>
<td>2</td>
<td>166</td>
</tr>
<tr>
<td>101</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>108</td>
<td>0</td>
<td>1</td>
<td>221</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>34</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>49</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>104</td>
</tr>
<tr>
<td>626</td>
<td>536</td>
<td>348</td>
<td>144</td>
<td>317</td>
<td>122</td>
<td>279</td>
<td>102</td>
<td>140</td>
<td>269</td>
<td>108</td>
<td>3651</td>
</tr>
</tbody>
</table>

- **NN/JJ**: official knowledge
- **NN**: made up the story
- **VBD RP/IN DT NN**: recently sold shares
What about better features?

- Choose the most common tag
  - 90.3% with a bad unknown word model
  - 93.7% with a good one

- What about looking at a word and its environment, but no sequence information?
  - Add in previous / next word
  - Previous / next word shapes
  - Occurrence pattern features
  - Crude entity detection
  - Phrasal verb in sentence?
  - Conjunctions of these things

- Uses lots of features: > 200K
Overview: Accuracies

- Roadmap of (known / unknown) accuracies:
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%
  - TnT (HMM++): 96.2% / 86.0%
  - Maxent $P(s_i|x)$: 96.8% / 86.8%

- Q: What does this say about sequence models?
- Q: How do we add more features to our sequence models?

- Upper bound: ~98%
MEMM Taggers

- **One step up**: also condition on previous tags

  \[ p(s_1 \ldots s_m|x_1 \ldots x_m) = \prod_{i=1}^{m} p(s_i|s_1 \ldots s_{i-1}, x_1 \ldots x_m) \]

  \[ = \prod_{i=1}^{m} p(s_i|s_{i-1}, x_1 \ldots x_m) \]

- Train up \( p(s_i|s_{i-1}, x_1 \ldots x_m) \) as a discrete log-linear (maxent) model, then use to score sequences

  \[ p(s_i|s_{i-1}, x_1 \ldots x_m) = \frac{\exp \left( w \cdot \phi(x_1 \ldots x_m, i, s_{i-1}, s_i) \right)}{\sum_{s'} \exp \left( w \cdot \phi(x_1 \ldots x_m, i, s_{i-1}, s') \right)} \]

- This is referred to as an MEMM tagger [Ratnaparkhi 96]
- Beam search effective! (Why?)
- What’s the advantage of beam size 1?
The HMM State Lattice / Trellis (repeat slide)

START       Fed           raises       interest         rates         STOP

q(N|V)  e(Fed|N)  q(V|N)  e(raises|V)  e(interest|V)  q(V|V)  e(rates|J)  q(J|V)  e(STOP|V)
The MEMM State Lattice / Trellis

x = START       Fed           raises       interest       rates      STOP

p(V|V,w)
p(V|N,w)
p(N|x)
p(J|V,w)
p(V|J,w)
p(D|J,w)
p($|D,w)$

x = START       Fed           raises       interest       rates      STOP

p(V|V,w)
p(V|N,w)
p(N|x)
p(J|V,w)
p(V|J,w)
p(D|J,w)
p($|D,w)$
Decoding

- Decoding maxent taggers:
  - Just like decoding HMMs
  - Viterbi, beam search, posterior decoding

- Viterbi algorithm (HMMs):
  - Define $\pi(i,s_i)$ to be the max score of a sequence of length $i$ ending in tag $s_i$
    \[
    \pi(i, s_i) = \max_{s_{i-1}} e(x_i | s_i) q(s_i | s_{i-1}) \pi(i - 1, s_{i-1})
    \]

- Viterbi algorithm (Maxent):
  - Can use same algorithm for MEMMs, just need to redefine $\pi(i,s_i)$!
    \[
    \pi(i, s_i) = \max_{s_{i-1}} p(s_i | s_{i-1}, x_1 \ldots x_m) \pi(i - 1, s_{i-1})
    \]
Overview: Accuracies

- Roadmap of (known / unknown) accuracies:
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%
  - TnT (HMM++): 96.2% / 86.0%
  - Maxent $P(s_i|x)$: 96.8% / 86.8%
  - MEMM tagger: 96.9% / 86.9%

- Upper bound: ~98%
Global Discriminative Taggers

- Newer, higher-powered discriminative sequence models
  - CRFs (also perceptrons, M3Ns)
  - Do not decompose training into independent local regions
  - Can be deathly slow to train – require repeated inference on training set

- Differences can vary in importance, depending on task

- However: one issue worth knowing about in local models
  - “Label bias” and other explaining away effects
  - MEMM taggers’ local scores can be near one without having both good “transitions” and “emissions”
  - This means that often evidence doesn’t flow properly
  - Why isn’t this a big deal for POS tagging?
  - Also: in decoding, condition on predicted, not gold, histories
Linear Models: Perceptron

The perceptron algorithm

- Iteratively processes the training set, reacting to training errors
- Can be thought of as trying to drive down training error

The (online) perceptron algorithm:

- Start with zero weights
- Visit training instances \((x_i, y_i)\) one by one
  - Make a prediction
    \[ y^* = \arg \max_y w \cdot \phi(x_i, y) \]
  - If correct \((y^* = y_i)\): no change, goto next example!
  - If wrong: adjust weights
    \[ w = w + \phi(x_i, y_i) - \phi(x_i, y^*) \]

Challenge: How to compute \(\arg\max\) efficiently?

[Collins 02]
Decoding

- **Linear Perceptron**
  \[ s^* = \arg \max_s w \cdot \Phi(x, s) \cdot \theta \]
  - Features must be local, for \( x=x_1 \ldots x_m \), and \( s=s_1 \ldots s_m \)
  \[ \Phi(x, s) = \sum_{j=1}^{m} \phi(x, j, s_{j-1}, s_j) \]
  - Define \( \pi(i, s_i) \) to be the max score of a sequence of length \( i \) ending in tag \( s_i \)
  \[ \pi(i, s_i) = \max_{s_{i-1}} w \cdot \phi(x, i, s_{i-1}, s_i) + \pi(i - 1, s_{i-1}) \]

- **Viterbi algorithm (HMMs):**
  \[ \pi(i, s_i) = \max_{s_{i-1}} e(x_i \mid s_i) q(s_i \mid s_{i-1}) \pi(i - 1, s_{i-1}) \]

- **Viterbi algorithm (Maxent):**
  \[ \pi(i, s_i) = \max_{s_{i-1}} p(s_i \mid s_{i-1}, x_1 \ldots x_m) \pi(i - 1, s_{i-1}) \]
Overview: Accuracies

- Roadmap of (known / unknown) accuracies:
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%
  - TnT (HMM++): 96.2% / 86.0%
  - Maxent $P(s_i|x)$: 96.8% / 86.8%
  - MEMM tagger: 96.9% / 86.9%
  - Perceptron: 96.7% / ??

- Upper bound: ~98%
Conditional Random Fields (CRFs) [Lafferty, McCallum, Pereira 01]

- **Maximum entropy (logistic regression)**

  \[
  p(y|x; w) = \frac{\exp (w \cdot \phi(x, y))}{\sum_{y'} \exp (w \cdot \phi(x, y'))}
  \]

  - **Learning**: maximize the (log) conditional likelihood of training data \( \{(x_i, y_i)\}_{i=1}^n \)

    \[
    \frac{\partial}{\partial w_j} L(w) = \sum_{i=1}^n \left( \phi_j(x_i, y_i) - \sum_y p(y|x_i; w) \phi_j(x_i, y) \right) - \lambda w_j
    \]

- **Computational Challenges?**

  - Most likely tag sequence, normalization constant, gradient
Decoding

- CRFs
  - Features must be local, for $x=x_1 \ldots x_m$, and $s=s_1 \ldots s_m$.

$$p(s|x; w) = \frac{\exp (w \cdot \Phi(x, s))}{\sum_{s'} \exp (w \cdot \Phi(x, s'))}$$

$$\Phi(x, s) = \sum_{j=1}^{m} \phi(x, j, s_{j-1}, s_j)$$

$$\arg \max_{s} \frac{\exp (w \cdot \Phi(x, s))}{\sum_{s'} \exp (w \cdot \Phi(x, s'))} = \arg \max_{s} \exp (w \cdot \Phi(x, s)) = \arg \max_{s} w \cdot \Phi(x, s)$$

- Same as Linear Perceptron!!!

$$\pi(i, s_i) = \max_{s_{i-1}} \phi(x, i, s_{i-1}, s_i) + \pi(i - 1, s_{i-1})$$
CRFs: Computing Normalization

\[
p(s|x; w) = \frac{\exp(w \cdot \Phi(x, s))}{\sum_{s'} \exp(w \cdot \Phi(x, s'))} \quad \Phi(x, s) = \sum_{j=1}^{m} \phi(x, j, s_{j-1}, s_{j})
\]

\[
\sum_{s'} \exp(w \cdot \Phi(x, s')) = \sum_{s'} \exp \left( \sum_{j} w \cdot \phi(x, j, s_{j-1}, s_{j}) \right) = \sum_{s'} \prod_{j} \exp(w \cdot \phi(x, j, s_{j-1}, s_{j}))
\]

Define \(\text{norm}(i,s_i)\) to sum of scores for sequences ending in position \(i\)

\[
\text{norm}(i, y_i) = \sum_{s_{i-1}} \exp(w \cdot \phi(x, i, s_{i-1}, s_i)) \cdot \text{norm}(i - 1, s_{i-1})
\]

- **Forward Algorithm! Remember HMM case:**

\[
\alpha(i, y_i) = \sum_{y_{i-1}} e(x_i|y_i)q(y_i|y_{i-1}) \cdot \alpha(i - 1, y_{i-1})
\]

  - Could also use backward?
CRFs: Computing Gradient

\[ p(s|x; w) = \frac{\exp(w \cdot \Phi(x, s))}{\sum_{s'} \exp(w \cdot \Phi(x, s'))} \quad \Phi(x, s) = \sum_{j=1}^{m} \phi(x, j, s_{j-1}, s_j) \]

\[ \frac{\partial}{\partial w_j} L(w) = \sum_{i=1}^{n} \left( \Phi_j(x_i, s_i) - \sum_s p(s|x_i; w) \Phi_j(x_i, s) \right) - \lambda w_j \]

\[ \sum_s p(s|x_i; w) \Phi_j(x_i, s) = \sum_s p(s|x_i; w) \sum_{j=1}^{m} \phi_k(x_i, j, s_{j-1}, s_j) \]

\[ = \sum_{j=1}^{m} \sum_{a,b} \sum_{s:s_{j-1}=a,s_b=b} p(s|x_i; w) \phi_k(x_i, j, s_{j-1}, s_j) \]

- Need forward and backward messages

See notes for full details!
Overview: Accuracies

- **Roadmap of (known / unknown) accuracies:**
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%
  - TnT (HMM++): 96.2% / 86.0%
  - Maxent P(s_i|x): 96.8% / 86.8%
  - MEMM tagger: 96.9% / 86.9%
  - Perceptron 96.7% / ??
  - CRF (untuned) 95.7% / 76.2%
  - Upper bound: ~98%
Cyclic Network

- Train two MEMMs, multiple together to score
- And be very careful
  - Tune regularization
  - Try lots of different features
  - See paper for full details

(a) Left-to-Right CMM

(b) Right-to-Left CMM

(c) Bidirectional Dependency Network

[Toutanova et al 03]
Overview: Accuracies

- Roadmap of (known / unknown) accuracies:
  - Most freq tag: ~90% / ~50%
  - Trigram HMM: ~95% / ~55%
  - TnT (HMM++): 96.2% / 86.0%
  - Maxent P(s_i|x): 96.8% / 86.8%
  - MEMM tagger: 96.9% / 86.9%
  - Perceptron: 96.7% / ??
  - CRF (untuned): 95.7% / 76.2%
  - Cyclic tagger: 97.2% / 89.0%
  - Upper bound: ~98%
Domain Effects

- Accuracies degrade outside of domain
  - Up to triple error rate
  - Usually make the most errors on the things you care about in the domain (e.g. protein names)

- Open questions
  - How to effectively exploit unlabeled data from a new domain (what could we gain?)
  - How to best incorporate domain lexica in a principled way (e.g. UMLS specialist lexicon, ontologies)