Today

• Finish word sense disambiguation

• Midterm Review

• The midterm is Thursday during class time. H students and 20 regular students are assigned to 517 Hamilton.

• Make-up for those with same time midterm is Thurs 6pm. For those with serious problems, second make-up Mon 6pm, but will be harder. RECOMMENDATION: Thurs 6pm if you can.

• Review questions and answer on NLP website.

• On Thursday, I will not be here. I am at a meeting in Maryland. Tas will proctor
Naïve Bayes Test

• On a corpus of examples of uses of the word line, naïve Bayes achieved about 73% correct

• Good?
Decision Lists: another popular method

- A case statement....

<table>
<thead>
<tr>
<th>Rule</th>
<th>Sense</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>fish</em> within window</td>
<td>bass¹</td>
</tr>
<tr>
<td><em>striped bass</em></td>
<td>bass¹</td>
</tr>
<tr>
<td><em>guitar</em> within window</td>
<td>bass²</td>
</tr>
<tr>
<td><em>bass player</em></td>
<td>bass²</td>
</tr>
<tr>
<td><em>piano</em> within window</td>
<td>bass²</td>
</tr>
<tr>
<td><em>tenor</em> within window</td>
<td>bass²</td>
</tr>
<tr>
<td><em>sea bass</em></td>
<td>bass¹</td>
</tr>
<tr>
<td><em>play/N bass</em></td>
<td>bass²</td>
</tr>
<tr>
<td><em>river</em> within window</td>
<td>bass¹</td>
</tr>
<tr>
<td><em>violin</em> within window</td>
<td>bass²</td>
</tr>
<tr>
<td><em>salmon</em> within window</td>
<td>bass¹</td>
</tr>
<tr>
<td><em>on bass</em></td>
<td>bass²</td>
</tr>
<tr>
<td><em>bass are</em></td>
<td>bass¹</td>
</tr>
</tbody>
</table>
Learning Decision Lists

- Restrict the lists to rules that test a single feature (1-decisionlist rules)
- Evaluate each possible test and rank them based on how well they work.
- Glue the top-N tests together and call that your decision list.
On a binary (homonymy) distinction used the following metric to rank the tests

\[
\frac{P(Sense_1 \mid Feature)}{P(Sense_2 \mid Feature)}
\]

This gives about 95% on this test...
WSD Evaluations and baselines

- *In vivo versus in vitro* evaluation
- In vitro evaluation is most common now
  - Exact match *accuracy*
    - % of words tagged identically with manual sense tags
  - Usually evaluate using held-out data from same labeled corpus
    - Problems?
    - Why do we do it anyhow?

- Baselines
  - Most frequent sense
  - The Lesk algorithm
Most Frequent Sense

- Wordnet senses are ordered in frequency order
- So “most frequent sense” in wordnet = “take the first sense”

<table>
<thead>
<tr>
<th>Freq</th>
<th>Synset</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>338</td>
<td>plant\textsuperscript1, works, industrial plant</td>
<td>buildings for carrying on industrial labor</td>
</tr>
<tr>
<td>207</td>
<td>plant\textsuperscript2, flora, plant life</td>
<td>a living organism lacking the power of locomotion</td>
</tr>
<tr>
<td>2</td>
<td>plant\textsuperscript3</td>
<td>something planted secretly for discovery by another</td>
</tr>
<tr>
<td>0</td>
<td>plant\textsuperscript4</td>
<td>an actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience</td>
</tr>
</tbody>
</table>
Ceiling

• Human inter-annotator agreement
  • Compare annotations of two humans
  • On same data
  • Given same tagging guidelines

• Human agreements on all-words corpora with Wordnet style senses
  • 75%-80%
Problems

• Given these general ML approaches, how many classifiers do I need to perform WSD robustly
  • One for each ambiguous word in the language
• How do you decide what set of tags/labels/senses to use for a given word?
  • Depends on the application
WordNet Bass

• Tagging with this set of senses is an impossibly hard task that’s probably overkill for any realistic application

1. bass - (the lowest part of the musical range)
2. bass, bass part - (the lowest part in polyphonic music)
3. bass, basso - (an adult male singer with the lowest voice)
4. sea bass, bass - (flesh of lean-fleshed saltwater fish of the family Serranidae)
5. freshwater bass, bass - (any of various North American lean-fleshed freshwater fishes especially of the genus Micropterus)
6. bass, bass voice, basso - (the lowest adult male singing voice)
7. bass - (the member with the lowest range of a family of musical instruments)
8. bass - (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)
Senseval History

- ACL-SIGLEX workshop (1997)
  - Yarowsky and Resnik paper
- SENSEVAL-I (1998)
  - Lexical Sample for English, French, and Italian
- SENSEVAL-II (Toulouse, 2001)
  - Lexical Sample and All Words
  - Organization: Kilkgarriff (Brighton)
- SENSEVAL-IV -> SEMEVAL (2007)
- SEMEVAL (2010)

SLIDE ADAPTED FROM CHRIS MANNING
WSD Performance

- Varies widely depending on how difficult the disambiguation task is
- Accuracies of over 90% are commonly reported on some of the classic, often fairly easy, WSD tasks (pike, star, interest)
- Senseval brought careful evaluation of difficult WSD (many senses, different POS)
- Senseval 1: more fine grained senses, wider range of types:
  - Overall: about 75% accuracy
  - Nouns: about 80% accuracy
  - Verbs: about 70% accuracy
Summary

• Lexical Semantics
  • Homonymy, Polysemy, Synonymy
  • Thematic roles
• Computational resource for lexical semantics
  • WordNet
• Task
  • Word sense disambiguation
• After midterm: semantic parsing, distributional semantics, neural nets
Requested topics

• POS tagging
• HMM
• Early parsing algorithm
POS tagging
POS tagging as a sequence classification task

• We are given a sentence (an “observation” or “sequence of observations”)
  • Secretariat is expected to race tomorrow

• What is the best sequence of tags which corresponds to this sequence of observations?

• Probabilistic view:
  • Consider all possible sequences of tags
  • Choose the tag sequence which is most probable given the observation sequence of n words w1...wn.
Getting to HMM

• Out of all sequences of $n$ tags $t_1...t_n$ want the single tag sequence such that $P(t_1...t_n|w_1...w_n)$ is highest.

\[ \hat{t}_1^n = \text{argmax}_{t_1^n} P(t_1^n|w_1^n) \]

• Hat $^\wedge$ means “our estimate of the best one”

• $\text{Argmax}_x f(x)$ means “the $x$ such that $f(x)$ is maximized”
Using Bayes Rule

\[ P(x|y) = \frac{P(y|x)P(x)}{P(y)} \]

\[ \hat{t}_1^n = \arg\max_{t_1^n} \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)} \]

\[ \hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n | t_1^n)P(t_1^n) \]
\[
\hat{t}_1^n = \arg \max_{t_1^n} P(t_1^n | w_1^n) \\
\]

\[
P(x|y) = \frac{P(y|x)P(x)}{P(y)}
\]

\[
\hat{t}_1^n = \arg \max_{t_1^n} \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}
\]

\[
\hat{t}_1^n = \arg \max_{t_1^n} P(w_1^n | t_1^n)P(t_1^n)
\]
Likelihood and prior

\[ \hat{t}_1^n = \arg\max_{t^n_1} \left\{ \hat{w}^n_{t_1^n} \right\} \]

\[ P(w^n_{t_1^n}) \approx \prod_{i=1}^{n} P(w_i|t_i) \]

\[ P(t^n_{t_1^n}) \approx \prod_{i=1}^{n} P(t_i|t_{i-1}) \]

\[ \hat{t}_1^n = \arg\max_{t^n_1} P(t^n_{t_1^n}|w^n_{t_1^n}) \approx \arg\max_{t^n_1} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1}) \]
Two kinds of probabilities (1)

- Tag transition probabilities $p(t_i | t_{i-1})$
  - Determiners likely to precede adjs and nouns
    - That/DT flight/NN
    - The/DT yellow/JJ hat/NN
    - So we expect $P(\text{NN}|\text{DT})$ and $P(\text{JJ}|\text{DT})$ to be high
    - But $P(\text{DT}|\text{JJ})$ to be low
  - Compute $P(\text{NN}|\text{DT})$ by counting in a labeled corpus:
    
    $$P(t_i | t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

    $$P(\text{NN}|\text{DT}) = \frac{C(\text{DT}, \text{NN})}{C(\text{DT})} = \frac{56,509}{116,454} = .49$$
Two kinds of probabilities (2)

- Word likelihood probabilities $p(w_i | t_i)$
  - VBZ (3sg Pres verb) likely to be “is”
  - Compute $P(is|VBZ)$ by counting in a labeled corpus:

$$P(w_i | t_i) = \frac{C(t_i, w_i)}{C(t_i)}$$

$$P(is|VBZ) = \frac{C(VBZ, is)}{C(VBZ)} = \frac{10,073}{21,627} = .47$$
An Example: the verb “race”

• Secretariat/NNP is/VBZ expected/VBN to/TO race/VB tomorrow/NR

• People/NNS continue/VB to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN

• How do we pick the right tag?
Disambiguating “race”

(a) Secretariat is expected to race tomorrow

(b) Secretariat is expected to race tomorrow
Disambiguating “race”

(a) Secretariat is expected to race tomorrow

(b) Secretariat is expected to race tomorrow
Disambiguating “race”

(a) Secretariat is expected to race tomorrow

(b) Secretariat is expected to race tomorrow
Disambiguating “race”
- \( P(\text{NN} | \text{TO}) = 0.00047 \)
- \( P(\text{VB} | \text{TO}) = 0.83 \)
- \( P(\text{race} | \text{NN}) = 0.00057 \)
- \( P(\text{race} | \text{VB}) = 0.00012 \)
- \( P(\text{NR} | \text{VB}) = 0.0027 \)
- \( P(\text{NR} | \text{NN}) = 0.0012 \)

- \( P(\text{VB} | \text{TO})P(\text{NR} | \text{VB})P(\text{race} | \text{VB}) = 0.00000027 \)
- \( P(\text{NN} | \text{TO})P(\text{NR} | \text{NN})P(\text{race} | \text{NN}) = 0.0000000032 \)
- So we (correctly) choose the verb reading,
Problem

• Observation likelihood: “Promise”

Count (#promise, VB)/#all verbs

• I promise to back the bill (N).
  I promise to back the bill (V)
Hidden Markov Models

• We don’t observe POS tags
  • We infer them from the words we see

• Observed events

• Hidden events
Hidden Markov Model

• For Markov chains, the output symbols are the same as the states.
  ▪ See *hot* weather: we’re in state *hot*

• But in part-of-speech tagging (and other things)
  ▪ The output symbols are *words*
  ▪ The hidden states are *part-of-speech tags*

• So we need an extension!

• A **Hidden Markov Model** is an extension of a Markov chain in which the input symbols are not the same as the states.

• This means we don’t know which state we are in.
Hidden Markov Models

- States $Q = q_1, q_2 \ldots q_N$;
- Observations $O = o_1, o_2 \ldots o_N$;
  - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \ldots v_V\}$
- Transition probabilities
  - Transition probability matrix $A = \{a_{ij}\}$
    
    $a_{ij} = P(q_t = j \mid q_{t-1} = i) \quad 1 \leq i, j \leq N$

- Observation likelihoods
  - Output probability matrix $B = \{b_i(k)\}$
    
    $b_i(k) = P(X_t = o_k \mid q_t = i)$

- Special initial probability vector $\pi$

  $\pi_i = P(q_1 = i) \quad 1 \leq i \leq N$
Hidden Markov Models

- Some constraints

\[
\sum_{j=1}^{N} a_{ij} = 1; \quad 1 \leq i \leq N
\]

\[
\pi_i = P(q_1 = i) \quad 1 \leq i \leq N
\]

\[
\sum_{k=1}^{M} b_i(k) = 1
\]

\[
\sum_{j=1}^{N} \pi_j = 1
\]
Assumptions

- Markov assumption:

\[ P(q_i | q_1 \ldots q_{i-1}) = P(q_i | q_{i-1}) \]

- Output-independence assumption

\[ P(o_t | O_1^{t-1}, q_1^t) = P(o_t | q_t) \]
Three fundamental Problems for HMMs

- **Likelihood**: Given an HMM $\lambda = (A, B)$ and an observation sequence $O$, determine the likelihood $P(O, \lambda)$.

- **Decoding**: Given an observation sequence $O$ and an HMM $\lambda = (A, B)$, discover the best hidden state sequence $Q$.

- **Learning**: Given an observation sequence $O$ and the set of states in the HMM, learn the HMM parameters $A$ and $B$.

*What kind of data would we need to learn the HMM parameters?*
Decoding

- The best hidden sequence
  - Weather sequence in the ice cream task
  - POS sequence given an input sentence
- We could use argmax over the probability of each possible hidden state sequence
  - *Why not?*

- Viterbi algorithm
  - Dynamic programming algorithm
  - Uses a dynamic programming trellis
    - Each trellis cell represents, $v_t(j)$, represents the probability that the HMM is in state $j$ after seeing the first $t$ observations and passing through the most likely state sequence
Viterbi intuition: we are looking for the best ‘path’

promised to back the bill
Intuition

• The value in each cell is computed by taking the MAX over all paths that lead to this cell.

\[ v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i) a_{ij} b_j(o_t) \]

• An extension of a path from state i at time t-1 is computed by multiplying:

- \( v_{t-1}(i) \) \textit{the previous Viterbi path probability} from the previous time step
- \( a_{ij} \) \textit{the transition probability} from previous state \( q_i \) to current state \( q_j \)
- \( b_j(o_t) \) \textit{the state observation likelihood} of the observation symbol \( o_t \) given the current state \( j \)
The Viterbi Algorithm

```
function VITERBI(observations of len T, state-graph) returns best-path

num-states ← NUM-OF-STATES(state-graph)
Create a path probability matrix viterbi[num-states+2,T+2]
viterbi[0,0] ← 1.0
for each time step t from 1 to T do
  for each state s from 1 to num-states do
    viterbi[s,t] ← \[\max_{1 \leq s' \leq \text{num-states}} viterbi[s',t-1] \cdot a_{s',s} \cdot b_s(o_t)\]
    backpointer[s,t] ← \[\arg\max_{1 \leq s' \leq \text{num-states}} viterbi[s',t-1] \cdot a_{s',s}\]
  Backtrace from highest probability state in final column of viterbi[] and return path
```
## The A matrix for the POS HMM

<table>
<thead>
<tr>
<th></th>
<th>VB</th>
<th>TO</th>
<th>NN</th>
<th>PPSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;s&gt;</td>
<td>.019</td>
<td>.0043</td>
<td>.041</td>
<td>.067</td>
</tr>
<tr>
<td>VB</td>
<td>.0038</td>
<td>.035</td>
<td>.047</td>
<td>.0070</td>
</tr>
<tr>
<td>TO</td>
<td>.83</td>
<td>0</td>
<td>.00047</td>
<td>0</td>
</tr>
<tr>
<td>NN</td>
<td>.0040</td>
<td>.016</td>
<td>.087</td>
<td>.0045</td>
</tr>
<tr>
<td>PPSS</td>
<td>.23</td>
<td>.00079</td>
<td>.0012</td>
<td>.00014</td>
</tr>
</tbody>
</table>

Figure 4.15 Tag transition probabilities (the $a$ array, $p(t_i|t_{i-1})$) computed from the 87-tag Brown corpus without smoothing. The rows are labeled with the conditioning event; thus $P(PPSS|VB)$ is .0070. The symbol <s> is the start-of-sentence symbol.

What is $P(VB|TO)$? What is $P(NN|TO)$? Why does this make sense?

What is $P(TO|VB)$? What is $P(TO|NN)$? Why does this make sense?
Look at $P(\text{want} | \text{VB})$ and $P(\text{want} | \text{NN})$. Give an explanation for the difference in the probabilities.
<table>
<thead>
<tr>
<th></th>
<th>VB</th>
<th>TO</th>
<th>NN</th>
<th>PPSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;s&gt;</td>
<td>.019</td>
<td>.0043</td>
<td>.041</td>
<td>.067</td>
</tr>
<tr>
<td>VB</td>
<td>.0038</td>
<td>.035</td>
<td>.047</td>
<td>.0070</td>
</tr>
<tr>
<td>TO</td>
<td>.83</td>
<td>0</td>
<td>.00047</td>
<td>0</td>
</tr>
<tr>
<td>NN</td>
<td>.0040</td>
<td>.016</td>
<td>.087</td>
<td>.0045</td>
</tr>
<tr>
<td>PPSS</td>
<td>.23</td>
<td>.00079</td>
<td>.0012</td>
<td>.00014</td>
</tr>
</tbody>
</table>

**Figure 4.15**  Tag transition probabilities (the $a$ array, $p(t_i|t_{i-1})$) computed from the 87-tag Brown corpus without smoothing. The rows are labeled with the conditioning event; thus $P(PPSS|VB)$ is .0070. The symbol $<s>$ is the start-of-sentence symbol.

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>want</th>
<th>to</th>
<th>race</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>0</td>
<td>.0093</td>
<td>0</td>
<td>.00012</td>
</tr>
<tr>
<td>TO</td>
<td>0</td>
<td>0</td>
<td>.99</td>
<td>0</td>
</tr>
<tr>
<td>NN</td>
<td>0</td>
<td>.000054</td>
<td>0</td>
<td>.00057</td>
</tr>
<tr>
<td>PPSS</td>
<td>.37</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 4.16**  Observation likelihoods (the $b$ array) computed from the 87-tag Brown corpus without smoothing.
Problem

- I want to race (possible states: PPS VB TO NN)
\[ v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i) a_{ij} b_j(o_t) \]
The A matrix for the POS HMM

<table>
<thead>
<tr>
<th></th>
<th>VB</th>
<th>TO</th>
<th>NN</th>
<th>PPSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;s&gt;</td>
<td>.019</td>
<td>.0043</td>
<td>.041</td>
<td>.067</td>
</tr>
<tr>
<td>VB</td>
<td>.0038</td>
<td>.035</td>
<td>.047</td>
<td>.0070</td>
</tr>
<tr>
<td>TO</td>
<td>.83</td>
<td>0</td>
<td>.00047</td>
<td>0</td>
</tr>
<tr>
<td>NN</td>
<td>.0040</td>
<td>.016</td>
<td>.087</td>
<td>.0045</td>
</tr>
<tr>
<td>PPSS</td>
<td>.23</td>
<td>.00079</td>
<td>.0012</td>
<td>.00014</td>
</tr>
</tbody>
</table>

Figure 4.15  Tag transition probabilities (the $a$ array, $p(t_i|t_{i-1})$) computed from the 87-tag Brown corpus without smoothing. The rows are labeled with the conditioning event; thus $P(PPSS|VB)$ is .0070. The symbol $<s>$ is the start-of-sentence symbol.
\[ v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i) a_{ij} b_j(o_t) \]

\[ v_1(4) = 0.041 \times 0 = 0 \]

J = NN

I = S

\[ \text{backtrace} \]

\[ t = 1 \]

i \hspace{1cm} \text{want} \hspace{1cm} \text{to} \hspace{1cm} \text{race}
The B matrix for the POS HMM

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>want</th>
<th>to</th>
<th>race</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>0</td>
<td>.0093</td>
<td>0</td>
<td>.00012</td>
</tr>
<tr>
<td>TO</td>
<td>0</td>
<td>0</td>
<td>.99</td>
<td>0</td>
</tr>
<tr>
<td>NN</td>
<td>0</td>
<td>.000054</td>
<td>0</td>
<td>.00057</td>
</tr>
<tr>
<td>PPSS</td>
<td>.37</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 4.16** Observation likelihoods (the \( b \) array) computed from the 87-tag Brown corpus without smoothing.

Look at \( P(\text{want}|\text{VB}) \) and \( P(\text{want}|\text{NN}) \). Give an explanation for the difference in the probabilities.
\[ v_t(j) = \max_{1 \leq i \leq N-1} v_{X_1(i)} a_{ij} b_j(o_t) \]

\[ v_1(4) = 0.041 	imes 0 = 0 \]

\[ v_1(3) = 0.0043 	imes 0 = 0 \]

\[ v_1(2) = 0.019 	imes 0 = 0 \]

\[ v_1(1) = 0.067 	imes 0.37 = 0.025 \]

I=S

<table>
<thead>
<tr>
<th>q</th>
<th>q_0</th>
<th>q_1</th>
<th>q_2</th>
<th>q_3</th>
<th>q_4</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>NN</td>
<td>TO</td>
<td>VB</td>
<td>VB</td>
<td>TO</td>
</tr>
<tr>
<td>S</td>
<td>PP</td>
<td>SS</td>
<td>SS</td>
<td>SS</td>
<td>SS</td>
</tr>
</tbody>
</table>

\( t=1 \)

i

want

to

race

\( o_1 \)

\( o_2 \)

\( o_3 \)

\( o_4 \)
\[
\begin{align*}
V_t(j) & = \max_{1 \leq i \leq N-1} V_X(i) a_{ij} b_j(o_t) \\
V_{1}(4) & = 0.041 \times 0 = 0
\end{align*}
\]
Show the 4 formulas you would use to compute the value at this node and the max.
Early Algorithm
Earley Algorithm

- March through chart left-to-right.
- At each step, apply 1 of 3 operators
  - Predictor
    - Create new states representing top-down expectations
  - Scanner
    - Match word predictions (rule with word after dot) to words
  - Completer
    - When a state is complete, see what rules were looking for that completed constituent
Predictor

- Given a state
  - With a non-terminal to right of dot (not a part-of-speech category)
  - Create a new state for each expansion of the non-terminal
  - Place these new states into same chart entry as generated state, beginning and ending where generating state ends.
- So predictor looking at
  - $S \rightarrow \cdot \ VP \ [0,0]$
- results in
  - $VP \rightarrow \cdot \ Verb \ [0,0]$
  - $VP \rightarrow \cdot \ Verb \ NP \ [0,0]$
Scanner

• Given a state
  • With a non-terminal to right of dot that is a part-of-speech category
  • If the next word in the input matches this POS
  • Create a new state with dot moved over the non-terminal
  • So scanner looking at $VP \rightarrow . \text{Verb NP}[0,0]$
  • If the next word, “book”, can be a verb, add new state:
    • $VP \rightarrow \text{Verb} . \text{NP}[0,1]$
  • Add this state to chart entry following current one
  • Note: Earley algorithm uses top-down input to disambiguate POS! Only POS predicted by some state can get added to chart!
Completer

- Applied to a state when its dot has reached right end of role.
- Parser has discovered a category over some span of input.
- Find and advance all previous states that were looking for this category
  - copy state, move dot, insert in current chart entry
- Given:
  - NP -> Det Nominal . [1,3]
  - VP -> Verb. NP [0,1]
- Add
  - VP -> Verb NP . [0,3]
How do we know we are done?

- Find an S state in the final column that spans from 0 to n+1 and is complete.
- If that’s the case you’re done.
  - $S \rightarrow \alpha \cdot [0,n+1]$
Earley

- More specifically...

1. Predict all the states you can upfront

2. Read a word
   - 1. Extend states based on matches
   - 2. Add new predictions
   - 3. Go to 2

3. Look at N+1 to see if you have a winner
Example

- Book that flight
- We should find... an S from 0 to 3 that is a completed state...
<table>
<thead>
<tr>
<th>CFG Rule</th>
<th>Grammar Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>VP → V</td>
</tr>
<tr>
<td>S → Aux NP VP</td>
<td>PP → Prep NP</td>
</tr>
<tr>
<td>NP → Det Nom</td>
<td>N → old</td>
</tr>
<tr>
<td>NP → PropN</td>
<td>V → dog</td>
</tr>
<tr>
<td>Nom → Adj Nom</td>
<td>Aux → does</td>
</tr>
<tr>
<td>Nom → N</td>
<td>Prep → from</td>
</tr>
<tr>
<td>Nom → N Nom</td>
<td>PropN → Bush</td>
</tr>
<tr>
<td>Nom → Nom PP</td>
<td>Det → that</td>
</tr>
<tr>
<td>VP → V NP</td>
<td>Adj → old</td>
</tr>
<tr>
<td>S → NP VP, <strong>S -&gt; VP</strong></td>
<td>VP → V</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>S → Aux NP VP</td>
<td>PP -&gt; Prep NP</td>
</tr>
<tr>
<td><strong>NP → Det Nom</strong></td>
<td>N → old</td>
</tr>
<tr>
<td><strong>NP → PropN, NP -&gt; Pro</strong></td>
<td>V → dog</td>
</tr>
<tr>
<td><strong>Aux → does</strong></td>
<td></td>
</tr>
<tr>
<td>Nom → N</td>
<td>Prep → from</td>
</tr>
<tr>
<td><strong>Nom → N Nom</strong></td>
<td>PropN → Bush</td>
</tr>
<tr>
<td>Nom → Nom PP</td>
<td>Det → <em>that</em></td>
</tr>
<tr>
<td>VP → V NP, VP -&gt; V</td>
<td>Adj → old</td>
</tr>
<tr>
<td>NP PP, VP -&gt; V PP, VP -&gt; VP PP</td>
<td></td>
</tr>
<tr>
<td>S \rightarrow NP \ VP, S \rightarrow VP</td>
<td>VP \rightarrow V</td>
</tr>
<tr>
<td>S \rightarrow Aux \ NP \ VP</td>
<td>PP \rightarrow Prep \ NP</td>
</tr>
<tr>
<td>NP \rightarrow Det \ Nom</td>
<td>N \rightarrow old \</td>
</tr>
<tr>
<td>NP \rightarrow PropN, NP \rightarrow Pro</td>
<td>V \rightarrow dog \</td>
</tr>
<tr>
<td>Aux \rightarrow does</td>
<td></td>
</tr>
<tr>
<td>Nom \rightarrow N</td>
<td>Prep \rightarrow from \</td>
</tr>
<tr>
<td>Nom \rightarrow N \ Nom</td>
<td>PropN \rightarrow Bush \</td>
</tr>
<tr>
<td>Nom \rightarrow Nom \ PP</td>
<td>Det \rightarrow that \</td>
</tr>
<tr>
<td>VP \rightarrow V \ NP, VP \rightarrow V</td>
<td>Adj \rightarrow old \</td>
</tr>
</tbody>
</table>
Example

Chart[0]  S0  $\gamma \rightarrow S$

S1  $S \rightarrow NP \ VP$

S2  $S \rightarrow Aux \ NP \ VP$

S3  $S \rightarrow VP$

S4  $NP \rightarrow Pronoun$

S5  $NP \rightarrow Proper-Noun$

S6  $NP \rightarrow Det \ Nominal$

S7  $VP \rightarrow Verb$

S8  $VP \rightarrow Verb \ NP$

S9  $VP \rightarrow Verb \ NP \ PP$

S10  $VP \rightarrow Verb \ PP$

S11  $VP \rightarrow VP \ PP$

[0,0] Dummy start state

[0,0] Predictor

[0,0] Predictor

[0,0] Predictor

[0,0] Predictor

[0,0] Predictor

[0,0] Predictor

[0,0] Predictor

[0,0] Predictor

[0,0] Predictor
Example

<table>
<thead>
<tr>
<th>Chart[1]</th>
<th>S12</th>
<th>Verb → book •</th>
</tr>
</thead>
<tbody>
<tr>
<td>S13</td>
<td>VP → Verb •</td>
<td></td>
</tr>
<tr>
<td>S14</td>
<td>VP → Verb • NP</td>
<td></td>
</tr>
<tr>
<td>S15</td>
<td>VP → Verb • NP PP</td>
<td></td>
</tr>
<tr>
<td>S16</td>
<td>VP → Verb • PP</td>
<td></td>
</tr>
<tr>
<td>S17</td>
<td>S → VP •</td>
<td></td>
</tr>
<tr>
<td>S18</td>
<td>VP → VP • PP</td>
<td></td>
</tr>
<tr>
<td>S19</td>
<td>NP → • Pronoun</td>
<td></td>
</tr>
<tr>
<td>S20</td>
<td>NP → • Proper-Noun</td>
<td></td>
</tr>
<tr>
<td>S21</td>
<td>NP → • Det Nominal</td>
<td></td>
</tr>
<tr>
<td>S22</td>
<td>PP → • Prep NP</td>
<td></td>
</tr>
</tbody>
</table>

Scanner
Completer
Completer
Completer
Completer
Predictor
Completer
Predictor
Predictor
Predictor
Example

Chart[1]  S12  Verb → book
          [0,1]
          Scanner
          Completer
          Completer
          Completer

          S13  VP → Verb
          [0,1]
          Predictor

          S14  VP → Verb • NP
          [0,1]
          Predictor

          S15  VP → Verb • NP PP
          [0,0]
          Predictor

          S16  VP → Verb • PP
          [0,0]
          Predictor

          S17  S → VP
          [0,1]
          Completer

          S18  VP → VP • PP
          [0,1]
          Completer

          S19  NP → • Pronoun
          [1,1]
          Predictor

          S20  NP → • Proper-Noun
          [1,1]
          Predictor

          S21  NP → • Det Nominal
          [1,1]
          Predictor

          S22  PP → • Prep NP
          [1,1]
          Predictor
Example

<table>
<thead>
<tr>
<th>Chart[2]</th>
<th>Rule</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>S23</td>
<td>Det → that •</td>
<td>[1,2]</td>
</tr>
<tr>
<td>S24</td>
<td>NP → Det • Nominal</td>
<td>[1,2]</td>
</tr>
<tr>
<td>S25</td>
<td>Nominal → • Noun</td>
<td>[2,2]</td>
</tr>
<tr>
<td>S26</td>
<td>Nominal → • Nominal Noun</td>
<td>[2,2]</td>
</tr>
<tr>
<td>S27</td>
<td>Nominal → • Nominal PP</td>
<td>[2,2]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chart[3]</th>
<th>Rule</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>S28</td>
<td>Noun → flight •</td>
<td>[2,3]</td>
</tr>
<tr>
<td>S29</td>
<td>Nominal → Noun •</td>
<td>[2,3]</td>
</tr>
<tr>
<td>S30</td>
<td>NP → Det Nominal •</td>
<td>[1,3]</td>
</tr>
<tr>
<td>S31</td>
<td>Nominal → Nominal • Noun</td>
<td>[2,3]</td>
</tr>
<tr>
<td>S32</td>
<td>Nominal → Nominal • PP</td>
<td>[2,3]</td>
</tr>
<tr>
<td>S33</td>
<td>VP → Verb NP •</td>
<td>[0,3]</td>
</tr>
<tr>
<td>S34</td>
<td>VP → Verb NP • PP</td>
<td>[0,3]</td>
</tr>
<tr>
<td>S35</td>
<td>PP → • Prep NP</td>
<td>[3,3]</td>
</tr>
<tr>
<td>S36</td>
<td>S → VP •</td>
<td>[0,3]</td>
</tr>
</tbody>
</table>
Details

• What kind of algorithms did we just describe
  • Not parsers – recognizers
    • The presence of an S state with the right attributes in the right place indicates a successful recognition.
    • But no parse tree... no parser
    • That’s how we solve (not) an exponential problem in polynomial time
Converting Earley from Recognizer to Parser

• With the addition of a few pointers we have a parser
• Augment the “Completer” to point to where we came from.
Augmenting the chart with structural information

<table>
<thead>
<tr>
<th>Chart[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>S8</td>
</tr>
<tr>
<td>$Verb \rightarrow book$</td>
</tr>
<tr>
<td>S9</td>
</tr>
<tr>
<td>$VP \rightarrow Verb$</td>
</tr>
<tr>
<td>S10</td>
</tr>
<tr>
<td>$S \rightarrow VP$</td>
</tr>
<tr>
<td>S11</td>
</tr>
<tr>
<td>$VP \rightarrow Verb \cdot NP$</td>
</tr>
<tr>
<td>S12</td>
</tr>
<tr>
<td>$NP \rightarrow \bullet Det \cdot NOMINAL$</td>
</tr>
<tr>
<td>S13</td>
</tr>
<tr>
<td>$NP \rightarrow \bullet Proper\cdot Noun$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chart[2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Det</td>
</tr>
<tr>
<td>$\rightarrow that$</td>
</tr>
<tr>
<td>NP</td>
</tr>
<tr>
<td>$\rightarrow Det \cdot NOMINAL$</td>
</tr>
<tr>
<td>NOMINAL</td>
</tr>
<tr>
<td>$\rightarrow \bullet Noun$</td>
</tr>
<tr>
<td>NOMINAL</td>
</tr>
<tr>
<td>$\rightarrow \bullet Noun \cdot NOMINAL$</td>
</tr>
</tbody>
</table>