CS5740: Natural Language Processing
Spring 2017

Sequence Prediction and Part-of-speech Tagging

Instructor: Yoav Artzi
Overview

• POS Tagging: the problem
• Hidden Markov Models (HMM)
  – Supervised Learning
  – Inference
    • The Viterbi algorithm
• Feature-rich models
  – Maximum-entropy Markov Models
  – Perceptron
  – Conditional Random Fields
Parts of Speech

Open class (lexical) words

- **Nouns**
  - Proper: IBM, Italy
  - Common: cat / cats, snow

- **Verbs**
  - Main: see, registered

- **Adjectives**
  - old, older, oldest

- **Adverbs**
  - slowly

- **Numbers**
  - 122,312, one

- **Modals**
  - can, had

Closed class (functional)

- **Determiners**
  - the, some

- **Conjunctions**
  - and, or

- **Pronouns**
  - he, its

- **Prepositions**
  - to, with

- **Particles**
  - off, up

- **Interjections**
  - Ow, Eh

- **… more**
POS Tagging

• Words often have more than one POS: back
  – The back door = JJ
  – On my back = NN
  – Win the voters back = RB
  – Promised to back the bill = VB

• The POS tagging problem is to determine the POS tag for a particular instance of a word.
POS Tagging

• Input: Plays well with others
• Ambiguity: NNS/VBZ UH/JJ/NN/RB IN NNS
• Output: Plays/VBZ well/RB with/IN others/NNS
• Uses:
  – Text-to-speech (how do we pronounce “lead”?)
  – Can write regular expressions like (Det) Adj* N+ over the output for phrases, etc.
  – As input to or to speed up a full parser
  – If you know the tag, you can back off to it in other tasks

Penn Treebank POS tags
Penn TreeBank Tagset

• Possible tags: 45
• Tagging guidelines: 36 pages
• Newswire text
Penn TreeBank Tagset

• How accurate are taggers? (Tag accuracy)
  – About 97% currently
  – But baseline is already 90%
    • Baseline is performance of simplest possible method
      – Tag every word with its most frequent tag
      – Tag unknown words as nouns
  – Partly easy because
    • Many words are unambiguous
    • You get points for them (the, a, etc.) and for punctuation marks!
  – Upperbound: probably 2% annotation errors
Hard Cases are Hard

• 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
How Difficult is POS Tagging?

• About 11% of the word types in the Brown corpus are ambiguous with regard to part of speech
• But they tend to be very common words. E.g., *that*
  – I know *that* he is honest = IN
  – Yes, *that* play was nice = DT
  – You can’t go *that* far = RB

• 40% of the word tokens are ambiguous
The Tagset

• Wait, do we really need all these tags?
• What about other languages?
  – Each language has its own tagset
Tagsets in Different Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Source</th>
<th># Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>PADT/CoNLL07 (Hajič et al., 2004)</td>
<td>21</td>
</tr>
<tr>
<td>Basque</td>
<td>Basque3LB/CoNLL07 (Aduriz et al., 2003)</td>
<td>64</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>BTB/CoNLL06 (Simov et al., 2002)</td>
<td>54</td>
</tr>
<tr>
<td>Catalan</td>
<td>CESS-ECE/CoNLL07 (Martí et al., 2007)</td>
<td>54</td>
</tr>
<tr>
<td>Chinese</td>
<td>Penn ChineseTreebank 6.0 (Palmer et al., 2007)</td>
<td>34</td>
</tr>
<tr>
<td>Chinese</td>
<td>Sinica/CoNLL07 (Chen et al., 2003)</td>
<td>294</td>
</tr>
<tr>
<td>Czech</td>
<td>PDT/CoNLL07 (Böhmová et al., 2003)</td>
<td>63</td>
</tr>
<tr>
<td>Danish</td>
<td>DDT/CoNLL06 (Kromann et al., 2003)</td>
<td>25</td>
</tr>
<tr>
<td>Dutch</td>
<td>Alpino/CoNLL06 (Van der Beek et al., 2002)</td>
<td>12</td>
</tr>
<tr>
<td>English</td>
<td>PennTreebank (Marcus et al., 1993)</td>
<td>45</td>
</tr>
<tr>
<td>French</td>
<td>FrenchTreebank (Abeillé et al., 2003)</td>
<td>30</td>
</tr>
<tr>
<td>German</td>
<td>Tiger/CoNLL06 (Brants et al., 2002)</td>
<td>54</td>
</tr>
<tr>
<td>German</td>
<td>Negra (Skut et al., 1997)</td>
<td>54</td>
</tr>
<tr>
<td>Greek</td>
<td>GDT/CoNLL07 (Prokopiidis et al., 2005)</td>
<td>38</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Szeged/CoNLL07 (Csendes et al., 2005)</td>
<td>43</td>
</tr>
<tr>
<td>Italian</td>
<td>ISST/CoNLL07 (Montemagni et al., 2003)</td>
<td>28</td>
</tr>
<tr>
<td>Japanese</td>
<td>Verbmobil/CoNLL06 (Kawata and Bartels, 2000)</td>
<td>80</td>
</tr>
<tr>
<td>Japanese</td>
<td>Kyoto4.0 (Kurohashi and Nagao, 1997)</td>
<td>42</td>
</tr>
<tr>
<td>Korean</td>
<td>Sejong (<a href="http://www.sejong.or.kr">http://www.sejong.or.kr</a>)</td>
<td>187</td>
</tr>
<tr>
<td>Portuguese</td>
<td>Floresta Sintá(t)ica/CoNLL06 (Afonso et al., 2002)</td>
<td>22</td>
</tr>
<tr>
<td>Russian</td>
<td>SynTagRus-RNC (Boguslavsky et al., 2002)</td>
<td>11</td>
</tr>
<tr>
<td>Slovene</td>
<td>SDT/CoNLL06 (Džeroski et al., 2006)</td>
<td>29</td>
</tr>
<tr>
<td>Spanish</td>
<td>Ancora-Cast3LB/CoNLL06 (Civit and Martí, 2004)</td>
<td>47</td>
</tr>
<tr>
<td>Swedish</td>
<td>Talbanken05/CoNLL06 (Nivre et al., 2006)</td>
<td>41</td>
</tr>
<tr>
<td>Turkish</td>
<td>METU-Sabanci/CoNLL07 (Oflazer et al., 2003)</td>
<td>31</td>
</tr>
</tbody>
</table>

[Petrov et al. 2012]
The Tagset

• Wait, do we really need all these tags?
• What about other languages?
  – Each language has its own tagset
    • But why is this bad?
    • Differences in downstream tasks
    • Harder to do language transfer
Alternative: The Universal Tagset

- 12 tags: NOUN, VERB, ADJ, ADV, PRON, DET, ADP, NUM, CONJ, PRT, ‘.’, and X.
- Deterministic conversion from tagsets in 22 languages.
- Better unsupervised parsing results
- Was used to transfer parsers

[Petrov et al. 2012]
Sources of Information

• What are the main sources of information for POS tagging?
  – Knowledge of neighboring words
    • Bill saw that man yesterday
    • NNP VB(D) DT NN NN
    • VB NN IN VB NN
  – Knowledge of word probabilities
    • *man* is rarely used as a verb….

• The latter proves the most useful, but the former also helps
Word-level Features

• Can do surprisingly well just looking at a word by itself:
  – Word the: the → DT
  – Lowercased words: importantly → RB
  – Prefixes unfathomable: un- → JJ
  – Suffixes Importantly: -ly → RB
  – Capitalization Meridian: CAP → NNP
  – Word shapes 35-year: d-x → JJ
Consider the problem of jointly modeling a pair of strings
- E.g.: part of speech tagging

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

DT  NN  IN  NN  VBD  NNS  VBD
The  average  of  interbank  offered  rates  plummeted  …

Q: How do we map each word in the input sentence onto the appropriate label?
A: We can learn a joint distribution:

\[ p(x_1 \ldots x_n, y_1 \ldots y_n) \]

And then compute the most likely assignment:

\[ \arg \max_{y_1 \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n) \]
Classic Solution: HMMs

We want a model of sequences $y$ and observations $x$

$$p(x_1 \ldots x_n, y_1 \ldots y_n) =$$

where $y_0=START$ and we call $q(y_i \mid y_{i-1})$ the transition distribution and $e(x_i \mid y_i)$ the emission (or observation) distribution.
Model Assumptions

- Tag/state sequence is generated by a Markov model
- Words are chosen independently, conditioned only on the tag/state
- These are totally broken assumptions for POS: why?

\[
p(x_1 \ldots x_n, y_1 \ldots y_n) = q(STOP|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1})e(x_i|y_i)
\]
HMM for POS Tagging

The Georgia branch had taken on loan commitments ...

- HMM Model:
  - States $Y = (DT, NNP, NN, VBD, VBN, RP, NN, NNS)$
  - Observations $X =$
  - Transition dist’n $q(y_i | y_{i-1})$ models
  - Emission dist’n $e(x_i | y_i)$ models
HMM for POS Tagging

The Georgia branch had taken on loan commitments ...

- **HMM Model:**
  - States $Y = \{\text{DT, NNP, NN, } \ldots \}$ are the POS tags
  - Observations $X = V$ are words
  - Transition dist’n $q(y_i|y_{i-1})$ models the tag sequences
  - Emission dist’n $e(x_i|y_i)$ models words given their POS
HMM Inference and Learning

• Learning
  – Maximum likelihood: transitions $q$ and emissions $e$
  \[ p(x_1 \ldots x_n, y_1 \ldots y_n) = q(STOP|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1})e(x_i|y_i) \]

• Inference
  – Viterbi
  \[ y^* = \arg \max_{y_1 \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n) \]

  – Forward backward
  \[ p(x_1 \ldots x_n, y_i) = \sum_{y_1 \ldots y_{i-1}} \sum_{y_{i+1} \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n) \]
Learning: Maximum Likelihood

\[ p(x_1 \ldots x_n, y_1 \ldots y_n) = q(\text{STOP}|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1})e(x_i|y_i) \]

- Maximum likelihood methods for estimating transitions \( q \) and emissions \( e \)

\[ q_{ML}(y_i|y_{i-1}) = \frac{c(y_{i-1}, y_i)}{c(y_{i-1})} \quad e_{ML}(x|y) = \frac{c(y, x)}{c(y)} \]

- Will these estimates be high quality?
  - Which is likely to be more sparse, \( q \) or \( e \)?

- Smoothing?
Learning: Low Frequency Words

\[ p(x_1 \ldots x_n, y_1 \ldots y_n) = q(\text{STOP}|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1})e(x_i|y_i) \]

- Typically, for transitions:
  - Linear Interpolation
    \[ q(y_i|y_{i-1}) = \lambda_1 q_{ML}(y_i|y_{i-1}) + \lambda_2 q_{ML}(y_i) \]

- However, other approaches used for emissions
  - **Step 1:** Split the vocabulary
    - Frequent words: appear more than \( M \) (often 5) times
    - Low frequency: everything else
  - **Step 2:** Map each low frequency word to one of a small, finite set of possibilities
    - For example, based on prefixes, suffixes, etc.
  - **Step 3:** Learn model for this new space of possible word sequences
Another Example: Chunking

• Goal: Segment text into spans with certain properties
• For example, named entities: PER, ORG, and LOC

Germany’s representative to the European Union’s veterinary committee Werner Zwingman said on Wednesday consumers should…

How is this a sequence tagging problem?
Named Entity Recognition

Germany’s representative to the European Union’s veterinary committee Werner Zwingman said on Wednesday consumers should…

[Germany]LOC’s representative to the [European Union]ORG’s veterinary committee [Werner Zwingman]PER said on Wednesday consumers should…

• HMM Model:
  – States $Y = \{NA, BL, CL, BO, CO, BP, CP\}$ represent beginnings (BL, BO, BP) and continuations (CL, CO, CP) of chunks, as well as other words (NA)
  – Observations $X = V$ are words
  – Transition dist’n $q(y_i|y_{i-1})$ models the tag sequences
  – Emission dist’n $e(x_i|y_i)$ models words given their type
Low Frequency Words: An Example

- Named Entity Recognition [Bickel et. al, 1999]

  - Used the following word classes for infrequent words:

<table>
<thead>
<tr>
<th>Word class</th>
<th>Example</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>twoDigitNum</td>
<td>90</td>
<td>Two digit year</td>
</tr>
<tr>
<td>fourDigitNum</td>
<td>1990</td>
<td>Four digit year</td>
</tr>
<tr>
<td>containsDigitAndAlpha</td>
<td>A8956-67</td>
<td>Product code</td>
</tr>
<tr>
<td>containsDigitAndDash</td>
<td>09-96</td>
<td>Date</td>
</tr>
<tr>
<td>containsDigitAndSlash</td>
<td>11/9/89</td>
<td>Date</td>
</tr>
<tr>
<td>containsDigitAndComma</td>
<td>23,000.00</td>
<td>Monetary amount</td>
</tr>
<tr>
<td>containsDigitAndPeriod</td>
<td>1.00</td>
<td>Monetary amount, percentage</td>
</tr>
<tr>
<td>othernum</td>
<td>456789</td>
<td>Other number</td>
</tr>
<tr>
<td>allCaps</td>
<td>BBN</td>
<td>Organization</td>
</tr>
<tr>
<td>capPeriod</td>
<td>M.</td>
<td>Person name initial</td>
</tr>
<tr>
<td>firstWord</td>
<td>first word of sentence</td>
<td>no useful capitalization information</td>
</tr>
<tr>
<td>initCap</td>
<td>Sally</td>
<td>Capitalized word</td>
</tr>
<tr>
<td>lowercase</td>
<td>can</td>
<td>Uncapitalized word</td>
</tr>
<tr>
<td>other</td>
<td>,</td>
<td>Punctuation marks, all other words</td>
</tr>
</tbody>
</table>
Low Frequency Words: An Example

Profits/NA soared/NA at/NA Boeing/SO Co./CO ,/NA easily/NA topping/NA forecasts/NA on/NA Wall/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP Mulally/CP announced/NA first/NA quarter/NA results/NA ./NA

- NA = No entity
- SO = Start Organization
- CO = Continue Organization
- SL = Start Location
- CL = Continue Location
- ...
Low Frequency Words: An Example

Profits/NA soared/NA at/NA Boeing/SO Co./CO ,/NA easily/NA topping/NA forecasts/NA on/NA Wall/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP Mulally/CP announced/NA first/NA quarter/NA results/NA ./NA

firstword/NA soared/NA at/NA initCap/SC Co./CC ,/NA easily/NA lowercase/NA forecasts/NA on/NA initCap/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP initCap/CP announced/NA first/NA quarter/NA results/NA ./NA

• NA = No entity
• SO = Start Organization
• CO = Continue Organization
• SL = Start Location
• CL = Continue Location
• …
HMM Inference and Learning

• Learning
  – Maximum likelihood: transitions q and emissions e

\[
p(x_1 \ldots x_n, y_1 \ldots y_n) = q(\text{STOP}|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1}) e(x_i|y_i)
\]

• Inference
  – Viterbi

\[
y^* = \arg \max_{y_1 \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n)
\]

  – Forward backward

\[
p(x_1 \ldots x_n, y_i) = \sum_{y_1 \ldots y_{i-1}} \sum_{y_{i+1} \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n)
\]
Inference (Decoding)

- **Problem:** find the most likely (Viterbi) sequence under the model

\[
y^* = \arg \max_{y_1 \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n)
\]

- Given model parameters, we can score any sequence pair

  NNP   VBZ   NN   NNS   CD   NN
  Fed   raises interest rates 0.5 percent .

  \[ q(\text{NNP}|\bullet) \cdot e(\text{Fed}|\text{NNP}) \cdot q(\text{VBZ}|\text{NNP}) \cdot e(\text{raises}|\text{VBZ}) \cdot q(\text{NN}|\text{VBZ}) \ldots. \]

- In principle, we’re done – list all possible tag sequences, score each one, pick the best one (the Viterbi state sequence)

  NNP VBZ NN NNS CD NN . \( \log p(x, y) = -23 \)
  NNP NNS NN NNS CD NN . \( \log(x, y) = -29 \)
  NNP VBZ VB NNS CD NN . \( \log p(x, y) = -27 \)

Any issue?
Finding the Best Trajectory

• Too many trajectories (state sequences) to list
• Option 1: Beam Search
  – A beam is a set of partial hypotheses
  – Start with just the single empty trajectory
  – At each derivation step:
    • Consider all continuations of previous hypotheses
    • Discard most, keep top $k$

• Beam search often works OK in practice, but …
  • … but sometimes you want the optimal answer
  • … and there’s usually a better option than naïve beams
The State Lattice / Trellis

START
^ Fed           raises       interest       rates
STOP

q(N|N) e(Fed|N) q(V|V) e(raises|V) e(interest|V) e(rates|J) e(STOP|V)
Scoring a Sequence

\[ y^* = \arg \max_{y_1 \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n) \]

\[ p(x_1 \ldots x_n, y_1 \ldots y_n) = q(STOP | y_n) \prod_{i=1}^{n} q(y_i | y_{i-1}) e(x_i | y_i) \]

- Define \( \pi(i, y_i) \) to be the max score of a sequence of length \( i \) ending in tag \( y_i \)

\[ \pi(i, y_i) = \max_{y_1 \ldots y_{i-1}} p(x_1 \ldots x_i, y_1 \ldots y_i) \]

\[ = \max_{y_{i-1}} e(x_i | y_i) q(y_i | y_{i-1}) \max_{y_1 \ldots y_{i-2}} p(x_1 \ldots x_{i-1}, y_1 \ldots y_{i-1}) \]

\[ = \max_{y_{i-1}} e(x_i | y_i) q(y_i | y_{i-1}) \pi(i - 1, y_{i-1}) \]

- We can now design an efficient algorithm.
  - How?
The Viterbi Algorithm

Dynamic program for computing (for all $i$)

$$
\pi(i, y_i) = \max_{y_1 \ldots y_{i-1}} p(x_1 \ldots x_i, y_1 \ldots y_i)
$$

Iterative computation:

$$
\pi(0, y_0) = \begin{cases} 
1 & \text{if } y_0 == START \\
0 & \text{otherwise}
\end{cases}
$$

For $i = 1 \ldots n$:

// Store score

$$
\pi(i, y_i) = \max_{y_{i-1}} e(x_i | y_i) q(y_i | y_{i-1}) \pi(i - 1, y_{i-1})
$$

// Store back-pointer

$$
bp(i, y_i) = \arg\max_{y_{i-1}} e(x_i | y_i) q(y_i | y_{i-1}) \pi(i - 1, y_{i-1})
$$
The State Lattice / Trellis

Tie breaking: Prefer first

<table>
<thead>
<tr>
<th>START</th>
<th>Fed raises interest STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>from \ to</td>
<td>^</td>
</tr>
<tr>
<td>^</td>
<td>0.0</td>
</tr>
<tr>
<td>N</td>
<td>0.0</td>
</tr>
<tr>
<td>V</td>
<td>0.0</td>
</tr>
<tr>
<td>$</td>
<td>0.0</td>
</tr>
</tbody>
</table>
The State Lattice / Trellis

Tie breaking: Prefer first

<table>
<thead>
<tr>
<th>from \ to</th>
<th>^</th>
<th>N</th>
<th>V</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>^</td>
<td>0.0</td>
<td>0.6</td>
<td>0.4</td>
<td>0.0</td>
</tr>
<tr>
<td>N</td>
<td>0.0</td>
<td>0.4</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>V</td>
<td>0.0</td>
<td>0.6</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td>$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>emissions</th>
<th>START</th>
<th>Fed</th>
<th>raises</th>
<th>interest</th>
<th>STOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>^</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>N</td>
<td>0.0</td>
<td>0.45</td>
<td>0.1</td>
<td>0.45</td>
<td>0.0</td>
</tr>
<tr>
<td>V</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>0.4</td>
<td>0.0</td>
</tr>
<tr>
<td>$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>
The Viterbi Algorithm: Runtime

- In term of sentence length $n$?
  - Linear
- In term of number of states $|K|$?
  - Polynomial

\[
\pi(i, y_i) = \max_{y_{i-1}} e(x_i | y_i) q(y_i | y_{i-1}) \pi(i - 1, y_{i-1})
\]

- Specifically:
  \[ O(n|K|) \] entries in $\pi(i, y_i)$
  \[ O(|K|) \] time to compute each $\pi(i, y_i)$
- Total runtime: \( O(n|K|^2) \)

Q: Is this a practical algorithm?
A: depends on $|K|$....
## Tagsets in Different Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Source</th>
<th># Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>PADT/CoNLL07 (Hajič et al., 2004)</td>
<td>21</td>
</tr>
<tr>
<td>Basque</td>
<td>Basque3LB/CoNLL07 (Aduriz et al., 2003)</td>
<td>64</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>BTB/CoNLL06 (Simov et al., 2002)</td>
<td>54</td>
</tr>
<tr>
<td>Catalan</td>
<td>CESS-ECE/CoNLL07 (Martí et al., 2007)</td>
<td>54</td>
</tr>
<tr>
<td>Chinese</td>
<td>Penn Chinese Treebank 6.0 (Palmer et al., 2007)</td>
<td>34</td>
</tr>
<tr>
<td>Chinese</td>
<td>Sinica/CoNLL07 (Chen et al., 2003)</td>
<td>294</td>
</tr>
<tr>
<td>Czech</td>
<td>PDT/CoNLL07 (Böhmová et al., 2003)</td>
<td>63</td>
</tr>
<tr>
<td>Danish</td>
<td>DDT/CoNLL06 (Kromann et al., 2003)</td>
<td>25</td>
</tr>
<tr>
<td>Dutch</td>
<td>Alpino/CoNLL06 (Van der Beek et al., 2002)</td>
<td>12</td>
</tr>
<tr>
<td>English</td>
<td>Penn Treebank (Marcus et al., 1993)</td>
<td>45</td>
</tr>
<tr>
<td>French</td>
<td>French Treebank (Abeillé et al., 2003)</td>
<td>30</td>
</tr>
<tr>
<td>German</td>
<td>Tiger/CoNLL06 (Brants et al., 2002)</td>
<td>54</td>
</tr>
<tr>
<td>German</td>
<td>Negra (Skut et al., 1997)</td>
<td>54</td>
</tr>
<tr>
<td>Greek</td>
<td>GDT/CoNLL07 (Prokopiidis et al., 2005)</td>
<td>38</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Szeged/CoNLL07 (Csendes et al., 2005)</td>
<td>43</td>
</tr>
<tr>
<td>Italian</td>
<td>ISST/CoNLL07 (Montemagni et al., 2003)</td>
<td>28</td>
</tr>
<tr>
<td>Japanese</td>
<td>Verbimobil/CoNLL06 (Kawata and Bartels, 2000)</td>
<td>80</td>
</tr>
<tr>
<td>Japanese</td>
<td>Kyoto4.0 (Kurohashi and Nagao, 1997)</td>
<td>42</td>
</tr>
<tr>
<td>Korean</td>
<td>Sejong (<a href="http://www.sejong.or.kr">http://www.sejong.or.kr</a>)</td>
<td>187</td>
</tr>
<tr>
<td>Portuguese</td>
<td>Floresta Sintá(ct)ica/CoNLL06 (Afonso et al., 2002)</td>
<td>22</td>
</tr>
<tr>
<td>Russian</td>
<td>SynTagRus-RNC (Boguslavsky et al., 2002)</td>
<td>11</td>
</tr>
<tr>
<td>Slovene</td>
<td>SDT/CoNLL06 (Džeroski et al., 2006)</td>
<td>20</td>
</tr>
<tr>
<td>Spanish</td>
<td>Ancora-Cast3LB/CoNLL06 (Civit and Martí, 2004)</td>
<td>47</td>
</tr>
<tr>
<td>Swedish</td>
<td>Talbanken05/CoNLL06 (Nivre et al., 2006)</td>
<td>41</td>
</tr>
<tr>
<td>Turkish</td>
<td>METU-Sabanci/CoNLL07 (Oflazer et al., 2003)</td>
<td>31</td>
</tr>
</tbody>
</table>

[Source: Petrov et al. 2012]
HMM Inference and Learning

• Learning
  – Maximum likelihood: transitions $q$ and emissions $e$

$$p(x_1 \ldots x_n, y_1 \ldots y_n) = q(STOP|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1}) e(x_i|y_i)$$

• Inference
  – Viterbi

$$y^* = \arg \max_{y_1 \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n)$$

  – Forward backward

$$p(x_1 \ldots x_n, y_i) = \sum_{y_1 \ldots y_{i-1}} \sum_{y_{i+1} \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n)$$
What about n-gram Taggers?

- States encode what is relevant about the past
- Transitions $P(s_i \mid s_{i-1})$ encode well-formed tag sequences
  - In a bigram tagger, states = tags
  
  In a trigram tagger, states = tag pairs
The State Lattice / Trellis

Not all edges are allowed

START

Fed

raises

interest

\(e(Fed|N)\)

\(e(\text{raises}|D)\)

\(e(\text{interest}|V)\)
### Tagsets in Different Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Source</th>
<th># Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>PADT/CoNLL07 (Hajič et al., 2004)</td>
<td>21</td>
</tr>
<tr>
<td>Basque</td>
<td>Basque3LB/CoNLL07 (Aduriz et al., 2003)</td>
<td>64</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>BTB/CoNLL06 (Simov et al., 2002)</td>
<td>54</td>
</tr>
<tr>
<td>Catalan</td>
<td>CESS-ECE/CoNLL07 (Martí et al., 2007)</td>
<td>54</td>
</tr>
<tr>
<td>Chinese</td>
<td>Penn Chinese Treebank 6.0 (Palmer et al., 2007)</td>
<td>24</td>
</tr>
<tr>
<td>Chinese</td>
<td>Sinica/CoNLL07 (Chen et al., 2003)</td>
<td>294</td>
</tr>
<tr>
<td>Czech</td>
<td>PDT/CoNLL07 (Böhmová et al., 2003)</td>
<td>63</td>
</tr>
<tr>
<td>Danish</td>
<td>DDT/CoNLL06 (Kromann et al., 2003)</td>
<td>25</td>
</tr>
<tr>
<td>Dutch</td>
<td>Alpino/CoNLL06 (Van der Beek et al., 2002)</td>
<td>12</td>
</tr>
<tr>
<td>English</td>
<td>Penn Treebank (Marcus et al., 1993)</td>
<td>45</td>
</tr>
<tr>
<td>French</td>
<td>French Treebank (Abeillé et al., 2003)</td>
<td>30</td>
</tr>
<tr>
<td>German</td>
<td>Tiger/CoNLL06 (Brants et al., 2002)</td>
<td>54</td>
</tr>
<tr>
<td>German</td>
<td>Negra (Skut et al., 1997)</td>
<td>54</td>
</tr>
<tr>
<td>Greek</td>
<td>GDT/CoNLL07 (Prokopiidis et al., 2005)</td>
<td>38</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Szeged/CoNLL07 (Csendes et al., 2005)</td>
<td>43</td>
</tr>
<tr>
<td>Italian</td>
<td>ISST/CoNLL07 (Montemagni et al., 2003)</td>
<td>28</td>
</tr>
<tr>
<td>Japanese</td>
<td>Verbmobil/CoNLL06 (Kawata and Bartels, 2000)</td>
<td>80</td>
</tr>
<tr>
<td>Japanese</td>
<td>Kyoto4.0 (Kurohashi and Nagao, 1997)</td>
<td>42</td>
</tr>
<tr>
<td>Korean</td>
<td>Sejong (<a href="http://www.sejong.or.kr">http://www.sejong.or.kr</a>)</td>
<td>187</td>
</tr>
<tr>
<td>Portuguese</td>
<td>Floresta Sintáctica/CoNLL06 (Afonso et al., 2002)</td>
<td>22</td>
</tr>
<tr>
<td>Russian</td>
<td>SynTagRus-RNC (Boguslavsky et al., 2002)</td>
<td>11</td>
</tr>
<tr>
<td>Slovene</td>
<td>SDT/CoNLL06 (Džeroski et al., 2006)</td>
<td>20</td>
</tr>
<tr>
<td>Spanish</td>
<td>Ancora-Cast3LB/CoNLL06 (Civit and Martí, 2004)</td>
<td>47</td>
</tr>
<tr>
<td>Swedish</td>
<td>Talbanken05/CoNLL06 (Nivre et al., 2006)</td>
<td>41</td>
</tr>
<tr>
<td>Turkish</td>
<td>METU-Sabanci/CoNLL07 (Oflazer et al., 2003)</td>
<td>31</td>
</tr>
</tbody>
</table>

\[294^2 = 86436\]
\[294^4 = 7471182096\]
\[45^2 = 2045\]
\[45^4 = 4100625\]

\[11^2 = 121\]
\[11^4 = 14641\]

[Petrov et al. 2012]
Some Numbers

• Rough accuracies:
  – Most freq tag:
  – Trigram HMM:
  – TnT (Brants, 2000):
    • A carefully smoothed trigram tagger
    • Suffix trees for emissions

  – Upper bound: ~98%

Most errors on unknown words