Tagging: An Overview
Rule-based Disambiguation

• Example after-morphology data (using Penn tagset):

```
I     watch       a       fly  .
NN    NN          DT      NN    .
PRP   VB          NN      VB
     VBP          VBP
```

• Rules using
  – word forms, from context & current position
  – tags, from context and current position
  – tag sets, from context and current position
  – combinations thereof
Example Rules

• If-then style:
  • $D_{eq,-1,Tag} \Rightarrow NN$
    (implies $NN_{in,0,Set}$ as a condition)
  • $PRP_{eq,-1,Tag}$ and $DT_{eq,+1,Tag} \Rightarrow VBP$
  • $\{DT,NN\}_{sub,0,Set} \Rightarrow DT$
  • $\{VB,VBZ,VBP,VBD,VBG\}_{inc,+1,Tag} \Rightarrow not \ DT$

• Regular expressions:
  • $not(<*,*,DT> <*,*,notNN>)$
  • $not(<*,*,PRP>,<*,*,notVBP>,<*,*,DT>)$
  • $not(<*,\{DT,NN\}_{sub},notDT>)$
  • $not(<*,*,DT>,<*,*,\{VB,VBZ,VBP,VBD,VBG\}>)$
Implementation

• Finite State Automata
  – parallel (each rule ~ automaton);
    • algorithm: keep all paths which cause all automata say yes
  – compile into single FSA (intersection)

• Algorithm:
  – a version of Viterbi search, but:
    • no probabilities (“categorical” rules)
    • multiple input:
      – keep track of all possible paths
Example: the FSA

- \textit{R1}: not(\texttt{<*,*,DT> <*,*,notNN>})
- \textit{R2}: not(\texttt{<*,*,PRP>,<*,*,notVBP>,<*,*,DT>})
- \textit{R3}: not(\texttt{<*,\{DT,NN\} sub,DT>})
- \textit{R4}: not(\texttt{<*,*,DT>,<*,*,\{VB,VBZ,VBP,VBD,VBG\}>})

- \textit{R1}:
  \begin{itemize}
    
  \end{itemize}

- \textit{R3}:
  \begin{itemize}
    
  \end{itemize}
Applying the FSA

- **R1**: \( \text{not}(<*,*,\text{DT}>,<*,*,\text{notNN}>) \)
- **R2**: \( \text{not}(<*,*,\text{PRP}>,<*,*,\text{notVBP}>,<*,*,\text{DT}>) \)
- **R3**: \( \text{not}(<*,\{\text{DT,NN}\}_{\text{sub}},\text{DT}>) \)
- **R4**: \( \text{not}(<*,*,\text{DT}>,<*,*,\{\text{VB,VBZ,VBP,VBD,VBG}\}>) \)

- **R1 blocks**: remains: or
- **R2 blocks**: remains e.g.: and more
- **R3 blocks**: remains only:
- **R4 \( \subseteq \) R1!
Applying the FSA (Cont.)

• Combine:

• Result:

I watch a fly.
Tagging by Parsing

- Build a parse tree from the multiple input:

```
S
/   \
|    |
| VP  |
|  /  |
| NP  |
|     |
| I   |
| watch |
|      |
|      |
| NN   |
| NN   |
| PRP  |
| VB   |
| VBP  |
|      |
| a    |
|      |
| DT   |
| NN   |
|      |
| fly  |
|      |
| NN   |
|      |
| VBP  |
```

- Track down rules: e.g., NP → DT NN: extract (a/DT fly/NN)
- More difficult than tagging itself; results mixed
Statistical Methods (Overview)

• “Probabilistic”:
  • HMM
    – Merialdo and many more (XLT)
  • Maximum Entropy
    – DellaPietra et al., Ratnaparkhi, and others

• Rule-based:
  • TBEDL (Transformation Based, Error Driven Learning)
    – Brill’s tagger
  • Example-based
    – Daelemans, Zavrel, others

• Feature-based (inflective languages)

• Classifier Combination (Brill’s ideas)