7 Sequential Models

1. Sequential Inference Problems
2. HMM
3. Advanced Models

Dan Roth
University of Illinois, Urbana-Champaign
danr@cs.uiuc.edu
http://L2R.cs.uiuc.edu/~danr

Tutorial on Machine Learning in Natural Language Processing and Information Extraction
Outline

- Shallow Parsing
  - What it is
  - Why we need it
- Shallow Parsing (Learning) Models
  - Learning Sequences
- Hidden Markov Model (HMM)
- Discriminative Approaches
  - HMM with Classifiers
  - PMM
- Learning and Inference
  - CSCL
- Related Approaches
Parsing

- Parsing is a task to analyze for syntactical structure of inputs
- An output is a “parse tree”
Mr. Brown blamed Mr. Bob for making things dirty.
Parsing

- Parsing is an important task toward understanding natural languages
- It is not an easy task
- Many NLP tasks do not require all information from parse trees
Shallow Parsing

- Shallow Parsing is a generic term for a task that identifies only a subset of parse trees.
Shallow Parsing

Brown blamed Mr. Bob making things dirty Mr. for

Diagram representation of the sentence: S → VP → NP → Mr. Brown blamed Mr. Bob making things dirty Mr.
Reasons for Studying Shallow Parsing

- Not all the parse tree information is needed
  - It should do better than full parsing in those specific parts
- It can be done more robustly
  - More adaptive to data from new domains (Li & Roth’ CoNLL’01)
- Shallow parsing as an intermediate step toward full parsing
- It is a generic chunking task – applicable to many other segmentation tasks such as named entity recognition. [LBJ generic segmentation implementation]
By shallow parsing we mean: identifying non-overlapping, non-embedding phrases.

Shallow Parsing = Text Chunking
Text Chunking

- Usually text chunking without any further specification follows the definition from CoNLL-2000 shared task [Demo]
Brown blamed Mr. Bob making things dirty Mr. for
Modeling Shallow Parsing

- Model shallow parsing as a sequence prediction problem
- For each word predict one of these:
  - B – Beginning of a chunk
  - I – Inside a chunk but not a beginning
  - O – Outside a chunk
Mr. Brown blamed Mr. Bob for making things dirty

NP
B I O
NP
B I O O
NP
B O
Similar Problems

☐ This model applies to many other problems

- NLP
  - Named entity recognition
  - Verb-argument identification

- Other domains
  - Identifying genes (splice sites, Chuang&Roth’01)
Outline

- Shallow Parsing
  - What it is
  - Why we need it
- Shallow Parsing (Learning) Models
  - Learning Sequences
- Hidden Markov Model (HMM)
- Discriminative Approaches
  - HMM with Classifiers
  - PMM
- Learning and Inference
  - CSCL
- Related Approaches
Hidden Markov Model (HMM)

- HMM is a probabilistic generative model
  - It models how an observed sequence is generated
- Let’s call each position in a sequence a time step
- At each time step, there are two variables
  - Current state (hidden)
  - Observation
HMM

- Initial state probability $P(s_1)$
- Transition probability $P(s_t|s_{t-1})$
- Observation probability $P(o_t|s_t)$
HMM for Shallow Parsing

- **States:**
  - \{B, I, O\}

- **Observations:**
  - Actual words and/or part-of-speech tags

```
s_1=B   s_2=I   s_3=O   s_4=B   s_5=I   s_6=O

O_1 Mr.  O_2 Brown  O_3 blamed  O_4 Mr.  O_5 Bob  O_6 for
```

....

18
HMM for Shallow Parsing

Given a sentence, we can ask what the most likely state sequence is

Initial state probability:
P(s_1=B), P(s_1=I), P(s_1=O)

Transition probability:
P(s_t=B|s_{t-1}=B), P(s_t=I|s_{t-1}=B), P(s_t=O|s_{t-1}=B),
P(s_t=B|s_{t-1}=I), P(s_t=I|s_{t-1}=I), P(s_t=O|s_{t-1}=I), ...

Observation Probability:
P(o_t=Mr.|s_t=B), P(o_t=Brown|s_t=B), ...
P(o_t=Mr.|s_t=I), P(o_t=Bob|s_t=I), ...
P(o_t=O|s_t=O), ...

Given a sentence, we can ask what the most likely state sequence is
Finding most likely state sequence in HMM (1)

\[ P(s_k, s_{k-1}, \ldots, s_1, o_k, o_{k-1}, \ldots, o_1) \]

\[ = P(o_k|o_{k-1}, o_{k-2}, \ldots, o_1, s_k, s_{k-1}, \ldots, s_1) \]

\[ \cdot P(o_{k-1}, o_{k-2}, \ldots, o_1, s_k, s_{k-1}, \ldots, s_1) \]

\[ = P(o_k|s_k) \cdot P(o_{k-1}, o_{k-2}, \ldots, o_1, s_k, s_{k-1}, \ldots, s_1) \]

\[ = P(o_k|s_k) \cdot P(s_k|s_{k-1}, s_{k-2}, \ldots, s_1, o_{k-1}, o_{k-2}, \ldots, o_1) \]

\[ \cdot P(s_{k-1}, s_{k-2}, \ldots, s_1, o_{k-1}, o_{k-2}, \ldots, o_1) \]

\[ = P(o_k|s_k) \cdot P(s_k|s_{k-1}) \]

\[ \cdot P(s_{k-1}, s_{k-2}, \ldots, s_1, o_{k-1}, o_{k-2}, \ldots, o_1) \]

\[ = P(o_k|s_k) \cdot \prod_{t=1}^{k-1} P(s_{t+1}|s_t) \cdot P(o_t|s_t) \]

\[ \cdot P(s_1) \]
Finding most likely state sequence in HMM (2)

\[
\arg \max_{s_k, s_{k-1}, \ldots, s_1} P(s_k, s_{k-1}, \ldots, s_1 | o_k, o_{k-1}, \ldots, o_1)
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} \frac{P(s_k, s_{k-1}, \ldots, s_1, o_k, o_{k-1}, \ldots, o_1)}{P(o_k, o_{k-1}, \ldots, o_1)}
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} P(s_k, s_{k-1}, \ldots, s_1, o_k, o_{k-1}, \ldots, o_1)
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} P(o_k | s_k) \cdot \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \cdot P(s_1)
\]
Finding most likely state sequence in HMM (3)

\[
\max_{s_k, s_{k-1}, \ldots, s_1} P(o_k | s_k) \cdot \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \cdot P(s_1)
\]

\[
= \max_{s_k} P(o_k | s_k) \cdot \max_{s_{k-1}, \ldots, s_1} \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \cdot P(s_1)
\]

\[
= \max_{s_k} P(o_k | s_k) \cdot \max_{s_{k-1}} \left[ \prod_{t=1}^{k-2} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1)
\]

\[
= \max_{s_k} P(o_k | s_k) \cdot \max_{s_{k-1}} \left[ \prod_{t=1}^{k-2} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot \max_{s_{k-2}} \left[ P(s_{k-1} | s_{k-2}) \cdot P(o_{k-2} | s_{k-2}) \right] \cdot \ldots \cdot \max_{s_1} \left[ P(s_2 | s_1) \cdot P(o_1 | s_1) \right] \cdot P(s_1)
\]
Finding most likely state sequence in HMM (4)

\[
\max_{s_k} P(o_k|s_k) \cdot \max_{s_{k-1}} [P(s_k|s_{k-1}) \cdot P(o_{k-1}|s_{k-1})] \\
\cdot \max_{s_{k-2}} [P(s_{k-1}|s_{k-2}) \cdot P(o_{k-2}|s_{k-2})] \cdot \ldots \\
\cdot \max_{s_2} [P(s_3|s_2) \cdot P(o_2|s_2)] \cdot \\
\cdot \max_{s_1} [P(s_2|s_1) \cdot P(o_1|s_1)] \cdot P(s_1)
\]

- Viterbi’s Algorithm
- Dynamic Programming
Learning the Model

- Estimate
  - Initial state probability $P(s_1)$
  - Transition probability $P(s_t|s_{t-1})$
  - Observation probability $P(o_t|s_t)$

- Unsupervised Learning
  - EM Algorithm

- Supervised Learning
  - Estimate each element directly from data
Experimental Results

- NP Prediction (POS Tags Only)
  - 89.08 Recall
  - 86.62 Precision
  - 87.83 F1

\[
R(\text{Recall}) = \frac{\text{Number of chunks predicted correctly}}{\text{Number of correct chunks}}
\]
\[
P(\text{Precision}) = \frac{\text{Number of chunks predicted correctly}}{\text{Number of predicted chunks}}
\]
\[
F_1 = \frac{2RP}{R+P}
\]
Note

− Experiments use a different representation

Mr. Brown blamed Mr. Bob for making things dirty

[ Mr. Brown] blamed [ Mr. Bob] for [ making] [ things ] dirty
Outline

- Shallow Parsing
  - What it is
  - Why we need it
- Shallow Parsing (Learning) Models
  - Learning Sequences
- Hidden Markov Model (HMM)
- Discriminative Approaches
  - HMM with Classifiers
  - PMM
- Learning and Inference
  - CSCL
- Related Approaches
Problems with HMM

- Long-term dependencies are hard to incorporate
- HMM is trained to maximize the likelihood of the data, not to maximize the predictions
  - Maximize \( P(S,O) \), not \( P(S|O) \)
  - What metric do we care about?
Discriminative Approaches

- Model the predictions directly
- Shallow Parsing
  - Goal: predict BIO sequences given a sentence
    - \( S^* = \text{argmax} \ P(S|O) \)
  - Generative approaches model \( P(S,O) \)
  - Discriminative approaches model \( P(S|O) \) directly
Discriminative Approaches

- Use classifiers to predict the labels based on the context of the inputs

Mr. Brown blamed Mr. Bob for making things dirty

- Good: larger context can be incorporated
Problems with using Classifiers

- Outputs may be inconsistent

Mr. Brown blamed Mr. Bob for making things dirty

A classifier

I I O O B O
Constraints

- An 'I' cannot follow an 'O'
- The sequence has to begin with a 'B' or an 'O'

How to maintain constraints?
HMM revisited

- HMM does not have this problem.
- These constraints are taken care for by the transition probabilities.
- Specifically, zero transition probabilities ensure that outputs always satisfy constraints.
HMM with Classifiers

- HMM requires observation probability
  - $P(o_t|s_t)$

- Classifiers give us
  - $P(s_t|o_t)$

- We can compute $P(o_t|s_t)$ by
  - $P(o_t|s_t) = P(s_t|o_t)P(o_t)/P(s_t)$
  - See details here
# Experimental Results

- **NP Prediction**

<table>
<thead>
<tr>
<th></th>
<th>Recall</th>
<th>Prediction</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM</td>
<td>89.08</td>
<td>86.62</td>
<td>87.83</td>
</tr>
<tr>
<td>HMM with Classifiers</td>
<td>91.49</td>
<td>91.53</td>
<td>91.51</td>
</tr>
<tr>
<td>HMM with Classifiers (with Lexical features)</td>
<td>93.85</td>
<td>93.46</td>
<td>93.65</td>
</tr>
</tbody>
</table>
Projection based Markov Model (PMM)

- Classifiers use previous prediction as part of inputs

A classifier

Mr. Brown blamed Mr. Bob for making things dirty

B I O B I O O B O
PMM

- Each classifier outputs $P(s_t|s_{t-1}, O)$

$$\arg\max_{s_k, s_{k-1}, \ldots, s_1} P(s_k, s_{k-1}, \ldots, s_1 | O)$$

$$= \arg\max_{s_k, s_{k-1}, \ldots, s_1} P(s_k | s_{k-1}, \ldots, s_1, O) \cdot P(s_{k-1}, \ldots, s_1, O)$$

$$= \arg\max_{s_k, s_{k-1}, \ldots, s_1} P(s_k | s_{k-1}, O) \cdot P(s_{k-1}, \ldots, s_1, O)$$

$$= \arg\max_{s_k, s_{k-1}, \ldots, s_1} \left[ \prod_{t=2}^{k} P(s_t | s_{t-1}, O) \right] \cdot P(s_1 | O)$$

- How do you Train/Test in this case?
## Experimental Results

### NP Prediction

<table>
<thead>
<tr>
<th>Model Description</th>
<th>Recall</th>
<th>Prediction</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM (POS)</td>
<td>89.08</td>
<td>86.62</td>
<td>87.83</td>
</tr>
<tr>
<td>HMM with Classifiers (POS)</td>
<td>91.49</td>
<td>91.53</td>
<td>91.51</td>
</tr>
<tr>
<td>HMM with Classifiers</td>
<td>93.85</td>
<td>93.46</td>
<td>93.65</td>
</tr>
<tr>
<td>PMM (POS)</td>
<td>92.04</td>
<td>91.77</td>
<td>91.90</td>
</tr>
<tr>
<td>PMM</td>
<td>94.28</td>
<td>94.02</td>
<td>94.15</td>
</tr>
</tbody>
</table>
Outline

- Shallow Parsing
  - What it is
  - Why we need it
- Shallow Parsing (Learning) Models
  - Learning Sequences
- Hidden Markov Model (HMM)
- Discriminative Approaches
  - HMM with Classifiers
  - PMM
- Learning and Inference
  - CSCL
- Related Approaches
Learning and Inference

- Learning: estimate parameters from data, and makes predictions for new data
- Inference: ensure that outputs satisfy constraints
Inference

- Assign values to the variables of interest (output; states) in a way that maximizes your objective function and satisfies constraints
Inference

- Boolean Constraint Satisfaction Problem
  - $V$ – set of variables
  - Cost – $c: V \rightarrow [0,1]$
  - Clauses – model constraints
  - Satisfying assignment – $\tau: V \rightarrow \{0,1\}$

- Find the solution $\tau$ that minimize the cost
  $$C(\tau) = \sum_{i=1}^{n} \tau(v_i)c(v_i)$$
Inference for Shallow Parsing

- Define a variable $v_i$ for each possible chunk
- Cost of each chunk = $-P(v_i \text{ is a chunk})$
- For any two overlapping chunks:
  $$(\neg v_i \lor \neg v_j)$$

- Find the solution $\tau$ that minimizes the cost
  $$C(\tau) = \sum_{i=1}^{n} \tau(v_i)c(v_i)$$

- This is a good objective function – it maximizes the expected number of correct chunks.
- CSP in general is hard
- Structure of the constraints yields a problem that can be solved by a shortest path algorithm
Inference for Shallow Parsing

\[
\begin{array}{ccccccc}
O_1 & O_2 & O_3 & O_4 & O_5 & O_6 & O_7 \\
\end{array}
\]

\[
\begin{array}{ccccccc}
\begin{array}{c}
O \\
\end{array} & \begin{array}{c}
O \\
\end{array} & \begin{array}{c}
O \\
\end{array} & \begin{array}{c}
C \\
\end{array} & \begin{array}{c}
C \\
\end{array} & \begin{array}{c}
C \\
\end{array} & \begin{array}{c}
C \\
\end{array} \\
\end{array}
\]

\[-0.48 \quad -0.36 \quad -0.22 \quad -0.66 \quad -0.56 \quad -0.44 \quad -0.77\]
## Experimental Results

### NP Prediction

<table>
<thead>
<tr>
<th></th>
<th>Recall</th>
<th>Prediction</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM (POS)</td>
<td>89.08</td>
<td>86.62</td>
<td>87.83</td>
</tr>
<tr>
<td>HMM with Classifiers (POS)</td>
<td>91.49</td>
<td>91.53</td>
<td>91.51</td>
</tr>
<tr>
<td>HMM with Classifiers</td>
<td>93.85</td>
<td>93.46</td>
<td>93.65</td>
</tr>
<tr>
<td>PMM</td>
<td>94.28</td>
<td>94.02</td>
<td>94.15</td>
</tr>
<tr>
<td>CSCL</td>
<td>94.12</td>
<td>93.45</td>
<td>93.78</td>
</tr>
</tbody>
</table>
Other Approaches to Shallow Parsing

- MEMM
- CRF
- Global Perceptron (Collins’)
- Others (e.g., global SVM)
Maximum Entropy Markov Model (MEMM)

- Similar to PMM

\[
\arg\max_{s_k, s_{k-1}, \ldots, s_1} \prod_{t=2}^{k} P(s_t | s_{t-1}, O) \cdot P(s_1 | O)
\]

- Each term is learned with maximum entropy model

Brown blamed Mr. Bob making things dirty Mr. I O B I O B O
Conditional Random Field (CRF)

- Similar to PMM, but based on Markov random field theory (undirected graph)

\[
\begin{align*}
\arg\max_{s_k, s_{k-1}, \ldots, s_1} P(s_k, s_{k-1}, \ldots, s_1|O) &= \arg\max_{s_k, s_{k-1}, \ldots, s_1} \frac{1}{Z(O)} \exp\left(\sum_i w_i f_i(s_k, s_{k-1}, O)\right) \\
\end{align*}
\]
Collins’ Perceptron Training for HMM

- Similar to HMM, but use Perceptron algorithm to learn

\[
\arg \max_{s_k, s_{k-1}, \ldots, s_1} P(s_k, s_{k-1}, \ldots, s_1 | O)
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} P(o_k | s_k) \cdot \left[ \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1)
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} \log(P(o_k | s_k)) \cdot \left[ \prod_{t=1}^{k-1} P(s_{t+1} | s_t) \cdot P(o_t | s_t) \right] \cdot P(s_1)
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} \log(P(o_k | s_k)) + \left[ \sum_{t=1}^{k-1} \log(P(s_{t+1} | s_t)) + \log(P(o_t | s_t)) \right] + \log(P(s_1))
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} w_{ifi}(o_k, s_k) + \left[ \sum_{t=1}^{k-1} utgt(s_{t+1}, s_t) + w_{ftf}(o_t, s_t) \right] + u_{1g1}(s_1)
\]
Maximum Entropy Markov Model (MEMM)

\[ P(s_k|s_{k-1}, O) = \frac{1}{Z(s_{k-1}, O)} \exp(\sum_i w_i f_i(s_k, s_{k-1}, O)) \]

\[ \arg \max_{s_k, s_{k-1}, \ldots, s_1} P(s_k, s_{k-1}, \ldots, s_1|O) \]

\[ = \arg \max_{s_k, s_{k-1}, \ldots, s_1} \left[ \prod_{t=2}^{k} P(s_t|s_{t-1}, O) \right] \cdot P(s_1|O) \]

\[ = \arg \max_{s_k, s_{k-1}, \ldots, s_1} \left[ \prod_{t=2}^{k} \frac{1}{Z(s_{t-1}, O)} \exp(\sum_i w_i f_i(s_t, s_{t-1}, O)) \right] \]

\[ \cdot \frac{1}{Z(O)} \exp(\sum_i w_i f_i(s_1, O)) \]
Maximum Entropy Markov Model (MEMM)

\[
\arg\max_{s_k, s_{k-1}, \ldots, s_1} P(s_k, s_{k-1}, \ldots, s_1 | O)
\]

\[
= \arg\max_{s_k, s_{k-1}, \ldots, s_1} \log\left( \prod_{t=2}^{k} \frac{1}{Z(s_{t-1}, O)} \exp\left( \sum_i w_i f_i(s_t, s_{t-1}, O) \right) \right) \\
\cdot \frac{1}{Z(O)} \exp\left( \sum_i w_i f_i(s_1, O) \right)
\]

\[
= \arg\max_{s_k, s_{k-1}, \ldots, s_1} \left[ \sum_{t=2}^{k} \log\left( \frac{1}{Z(s_{t-1}, O)} \right) + \sum_i w_i f_i(s_t, s_{t-1}, O) \right] \\
+ \log\left( \frac{1}{Z(O)} \right) + \sum_i w_i f_i(s_1, O)
\]
Conditional Random Field (CRF)

\[
\arg \max_{s_k, s_{k-1}, \ldots, s_1} P(s_k, s_{k-1}, \ldots, s_1 | O)
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} \frac{1}{Z(O)} \exp \left( \sum_i w_i f_i(s_k, s_{k-1}, O) \right)
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} \log \left( \frac{1}{Z(O)} \exp \left( \sum_i w_i f_i(s_k, s_{k-1}, O) \right) \right)
\]

\[
= \arg \max_{s_k, s_{k-1}, \ldots, s_1} \log \left( \frac{1}{Z(O)} \right) + \sum_i w_i f_i(s_k, s_{k-1}, O)
\]
Conclusions

- All approaches use linear representation
- The differences are
  - Features
  - How to learn weights
  - Training Paradigms:
    - Global Training (HMM, CRF, Global Perceptron)
    - Modular Training (PMM, MEMM, CSCL)
      - These approaches are easier to train, but may require additional mechanisms to enforce global constraints.