LING/C SC/PSYC 438/538

Lecture 10

Sandiway Fong
Today's Topics

• A note on the UIUC POS Tagger
• Fun with POS Tagging
• Perl regex wrap-up
UIUC Tagger

- **Mystery**: I mentioned last time that (when I copied the output of the tagger) the forward slashes disappeared.
- Your classmate Hongyi Zhu not only did some excellent sleuthing but also supplied a script to work around the problem.
Hongyi Zhu:

- I looked at the UIUC POS tagging website and found that the displayed "/" was created by CSS pseudo element, which is not a part of the text. I wrote a short javascript to generate and copy the text with [forward] slash.

```javascript
st=document.styleSheets;for(var i=0;i<st.length;i++){if(st[i].href!=null&&st[i].href.indexOf("POS.css")>-1){rs=st[i].cssRules;var t=0;for(var j=0;j<rs.length;j++){if(rs[j].cssText.indexOf("::after")>-1){t=j;}};if(t!=0){st[i].deleteRule(t);}}};lb=document.getElementsByClassName("label");for(var i=0;i<lb.length;i++){lb[i].innerHTML+="/";};window.prompt("Press Ctrl-C or Right-click and Copy",document.getElementsByClassName("output")[0].innerText.replace(// /g,"/"));
```

- After you clicked 'submit', you can call a browser console (F12 in Chrome or Option-Command-C in Safari) and then copy and execute the script. To make it easy to copy and execute, I removed all the white spaces and line breaks. You should be able to copy the result with [forward] slashes from a prompt.
POS Tagger

• We'll return to the topic of POS tagging later in the course...

• POS tagging is not 100% accurate
POS Tagger

• Let's digress a bit and look at a worst case scenario. Consider the Buffalo sentence from Homework 4 ...

Buffalo buffalo buffalo buffalo buffalo buffalo buffalo buffalo.

Definition of BUFFALO

City & port on W New York, on Lake Erie & Niagara River, pop 281,310

buffalo

Verb

Substantive meaning: Substituting the synonymous "bison" for "buffalo" (animal), "bully" for "buffalo" (verb), and leaving "buffalo" to mean the city yields:

- Buffalo Bison, whom other buffalo bison bully, threaten or bully buffalo bison.
POS Tagger

• Buffalo buffalo = *buffalo from Buffalo*
• Since *buffalo* is a transitive verb, we can form:
  – NNP/Buffalo NNS/buffalo VBP/buffalo NNP/Buffalo NNS/buffalo
POS Tagger

- Object relative clause construction:
  - Buffalo buffalo (that) Buffalo buffalo buffalo
  - NNP/Buffalo NNS/buffalo NNP/Buffalo NNS/buffalo VBP/buffalo
POS Tagger

• Substitute the relative clause into the sentence:
  • NNP/Buffalo NNS/buffalo NNP/Buffalo NNS/buffalo VBP/buffalo VBP/buffalo NNP/Buffalo NNS/buffalo

Syntactic Analysis
POS Tagger

• The UIUC tagger:

Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo.

The Part-of-Speech tagger has automatically labeled the input in the following way.

NNP/ Buffalo NN/ buffalo NNP/ Buffalo NN/ buffalo NN/ buffalo NN/ buffalo NNP/ Buffalo NN/ buffalo . /
POS Tagger

• Stanford Parser:

Your query

Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo buffalo.

Tagging

Buffalo/NNP buffalo/NNP Buffalo/NNP buffalo/JJ buffalo/NN buffalo/NN Buffalo/NNP buffalo/VBZ ./.

Parse

(ROOT
  (S
    (NP (NNP Buffalo) (NNP buffalo) (NNP Buffalo) (JJ buffalo) (NN buffalo) (NN buffalo) (NN buffalo) (NNP Buffalo))
    (VP (VBZ buffalo))
    (.
     .)))

NNP = proper noun,
JJ = adjective,
VBZ = verb 3rd person singular present
POS Tagger

• Berkeley Parser:

```
Buffalo buffalo Buffalo buffalo buffalo buffalo buffalo Buffalo buffalo.
```

VBP = verb
non-3rd person singular present

Parse!
Arrays as Stacks and Queues

- **Arrays:**
  - insertion and deletion from the ends

Perl functions may have **side effects** and also return values.

- **shift ARRAY**
- **shift**
  Shifts the first value of the array off and returns it, shortening the array by 1 and moving everything down.

- **unshift ARRAY, LIST**
  Does the opposite of a **shift**. Or the opposite of a **push**, depending on how you look at it. Prepends list to the front of the array, and returns the new number of elements in the array.

- **push ARRAY, LIST**
  Treats ARRAY as a stack, and pushes the values of LIST onto the end of ARRAY.

- **pop**
  Pops and returns the last value of the array, shortening the array by one element.
Arrays as Stacks and Queues

• Example:

```perl
@ch = split //, $ARGV[0];
push @ch, '$';
unshift @ch, '^';
print '@ch\n'
```

$perl pushunshift.perl this
^ t h i s $

• Another example:

```perl
@ch = split //, $ARGV[0];
pop @ch;
shift @ch;
print '@ch\n'
```

$perl popshift.perl "this"
th i s

Generalized form:

- splice ARRAY,OFFSET,LENGTH,LIST
Regex Recursion

- Pallindrome = something that reads the same backwards or forwards, e.g. *kayak* and *racecar*.
- Regexp libraries cannot express pallindromes but *Perl* regexp libraries can because we can use backreferences recursively.
Regex Recursion

• Program:

```perl
$word = $ARGV[0];
print "Word: $word is ";
if ($word !~ /^\w?|\w(?1)\2$/) {
    print "not 
}
print "a pallindrome\n"
```

```perl
$perl pallindrome.perl kaba
Word: kaba is not a pallindrome
$perl pallindrome.perl kabak
Word: kabak is a pallindrome
$perl pallindrome.perl kaak
Word: kaak is a pallindrome
$perl pallindrome.perl ka
Word: ka is not a pallindrome
$perl pallindrome.perl k
Word: k is a pallindrome
$perl pallindrome.perl kak
Word: kak is a pallindrome
$perl pallindrome.perl
Word: is a pallindrome
```
Regex Lookahead and Lookback

- **Zero-width regexs:**
  - `^` (start of string)
  - `$` (end of string)
  - `\b` (word boundary)
    - matches the imaginary position between \w\W (or \W\w, or just before beginning of string if ^\w, just after the end of the string if \w$)

- **Current position of match (so far) doesn't change!**
  - `(?=regex)` (lookahead from current position)
  - `(<=regex)` (lookback from current position)
  - `(!regex)` (negative lookahead)
  - `(!<regex)` (negative lookback)
Regex Lookahead and Lookback

• Example:

```perl
$s = "_bison _cat snake _dog cat _snake dog";
while ($s =~ /_(\w+)\b(?:=.\*\1\b)/g) {
    print "<$1>\n"
}
```

Looks for a word beginning with _ such that there is a duplicate ahead without the _

• Note: lookback cannot be variable length in Perl
Debugging Perl regex

- `(?{ Perl code })` can be inserted anywhere in a regex
- can assist with debugging
- **Example:**

```perl
$s = "_bison _cat snake _dog cat _snake dog";
while ($s =~ /(_(\w+))\b(?{print "$1\n"})(?=.*\1\b)/g) {
    print "<$1>\n"
}
```