Administrivia

• Homework 5 graded
Today's Topics

- Homework 5 Review
- We'll begin looking at formal language theory
On Perl ...

Webcomic: xkcd.com

Acknowledgement: Erwin Chan
Homework 5 Review

Demo

Part of Speech Tagging Demo

737,457 views

If you wish to cite this work, please cite this publication.

Helicopters will patrol the temporary no-fly zone around New Jersey's MetLife Stadium Sunday, with F-16s based in Atlantic City ready to be scrambled if an unauthorized aircraft does enter the restricted airspace.

Down below, bomb-sniffing dogs will patrol the trains and buses that are expected to take approximately 30,000 of the 80,000-plus spectators to Sunday's Super Bowl between the Denver Broncos and Seattle Seahawks.

The Transportation Security Administration said it has added about two dozen dogs to monitor passengers arriving at Philadelphia's Convention Center on Friday.
Homework 5 Review

- Example (from the UIUC demo) (corrected):
  - Helicopters] will patrol [the temporary no-fly zone] around [New Jersey's MetLife Stadium] [Sunday], with [F-16s] based in [Atlantic City] ready to be scrambled if [an unauthorized aircraft] does enter [the restricted airspace].
  - Down below, [bomb-sniffing dogs] will patrol [the trains] and [buses] that are expected to take approximately [30,000] of [the 80,000-plus spectators] to [Sunday's Super Bowl] between [the Denver Broncos] and [Seattle Seahawks].
  - [The Transportation Security Administration] said [it] has added about [two dozen dogs] to monitor [passengers] coming in and out of [the airport] around [the Super Bowl].
  - On [Saturday], [TSA agents] demonstrated how [the dogs] can sniff out [many different types] of [explosives]. Once [they] do, [they]'re trained to sit rather than attack, so as not to raise [suspicion] or create [a panic].
  - [TSA spokeswoman Lisa Farbstein] said [the dogs] undergo [12 weeks] of [training], which costs about [200,000], factoring in [food], [vehicles] and [salaries] for [trainers].
  - [Dogs] have been used in [cargo areas] for [some time], but have just been introduced recently in [passenger areas] at [Newark] and [JFK airports]. [JFK] has [one dog] and [Newark] has [a handful], [Farbstein] said.
Homework 5 Review

• Example:
  1. NNPS/Helicopters
  2. NN/patrol
  3. DT/the JJ/temporary JJ/no-fly NN/zone
  4. NNP/New NNP/Jersey POS/'s NNP/MetLife NNP/Stadium NNP/Sunday
  5. NNP/F-16s
  6. NNP/Atlantic NNP/City
  7. DT/an JJ/unauthorized NN/aircraft
  8. DT/the VBN/restricted NN/airspace

  [Helicopters] will patrol [the temporary no-fly zone] around [New Jersey's MetLife Stadium] [Sunday], with [F-16s] based in [Atlantic City] ready to be scrambled if [an unauthorized aircraft] does enter [the restricted airspace].
Homework 5 Review

• Example:
  1. JJ/bomb-sniffing NNS/dogs
  2. NN/patrol
  3. DT/the NNS/trains
  4. NNS/buses
  5. DT/the JJ/80,000-plus NNS/spectators
  6. NNP/Sunday/POS/'s NNP/Super NNP/Bowl
  7. DT/the NNP/Denver NNS/Broncos
  8. NNP/Seattle NNP/Seahawks

  – Down below, [bomb-sniffing dogs] will patrol [the trains] and [buses] that are expected to take approximately [30,000] of [the 80,000-plus spectators] to [Sunday's Super Bowl] between [the Denver Broncos and Seattle Seahawks].
Homework 5 Review

• Example:
  1. DT/The NNP/Transportation NNP/Security NNP/Administration
  2. PRP/it
  3. CD/two NN/dozen NNS/dogs
  4. NNS/passengers
  5. DT/the NN/airport
  6. DT/the NNP/Super NNP/Bowl
     – [The Transportation Security Administration] said [it] has added about [two dozen dogs] to monitor [passengers] coming in and out of [the airport] around [the Super Bowl].
Homework 5 Review

• Example:
1. NNP/Saturday
2. NNP/TSA NNS/agents
3. DT/the NNS/dogs
4. JJ/many JJ/different NNS/types
5. NNS/explosives
6. PRP/they
7. PRP/they
8. NN/attack
9. NN/suspicion
10. DT/a NN/panic

   - On [Saturday], [TSA agents] demonstrated how [the dogs] can sniff out [many different types] of [explosives]. Once [they] do, [they]'re trained to sit rather than attack, so as not to raise [suspicion] or create [a panic].
Homework 5 Review

• Example:
  1. NNP/TSA NN/spokeswoman NNP/Lisa NNP/Farbstein
  2. DT/the NNS/dogs
  3. CD/12 NNS/weeks
  4. NN/training
  5. NN/$200,000
  6. **NN/factoring**
  7. NN/food
  8. NNS/vehicles
  9. NNS/salaries
  10. NNS/trainers

    - [TSA spokeswoman Lisa Farbstein] said [the dogs] undergo [12 weeks] of [training], which costs about [$200,000], factoring in [food], [vehicles] and [salaries] for [trainers].
Homework 5 Review

• Example:
  1. NNS/Dogs
  2. NN/cargo NNS/areas
  3. DT/some NN/time
  4. NN/passenger NNS/areas
  5. NNP/Newark
  6. NNP/JFK NNS/airports
  7. NNP/JFK
  8. CD/one NN/dog
  9. NNP/Newark
  10. DT/a NN/handful
  11. NNP/Farbstein

  [Dogs] have been used in [cargo areas] for [some time], but have just been introduced recently in [passenger areas] at [Newark] and [JFK airports]. [JFK] has [one dog] and [Newark] has [a handful], [Farbstein] said.
Complex NPs (CNPs) can be formed out of "basic"-NPs using prepositions and conjunctions as connectives.

Examples:

- Helicopters will patrol \([\text{CNP the temporary no-fly zone around New Jersey's MetLife Stadium}]\)
- \([\text{CNP the [Denver Broncos and Seattle Seahawks]}]\)
- \([\text{CNP Sunday's Super Bowl between [CNP the Denver Broncos and Seattle Seahawks]}]\)
Homework 5 Review

• Modified code:
  – (1) recursion or (2) iteration (shown here)

```perl
open($fh, $ARGV[0]) or die "$ARGV[0] not found!

$np = qr/(DT/S+ )?((CD|JJ|VBN)/S+ )*((NNS|NNPS|POS)/S+ )*((NNS|NNPS|POS|PRP)/S+)=/;  
while ($line = <$fh>) {
  $pos_line = "";
  while ($line =~ m!$np( (IN|CC)/S+ $np)*!g) {
    print "$\n"
  }
  print "Line: $line"
}
```
Homework 5 Review

• Example:

• DT/the JJ/temporary JJ/no-fly NN/zone IN/around NNP/New NNP/Jersey POS/'s NNP/MetLife NNP/Stadium NNP/Sunday

  – [Helicopters] will patrol [the temporary no-fly zone] around [New Jersey's MetLife Stadium] [Sunday], with [F-16s] based in [Atlantic City] ready to be scrambled if [an unauthorized aircraft] does enter [the restricted airspace].
Homework 5 Review

• **Example:**

1. DT/the NNS/trains CC/and NNS/buses
2. NNP/Sunday/POS/'s NNP/Super NNP/Bowl IN/between DT/the NNP/Denver NNS/Broncos CC/and NNP/Seattle NNP/Seahawks

   – Down below, [**bomb-sniffing dogs**] will patrol [**the trains**] and [**buses**] that are expected to take approximately [30,000] of [**the 80,000-plus spectators**] to [**Sunday's Super Bowl**] between [**the Denver Broncos**] and [**Seattle Seahawks**].
Homework 5 Review

• Example:
• DT/the NN/airport IN/around DT/the NNP/Super NNP/Bowl
  – [The Transportation Security Administration] said it has added about [two dozen dogs] to monitor [passengers] coming in and out of [the airport] around [the Super Bowl].
Example:

JJ/many JJ/different NNS/types IN/of NNS/explosives

On [Saturday], [TSA agents] demonstrated how [the dogs] can sniff out [many different types] of [explosives]. Once [they] do, [they]'re trained to sit rather than attack, so as not to raise [suspicion] or create [a panic].
Homework 5 Review

• Example:
  1. CD/12 NNS/weeks IN/of NN/training
  2. NN/factoring IN/in NN/food
  3. NNS/vehicles CC/and NNS/salaries IN/for NNS/trainers
     – [TSA spokeswoman Lisa Farbstein] said [the dogs] undergo [12 weeks] of [training], which costs about [$200,000], factoring in [food], [vehicles] and [salaries] for [trainers].
Homework 5 Review

- **Example:**
  1. NN/cargo NNS/areas IN/for DT/some NN/time
  2. NN/passenger NNS/areas IN/at NNP/Newark CC/and NNP/JFK NNS/airports
  3. CD/one NN/dog CC/and NNP/Newark
     - [Dogs] have been used in [cargo areas] for [some time], but have just been introduced recently in [passenger areas] at [Newark] and [JFK airports]. [JFK] has [one dog] and [Newark] has [a handful], [Farbstein] said.
More complex still...

• Recursive nature of natural languages:
  – Complex NPs can contain *sentences* that contain NPs and so on...

• Example (restrictive relative clause attached):
  – F-16s based in Atlantic City
  – F-16s *that are* based in Atlantic City

• Example (non-restrictive):
  – 12 weeks of training, which costs about $200,000
Beyond POS tagging

• How does a parser handle these examples?
  – Syntactic analysis should be able to help resolve and disambiguate some complex NPs
Berkeley Parser

- Helicopters will patrol the temporary no-fly zone around New Jersey's MetLife Stadium Sunday, with F-16s based in Atlantic City ready to be scrambled if an unauthorized aircraft does enter the restricted airspace.
Down below, bomb-sniffing dogs will patrol the trains and buses that are expected to take approximately 30,000 of the 80,000-plus spectators to Sunday's Super Bowl between the Denver Broncos and Seattle Seahawks.
The Transportation Security Administration said it has added about two dozen dogs to monitor passengers coming in and out of the airport around the Super Bowl.
• On Saturday, TSA agents demonstrated how the dogs can sniff out many different types of explosives.
Berkeley Parser

- Once they do, they're trained to sit rather than attack, so as not to raise suspicion or create a panic.
• TSA spokeswoman Lisa Farbstein said the dogs undergo 12 weeks of training, which costs about $200,000, factoring in food, vehicles and salaries for trainers.
Berkeley Parser

- Dogs have been used in cargo areas for some time, but have just been introduced recently in passenger areas at Newark and JFK airports. JFK has one dog and Newark has a handful, Farbstein said.
Regular Languages

• Three formalisms
  – All formally equivalent (no difference in expressive power)
  – i.e. if you can encode it using a RE, you can do it using a FSA or regular grammar, and so on ...

Note: Perl regexs are more powerful than the math characterization, e.g. backreferences Prime number testing...

Talk about formal equivalence next time
Regular Languages

• A regular language
  • is the set of strings
  • (including possibly the empty string)
  • (set itself could also be empty)
  • (set can be infinite)
  • generated by a RE/FSA/Regular Grammar
Regular Languages

• Example:
  – Language: \( L = \{ a^+b^+ \} \)
    “one or more a’s followed by one or more b’s”
  
    **L is a regular language**
  – described by a regular expression *(we’ll define it formally next time)*

  – Note:
    • infinite set of strings belonging to language \( L \)
      – e.g. abbb, aaaab, aabb, *abab, *\( \lambda \)

  – Notation:
    • \( \lambda \) is the empty string (or string with zero length),
      sometimes \( \varepsilon \) is used instead
    • * means string is not in the language
Finite State Automata (FSA)

• $L = \{ a^+ b^+ \}$ can be also be generated by the following FSA

> Indicates start state
Red circle indicates end (accepting) state
we accept a input string only when we’re in an end state and we’re at the end of the string
Finite State Automata (FSA)

- \( L = \{ a^+ b^+ \} \) can be also be generated by the following FSA

There is a natural correspondence between components of the FSA and the regex defining \( L \)

Note:
- \( L = \{ a^+ b^+ \} \)
- \( L = \{ aa^* bb^* \} \)
Finite State Automata (FSA)

- $L = \{ a^+ b^+ \}$ can be also be generated by the following FSA

**deterministic FSA (DFSA)**

no ambiguity about where to go at any given state
i.e. for each input symbol in the alphabet at any given state, there is a unique “action” to take

**non-deterministic FSA (NDFSA)**

no restriction on ambiguity (surprisingly, no increase in power)
Finite State Automata (FSA)

• more formally
  – \((Q,s,f,\Sigma,\delta)\)

1. set of states \((Q)\): \{s,x,y\} \textit{must be a finite set}
2. start state \((s)\): s
3. end state(s) \((f)\): y
4. alphabet \((\Sigma)\): \{a, b\}
5. transition function \(\delta\):

  \textit{signature}: character \times \text{state} \rightarrow \text{state}

  - \(\delta(a,s)=x\)
  - \(\delta(a,x)=x\)
  - \(\delta(b,x)=y\)
  - \(\delta(b,y)=y\)
Finite State Automata (FSA)

- **In Perl**
  
  transition function $\delta$:
  
  - $\delta(a,s)=x$
  - $\delta(a,x)=x$
  - $\delta(b,x)=y$
  - $\delta(b,y)=y$

  We can simulate our 2D transition table using a hash whose elements are themselves (anonymized) hashes:

  ```perl
  %transitiontable = {
    s => {
      a => "x",
    },
    x => {
      a => "x",
      b => "y",
    },
    y => {
      b => "y",
    }
  };
  ```

  **Example:**
  
  ```perl
  print "$transitiontable{s}{a}\n";
  ```

  [Syntactic sugar for %transitiontable = ( "s", {"a", "x"},, "x", {"a", "x", "b", "y"}, "y", {"b", "y"}, );]
Finite State Automata (FSA)

- Given transition table encoded as a hash
- How to build a decider (Accept/Reject) in Perl?

Complications:
- How about $\epsilon$-transitions?
- Multiple end states?
- Multiple start states?
- Non-deterministic FSA?
Finite State Automata (FSA)

%transitiontable = (
    s => {
        a  => "x",
    },
    x => {
        a  => "x",
        b  => "y",
    },
    y => {
        b  => "y",
    }
);

@input = @ARGV;

$state = "s";

foreach $c (@input) {
    $state = $transitiontable{$state}{$c};
}

if ($state eq "y") {
    print "Accept\n";
} else {
    print "Reject\n";
}
Finite State Automata (FSA)

```plaintext
function D-RECOGNIZE(tape, machine) returns accept or reject

    index ← Beginning of tape
    current-state ← Initial state of machine
    loop
        if End of input has been reached then
            if current-state is an accept state then
                return accept
            else
                return reject
        elseif transition-table[current-state, tape[index]] is empty then
            return reject
        else
            current-state ← transition-table[current-state, tape[index]]
            index ← index + 1
        end

Figure 2.12 An algorithm for deterministic recognition of FSAs. This algorithm returns accept if the entire string it is pointing at is in the language defined by the FSA, and reject if the string is not in the language.
```

this is pseudo-code
not any real programming language
Finite State Automata (FSA)

• **practical applications**
  - *can be encoded and run efficiently on a computer*
  - *widely used*

  – encode regular expressions (e.g. Perl regex)
  – morphological analyzers
    - Different word forms, e.g. want, wanted, unwanted (suffixation/prefixation)
    - *see chapter 3 of textbook*

• **speech recognizers**
  - Markov models
    - = FSA + probabilities

• *and much more ...*