Natural Language Processing

Lecture 11: Sequence Models
Sequence to Sequence Modeling

• Part of Speech labelling
  • String of Words → String of POS Tags
• The choice of tag is dependent on both previous words and previous tags
• Two types of model
  • Probabilistic: noisy channel
  • Neural
Finding POS Tags

Bill directed plays about English kings
Running Example

Bill directed plays about English kings

PropN Verb Noun Adj Verb Verb PIN Prep Adv Part Adj Noun PIN Verb
### Running Example

**Bill directed** plays about **English kings**

<table>
<thead>
<tr>
<th>PropN</th>
<th>Adj</th>
<th>Verb</th>
<th>Prep</th>
<th>Adj</th>
<th>PIN</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>0.118</td>
<td>0.000</td>
<td>1546</td>
<td>0.750</td>
<td>0.250</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.006</td>
<td>0.000</td>
<td>502</td>
<td>0.750</td>
<td>0.250</td>
<td>0.000</td>
</tr>
<tr>
<td>303</td>
<td>0.870</td>
<td>0.000</td>
<td>120</td>
<td>0.750</td>
<td>0.250</td>
<td>0.000</td>
</tr>
</tbody>
</table>

**p(t |Bill)**

**p(t|directed)**

**p(t|plays)**

**p(t|about)**
Running Example: POS

Bill directed plays about English kings

<table>
<thead>
<tr>
<th>PropN</th>
<th>Adj</th>
<th>Verb</th>
<th>Prep</th>
<th>Adj</th>
<th>PIN</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.344</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.656</td>
<td></td>
<td>1.000</td>
<td></td>
<td></td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

p(t |English)     p(t |kings)

Adj  11    0.344
Noun 21    0.656
Hidden Markov Model

- $q_0$: start state ("silent")
- $q_f$: final state ("silent")
- $Q$: set of "normal" states (excludes $q_0$ and final $q_f$)
- $\Sigma$: vocabulary of observable symbols
- $\gamma_{i,j}$: probability of transitioning to $q_j$ given current state $q_i$
- $\eta_{i,w}$: probability of emitting $w \in \Sigma$ given current state $q_i$
HMM as a Noisy Channel

$p(y)$ using $\{\gamma_i,j\}$

$\rightarrow y \text{ (tags)} \rightarrow$  

$\rightarrow x \text{ (words)} \rightarrow$

$p(x \mid y)$ using $\{\eta_i,w\}$

source

channel

decode
States vs. Tags
Bill directed plays about English kings

p(PropN | <S> <S>) 0.202
p(Verb | <S> <S>) 0.023
p(Noun | <S> <S>) 0.040
Bill directed plays about English kings

<table>
<thead>
<tr>
<th>PropN</th>
<th>Adj</th>
<th>Verb</th>
<th>Prep</th>
<th>Adv</th>
<th>Adj</th>
<th>PIN</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>p(PropN</td>
<td>&lt;S&gt; &lt;S&gt;)</td>
<td>0.202</td>
<td>p(Adj</td>
<td>&lt;S&gt; PropN)</td>
<td>0.004</td>
<td>0.00081</td>
<td></td>
</tr>
<tr>
<td>p(Verb</td>
<td>&lt;S&gt; PropN)</td>
<td>0.139</td>
<td>0.02808</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(Verb</td>
<td>&lt;S&gt; &lt;S&gt;)</td>
<td>0.023</td>
<td>p(Adj</td>
<td>&lt;S&gt; Verb)</td>
<td>0.062</td>
<td>0.00143</td>
<td></td>
</tr>
<tr>
<td>p(Verb</td>
<td>&lt;S&gt; Verb)</td>
<td>0.032</td>
<td>0.00074</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(Noun</td>
<td>&lt;S&gt; &lt;S&gt;)</td>
<td>0.040</td>
<td>p(Adj</td>
<td>&lt;S&gt; Noun)</td>
<td>0.005</td>
<td>0.00020</td>
<td></td>
</tr>
<tr>
<td>p(Verb</td>
<td>&lt;S&gt; Noun)</td>
<td>0.222</td>
<td>0.00888</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Running Example

Bill directed plays about English kings.

<table>
<thead>
<tr>
<th>PropN</th>
<th>Adj</th>
<th>Verb</th>
<th>PIN</th>
<th>Prep</th>
<th>Adv</th>
<th>Part</th>
<th>Adj</th>
<th>PIN</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(Adj</td>
<td>&lt;S&gt; PropN)</td>
<td>0.00081</td>
<td>p(Verb</td>
<td>PropN Adj)</td>
<td>0.011</td>
<td>0.00001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(Verb</td>
<td>&lt;S&gt; PropN)</td>
<td>0.02808</td>
<td>p(Verb</td>
<td>PropN Verb)</td>
<td>0.162</td>
<td><strong>0.00455</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(Adj</td>
<td>&lt;S&gt; Verb)</td>
<td>0.00143</td>
<td>p(Verb</td>
<td>Verb Adj)</td>
<td>0.009</td>
<td>0.00001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(Verb</td>
<td>&lt;S&gt; Verb)</td>
<td>0.00074</td>
<td>p(Verb</td>
<td>Verb Verb)</td>
<td>0.078</td>
<td>0.00006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(Adj</td>
<td>&lt;S&gt; Noun)</td>
<td>0.00020</td>
<td>p(Verb</td>
<td>Noun Adj)</td>
<td>0.020</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(Verb</td>
<td>&lt;S&gt; Noun)</td>
<td>0.00888</td>
<td>p(Verb</td>
<td>Noun Verb)</td>
<td>0.176</td>
<td>0.00156</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Running Example (posterior)

Bill directed plays about English kings

<table>
<thead>
<tr>
<th>PropN</th>
<th>Adj</th>
<th>Verb</th>
<th>PIN</th>
<th>Prep</th>
<th>Adv</th>
<th>Part</th>
<th>Adj</th>
<th>Noun</th>
<th>PIN</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p(t</td>
<td>Bill)</td>
<td>p(Bill</td>
<td>t)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>0.118</td>
<td>0.00044</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.006</td>
<td>0.00002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>303</td>
<td>0.870</td>
<td>0.00228</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Running Example

Bill directed plays about English kings

<table>
<thead>
<tr>
<th>PropN</th>
<th>Adj</th>
<th>Verb</th>
<th>Prep</th>
<th>Adv</th>
<th>Part</th>
<th>Adj</th>
<th>PIN</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
p(t | \text{directed}) \quad p(\text{directed} | t)
\]

<table>
<thead>
<tr>
<th>Adj</th>
<th>0</th>
<th>0.000</th>
<th>0.00000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb</td>
<td>10</td>
<td>1.000</td>
<td>0.00008</td>
</tr>
</tbody>
</table>
Running Example

Bill directed plays about English kings

<table>
<thead>
<tr>
<th>PropN</th>
<th>Adj</th>
<th>Verb</th>
<th>Prep</th>
<th>Adj</th>
<th>PIN</th>
<th>Verb</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ p(t | \text{plays}) \quad p(\text{plays} | t) \]

| Verb | PIN | p(t | \text{plays}) | p(\text{plays} | t) |
|------|-----|-----------------|-------|
| 18   | 6   | 0.750           | 0.00014 |
| 0.250| 0.00010 |
Combining Two Components

• Prior $p(Y)$ the “language model”
  • What is the likelihood of a tag sequence
• Posterior $p(x|y)$ the “observation”
  • What is likelihood of word given tag
• We want to find the max for both
  • Bayes Rule $p(Y|X) = \frac{p(Y) \ p(X|Y)}{p(X)}$
HMM as a Noisy Channel

$p(y)$ using $\{\gamma_{i,j}\}$

$p(x \mid y)$ using $\{\eta_{i,w}\}$

Source $\rightarrow y$ (tags) $\rightarrow x$ (words)

channel

decode
Part-of-Speech Tagging Task

- **Input:** a sequence of word tokens $x$
- **Output:** a sequence of part-of-speech tags $y$, one per word

HMM solution: find the most likely tag sequence, given the word sequence.
If I knew the best state sequence for words \( x_1 \ldots x_{n-1} \), then I could figure out the last state.

That decision would depend only on state \( n-1 \).

\[
y_n^* = \arg \max_{q_i \in Q} p(Y_1 = y_1^*, \ldots, Y_{n-1} = y_{n-1}^*, Y_n = q_i \mid \mathbf{x})
\]

\[
= \arg \max_{q_i \in Q} V[n-1, y_{n-1}^*] \cdot \gamma_{y_{n-1}^*, i} \cdot \eta_i, x_n \cdot \gamma_i, f
\]

\[
= \arg \max_{q_i \in Q} \gamma_{y_{n-1}^*, i} \cdot \eta_i, x_n \cdot \gamma_i, f
\]

I don’t know that best sequence, but there are only \( |Q| \) options at \( n-1 \).

So I only need the score of the best sequence up to \( n-1 \), ending in each possible state at \( n-1 \). Call this \( V[n-1, q] \) for \( q \in Q \).

Ditto, at every other timestep \( n-2, n-3, \ldots 1 \).
Viterbi Algorithm
(Recursive Equations)

\[ V[0, q_0] = 1 \]
\[ V[t, q_j] = \max_{q_i \in Q \cup \{q_0\}} V[t - 1, q_i] \cdot \gamma_{i,j} \cdot \eta_{j,x_t} \]

\[ \text{goal} = \max_{q_i \in Q} V[n, q_i] \cdot \gamma_{i,f} \]
Viterbi Algorithm (Procedure)

\[
\begin{align*}
V[*, *] & \leftarrow 0 \\
V[0, q0] & \leftarrow 1 \\
\text{for } t = 1 \ldots n \\
\quad \text{foreach } qj \\
\quad \quad \text{foreach } qi \\
\quad \quad \quad V[t, qj] & \leftarrow \max\{V[t, qj] , V[t - 1, qi] \times \gamma_{i,j} \times \eta_{i,xt}\} \\
\text{foreach } qi \\
\quad \quad \text{goal } & \leftarrow \max\{ \text{goal, } V[n, qi] \times \gamma_{i,f}\} \\
\text{return } \text{goal}
\end{align*}
\]
Bill directed plays about English kings

<table>
<thead>
<tr>
<th>( q0 )</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q1 )</td>
<td></td>
</tr>
<tr>
<td>( q2 )</td>
<td></td>
</tr>
<tr>
<td>( q3 )</td>
<td></td>
</tr>
<tr>
<td>( q4 )</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>( q</td>
<td>)</td>
</tr>
<tr>
<td>( Q</td>
<td>)</td>
</tr>
<tr>
<td>( qf )</td>
<td></td>
</tr>
</tbody>
</table>
Neural S2S Example

• What is the PoS distribution of OOVs
  • Assume overall distribution from corpora
  • (Though less likely to be a Det, Conj, than Noun)

• Looking at the letters
  • Starts with a capital letter
  • Contains a number
  • Ends in “ed” or “ing”
Example from 11-737 (Neubig)

http://demo.clab.cs.cmu.edu/11737fa20/assignments/assign1.pdf

• Assumes PyTorch installation (ana)conda
• Doesn’t require GPUs, but can use them
• Builds POS tagger models for 8 languages
  • af ar cs en es hy lt ta
Data Examples

• Data split into train, dev and test sets
• 7K-400K examples
• Two column
  AI PROPN
  - PUNCT
  Zaman PROPN
  : PUNCT
  American ADJ
  forces NOUN
neural network v4.0: recurrent neural networks

Big idea: use hidden layers to represent sequential state

Feed-forward neural networks

Recurrent neural networks

How did we represent $x$ for document classification?

Real .... Madrid = Real Madrid

Do we share parameters across states?

Figure credits: Andrej Karpathy
neural network v4.1: output sequences

Figure credits: Andrej Karpathy
neural network v4.3: bidirectional RNNs

Unidirectional RNNs

Bidirectional RNNs

Figure credits: Christopher Olah
Model BiLSTM

{
    "embedding_dim": 100,
    "hidden_dim": 128,
    "n_layers": 2,
    "bidirectional": true,
    "dropout": 0.25,
    "batch_size": 128
}
## Model Accuracy

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
<th>Test Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Afrikaans</td>
<td>af</td>
<td>89.81%</td>
</tr>
<tr>
<td>Arabic</td>
<td>ar</td>
<td>94.48%</td>
</tr>
<tr>
<td>Czech</td>
<td>cs</td>
<td>94.36%</td>
</tr>
<tr>
<td>English</td>
<td>en</td>
<td>91.74%</td>
</tr>
<tr>
<td>Spanish</td>
<td>es</td>
<td>94.03%</td>
</tr>
<tr>
<td>Armenian</td>
<td>hy</td>
<td>81.26%</td>
</tr>
<tr>
<td>Lithuanian</td>
<td>lt</td>
<td>76.76%</td>
</tr>
<tr>
<td>Tamil</td>
<td>ta</td>
<td>40.95%</td>
</tr>
</tbody>
</table>
Unknown words

• What is the PoS distribution of OOVs
  • Assume overall distribution from corpora
  • (Though less likely to be a Det, Conj, than Noun)
• Looking at the letters
  • Starts with a capital letter
  • Contains a number
  • Ends in “ed” or “ing”
Part of Speech in other Languages

• Need labeled data
  • Can be approximate, then correct it
• Morphologically rich languages
  • Need to decompose tokens to morphemes
  • Partly easier (but still PoS ambiguities)
Unsupervised PoS Tagging

• Words in the same context are the same Tag
  • Find all contexts: w1 X w2
  • Find most frequent Xs make them a tag
  • Repeat until you want to stop
• For English: do this 20 times
  • BE/HAVE  MR/MRS  AND/BUT/AT/AS
  • TO/FOR/OF/IN VERY/SO SHE/HE/IT/I/YOU
  • But no Nouns/Verb/Adj distinctions
Brown Clustering

- Unsupervised Word Clustering
- Non-syntax derived clusters
- “Semantically” related classes
- For example in a database of Flight information
  - To Shanghai, To Beijing, To London
  - To CLASS13, To CLASS13, To CLASS13
- Brown Clustering:
  - hierarchical agglomerative cluster.
  - Gives a binary tree, so it can easily scaled
Part of Speech and Tagging

• Reduced set of linguistic tags
  • Closed Class: Determiners, Pronouns ...
  • Open Class: Nouns, Verbs, Adjs, Adverbs

• Probabilistic Labeling
  • Bayes/Noisy Channel
  • $P(\text{tag} | \text{word}) \times P(\text{tag})$

• HMMs, Viterbi decoding
• Sequence Models: BiLSTM tagging
• Unsupervised tagging/clustering

• Use what is *best* for your task
  • (and use what is available)