CS 378 Lecture 14
Shift-reduce parsing, review

Announcements

MIDTERM TUESDAY in class A3 due today

Recap Dependencies

The cat jumped and landed on the desk

- Verbs are heads of clauses/sentences
- Verbs have nouns/prepositions as children
- Nouns are modified by stuff
Today  Shift-reduce parsing

State-of-the-art parsers (in video on website)

Move through a sentence and make decisions greedily

Input: sentence
Output: dep parse

Stack: partial parse trees
Buffer: remaining words of the sentence
Root: I ate some spaghetti bolognese

Initial state:
stack: [ROOT]
Buffer: [I ate some spaghetti bolognese]

Operations: (arc-standard)

1. Shift: first word of buffer → end of stack

2. Left-arc: take top two words from stack,

3. Right-arc: add an arc, add back to the stack
What is the correct (oracle) sequence of operations to build this tree?

\[ S \ [\text{ROOT}] \ \beta \ [\text{I ate some spa bo}] \]

1. Shift

\[ S \ [\text{ROOT I}] \ [\text{ate some spa bo}] \]

2. Shift

\[ S \ [\text{ROOT I ate}] \ [\text{some spa bo}] \]

3. Left-arc

\[ S \ [\text{ROOT ate}] \ [\text{some spa bo}] \]

\[ i \]
4 5 Shifts

S [ROOT ate some spa] [bo]

↓  I

6 Left-arc

S [ROOT ate spa] [bo]

↓  I  ↓  some

Can't Right-arc yet because spaghetti isn't "finished"

7 Shift

S [ROOT ate spa bo]

↓  I  ↓  some
8) Right-arc

\[
\begin{array}{c}
\text{[ROOT ate spa]} \\
\downarrow \\
\text{I} \\
\downarrow \\
\text{some bo}
\end{array}
\]

9) Right-arc

\[
\begin{array}{c}
\text{[ROOT ate]} \\
\downarrow \\
\text{I} \\
\downarrow \\
\text{spe} \\
\downarrow \\
\text{some bo}
\end{array}
\]

10) R-A
Property: Arc-standard can build any projective tree

(non-projective)

Building shift-reduce parsers

Parser is a classifier

Maps from (stack, buffer) → \{ S, LA, RA \}

Features: \( f(S, B) \)
Feats are very complex

\[ S[\text{Root ate spa }] [b_0] \]

\[ \downarrow \quad \downarrow \quad \text{some} \]

R-A? How do we know it's wrong?

Still need spa \[ b_0 \]

feature \((S[-1], B[0])\)

feature \((S[-2], S[-1])\)
Look at a lot of signals:
- first few of buffer
- last few of stack
- children of stack \([I, \text{some}]\)

Lots of indicator features
(In tagging: \(\text{PrevWord} = X\))

(a) \(\text{Stack Last} = X\)
(b) \(\text{Stack Last} = X \& \text{Buf First} = Y\)
(c) \(\text{Stack Last Tag} = N \& \text{Buf First} = \text{Adj}\)

many types of features
CKY again

Fall '21 / HMMs

Sp 20 Q6

K=1 only optimal loop over rules:
  - binary rule

DT N

NP \rightarrow DT N

the ring

0 1 2
the ring and rings

NP: -2  NP: -2  
DT  NN  DT  NN  VBD  VBD
the  rat  the  cat  chased  ate

k=3  ×
k=4

NP  VBD
S → NP  VBD
\( W_n = (0, 0) \) \hspace{1cm} \( W_{sc} = (0, 0) \) \hspace{1cm} \( W_{sp} = (0, 2) \) \hspace{1cm} \( W_{po} = (2, 2) \)

\( (0, 0) \sim \text{health} \hspace{1cm} y_{pred} = \text{sports} \)

new weights: \( (10 \ 00 \ 12 \ 22) \)

\( (0, 1) \sim \text{science} \hspace{1cm} y_{pred} = \text{sports} \)

new weights: \( (10 \ 01 \ 11 \ 22) \)