Unsupervised Neural Hidden Markov Models

Ke Tran¹, Yonatan Bisk, Ashish Vaswani², Daniel Marcu and Kevin Knight

USC Information Sciences Institute
¹Univ of Amsterdam, ²Google Brain

https://github.com/ketranm/neuralHMM
Bayesian Models

• HMMs, CFGs, … have been standard workhorses of the NLP community

• Generative models lend themselves to unsupervised estimation

• Bayesian models have elegant, but often very parametrically expensive smoothing approaches
Why Neuralize Bayesian Models?

- Unsupervised structure learning
- Simple modular extensions
- Embeddings and vector representations have been shown to generalize well.
This is a nice direction

Relevant EMNLP 2016 Papers:

Online Segment to Segment Neural Transduction. Lei Yu, Jan Buys, and Phil Blunsom.

Unsupervised Neural Dependency Parsing. Yong Jiang, Wenjuan Han, and Kewei Tu.
Hidden Markov Models

Given an observed sequence of text: \( \mathbf{x} \)

Probability of a given token: \( p(x_t | z_t) \times P(z_t | z_{t-1}) \)

\[
p(x, z) = \prod_{t=1}^{n+1} p(z_t | z_{t-1}) \prod_{t=1}^{n} p(x_t | z_t)
\]
Supervised POS Tagging

The orange man will lose the election

<table>
<thead>
<tr>
<th>The</th>
<th>orange</th>
<th>man</th>
<th>will</th>
<th>lose</th>
<th>the</th>
<th>election</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>JJ</td>
<td>NN</td>
<td>MD</td>
<td>VB</td>
<td>DT</td>
<td>NN</td>
</tr>
</tbody>
</table>

Goal: Predict the correct class for each word in the sentence
Solution: Count and divide

\[ p(\text{orange}|\text{JJ}) = \frac{|\text{orange, JJ}|}{|\text{JJ}|} \quad p(\text{JJ}|\text{DT}) = \frac{|\text{DT, JJ}|}{|\text{DT}|} \]

Parameters: \[ V \times K \quad K \times K \]
Simple Supervised Neural HMM

The orange man will lose the election

Replace parameter matrices with NNs + Softmax
Train with Cross Entropy

Emission Network

Transition Network
Unsupervised Neural HMM

The orange man will lose the election

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

Emission Network

Transition Network
Bayesian POS Tag Induction

Goal: Discover the set of classes which best model the observed data.

Solution: Baum-Welch
Posteriors

Probability of a specific cluster assignment

\[ p(z_t = i | x) \]

Probability of a specific cluster transition

\[ p(z_t = i, z_{t+1} = j | x) \]

Bayesian update: Count and Divide
Count and Divide

Initialize

\[ p(w_i|C_j) \]

\[
\begin{array}{c}
0.3 \\
0.1 \\
0.2 \\
0.4 \\
\end{array}
\]

Compute Posterials

\[
\sum_{\text{corpus}} p(w_i, C_j) \]

\[
\begin{array}{c}
50 \\
2 \\
4 \\
35 \\
\end{array}
\]

Normalize

\[
\hat{p}(w_i|C_j) \]

\[
\begin{array}{c}
0.55 \\
0.02 \\
0.04 \\
0.38 \\
\end{array}
\]
Unsupervised Neural HMM

The orange man will lose the election?

\begin{align*}
\Pr(z_t = i \mid x) \\
\Pr(z_t = i, z_{t+1} = j \mid x)
\end{align*}

Emission Network

Transition Network
Generalized EM

$$\ln p(x|\theta) =$$

$$\mathbb{E}_{q(z)}[\ln p(x, z|\theta)] + H[q(z)] + KL[q(z)||p(z|x, \theta)]$$

E-Step Compute Surrogate q

M-Step Maximize Expectation
What is the gradient?

Set \( q(z) = p(z|x, \theta) \)

\[
\mathbb{E}_{q(z)}[\ln p(x, z|\theta)] + H[q(z)] + \text{KL}[q(z)||p(z|x, \theta)]
\]

Take Derivative w.r.t. \( \theta \)

\[
\mathbb{E}_{q(z)}[\ln p(x, z|\theta)] + H[q(z)]
\]

\[
J(\theta) = \sum_z p(z|x) \frac{\partial \ln p(x, z|\theta)}{\partial \theta}
\]
Initial Evaluation
Induction Metrics

• 1-1: Bijection between induced and gold classes
• M-1: Map induced class to its closest gold class
• V-M: Harmonic mean of $H(c,g)$ and $H(g,c)$

Higher numbers are better
## Evaluation

<table>
<thead>
<tr>
<th></th>
<th>1-1</th>
<th>M-1</th>
<th>V-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>41.4</td>
<td>62.5</td>
<td>53.3</td>
</tr>
<tr>
<td>Neural HMM</td>
<td>45.7</td>
<td>59.8</td>
<td>54.2</td>
</tr>
</tbody>
</table>

The neural model has access to no additional information.
Morphology

CNN based embeddings provide morphological information

- kernels = \{1, 2, 3, 4, 5, 6, 7\}
- feature maps = \{50, 100, 128, 128, 128, 128, 128\}
## Evaluation

<table>
<thead>
<tr>
<th></th>
<th>1-1</th>
<th>M-1</th>
<th>V-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>41.4</td>
<td>62.5</td>
<td>53.3</td>
</tr>
<tr>
<td>Neural HMM</td>
<td>45.7</td>
<td>59.8</td>
<td>54.2</td>
</tr>
<tr>
<td>+ Conv</td>
<td><strong>48.3</strong></td>
<td><strong>74.1</strong></td>
<td><strong>66.1</strong></td>
</tr>
</tbody>
</table>
Extended Context

Traditional:

Bi-gram transition \( p(z_t|z_{t-1}) \) \( K^2 \)

Tri-gram transition \( p(z_t|z_{t-1}, z_{t-2}) \) \( K^3 \)

N-gram transition \( p(z_t|z_{t-1}, z_{t-2}, \ldots, z_{t-n}) \) \( K^{n+1} \)

Alternative:

Previous tag and word \( p(z_t|z_{t-1}, x_{t-1}) \) \( V \times K^2 \)

Previous tag and sentence \( p(z_t|z_{t-1}, x_{t-1}, \ldots, x_0) \) \( V^t \times K^2 \)
LSTM Context

LSTM consumes the sentence and produces a transition matrix

\[ p(z_t | z_{t-1}, x_{t-1}, \ldots, x_0) \]

\[ T_{t-1,t} \]
## Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>1-1</th>
<th>M-1</th>
<th>V-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>41.4</td>
<td>62.5</td>
<td>53.3</td>
</tr>
<tr>
<td>Neural HMM</td>
<td>45.7</td>
<td>59.8</td>
<td>54.2</td>
</tr>
<tr>
<td>+ Conv</td>
<td>48.3</td>
<td>74.1</td>
<td>66.1</td>
</tr>
<tr>
<td>+ LSTM</td>
<td>52.4</td>
<td>65.1</td>
<td>60.4</td>
</tr>
<tr>
<td>+ Conv &amp; LSTM</td>
<td>60.7</td>
<td>79.1</td>
<td>71.7</td>
</tr>
<tr>
<td>Blunsom 2011</td>
<td>77.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yatbaz 2012</td>
<td>80.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Types / Cluster

- Gold
- LSTM
- FF Conv
- Conv+LSTM
## Clusterings

### Largest Cluster

<table>
<thead>
<tr>
<th>LSTM</th>
<th>Conv</th>
</tr>
</thead>
<tbody>
<tr>
<td>of</td>
<td>years</td>
</tr>
<tr>
<td>in</td>
<td>trading</td>
</tr>
<tr>
<td>to</td>
<td>sales</td>
</tr>
<tr>
<td>for</td>
<td>president</td>
</tr>
<tr>
<td>on</td>
<td>companies</td>
</tr>
<tr>
<td>from</td>
<td>prices</td>
</tr>
</tbody>
</table>

### Numbers

<table>
<thead>
<tr>
<th>LSTM</th>
<th>Conv</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>million</td>
</tr>
<tr>
<td>million</td>
<td></td>
</tr>
<tr>
<td>year</td>
<td>share</td>
</tr>
<tr>
<td>cents</td>
<td>points</td>
</tr>
<tr>
<td>1/2</td>
<td>point</td>
</tr>
<tr>
<td>trillion</td>
<td>point</td>
</tr>
</tbody>
</table>
What's a good clustering?

<table>
<thead>
<tr>
<th>NNP</th>
<th>$C_{15}$</th>
<th>$C_{25}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>Corp.</td>
<td></td>
</tr>
<tr>
<td>British</td>
<td>Inc.</td>
<td></td>
</tr>
<tr>
<td>National</td>
<td>Co.</td>
<td></td>
</tr>
<tr>
<td>Congress</td>
<td>Board</td>
<td></td>
</tr>
<tr>
<td>Japan</td>
<td>Group</td>
<td></td>
</tr>
<tr>
<td>San</td>
<td>Bank</td>
<td></td>
</tr>
<tr>
<td>Federal</td>
<td>Inc</td>
<td></td>
</tr>
<tr>
<td>West</td>
<td>Bush</td>
<td></td>
</tr>
<tr>
<td>Dow</td>
<td>Department</td>
<td></td>
</tr>
</tbody>
</table>
Future Work

• Harnessing Extra Data
• Modifying the objective function
• Multilingual experiments
• Using this approach with other generative models
Thanks!

https://github.com/ketranm/neuralHMM

Parameter Initialization, Tricks, Ablation in paper and in Github README