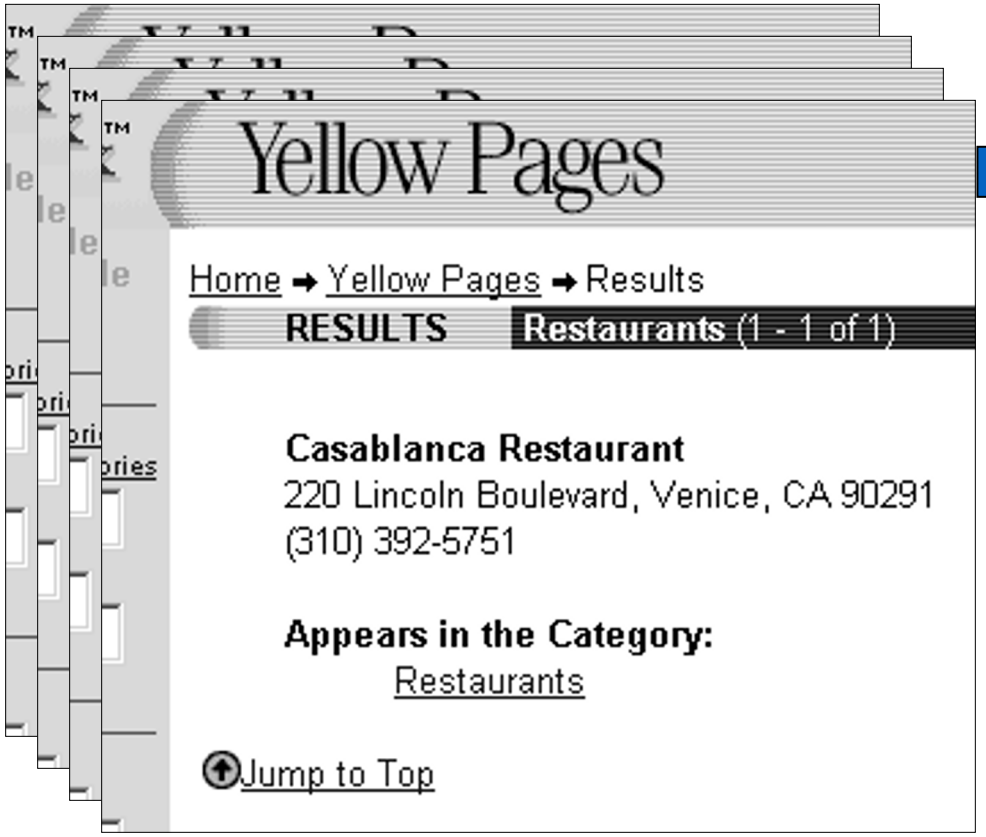
Mayank Kejriwal
University of Southern California

Wrappers

Extracting Data from Semi-structured Sources



NAME	Casablanca Restaurant
STREET	220 Lincoln Boulevard
CITY	Venice
PHONE	(310) 392-5751

Definition of wrapper

- (from your text) A tuple ($\mathbf{T_W}$, $\mathbf{E_W}$) where $\mathbf{T_W}$ is a target schema, $\mathbf{E_W}$ is an extraction program that uses the format $\mathbf{F_s}$ to extract from each page a data instance conforming to $\mathbf{T_W}$

Four types of wrappers

- Manual
- Learning
- Automatic
- Interactive

Manual Wrapper Construction

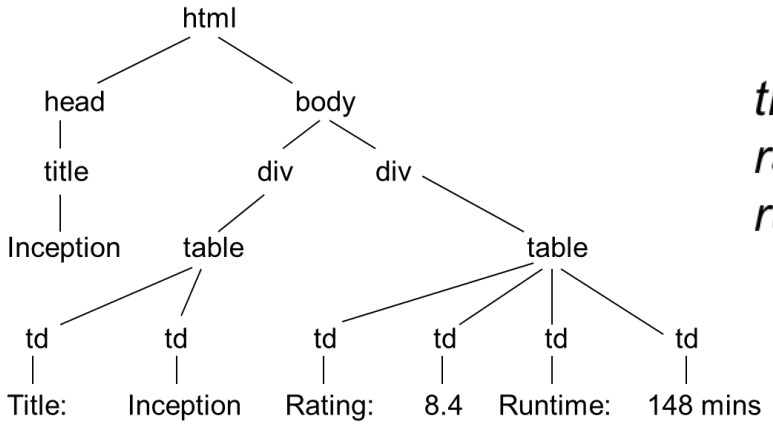
- Developer examines a set of Web pages
  - manually creates target schema $\mathbf{T}_W$ and extraction program $\mathbf{E}_W$
  - often writes $\mathbf{E}_W$ using a procedural language such as Perl


```
#!/usr/bin/perl -w

open(INFILE, $ARGV[0]) or die "can't open file\n";
while ($line = <INFILE>) {
  if ($line =~ m/<B\>(.+?)\<\/B\>\s+?\<I\>(\d+?)\<\/I\>\<BR\>/) {
     print "($1,$2)\n";
  }
}
close(INFILE);
```

Manual Wrapper Construction

- There are multiple ways to view a page
  - as a string ➔ can write wrapper as Perl program
  - as a DOM tree ➔ can write wrapper using XPath language



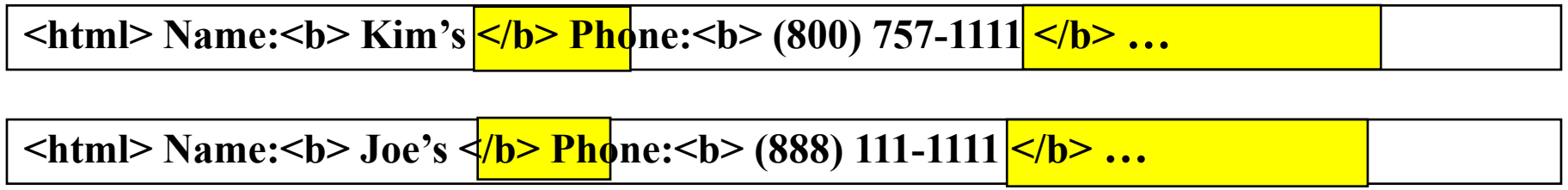
title = /html/body/div[1]/table/td[2]/text()
rating = /html/body/div[2]/table/td[2]/text()
runtime = /html/body/div[2]/table/td[4]/text()

  - as a visual page, consisting of blocks

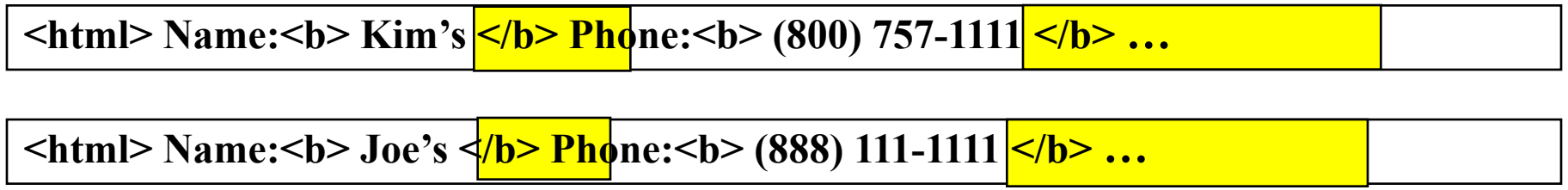
Rule Learning

- Machine learning:
  - Goal: Find a instance of the given wrapper type that covers the given examples
  - INPUT:
    - Labeled examples: training & testing data
    - Admissible rules (hypotheses space)
    - Search strategy
  - Desired output:
    - Rule that performs well both on training and testing data
  - Termination
    - Train on sufficient data to be probably approximately correct (PAC)

Learning LR extraction rules

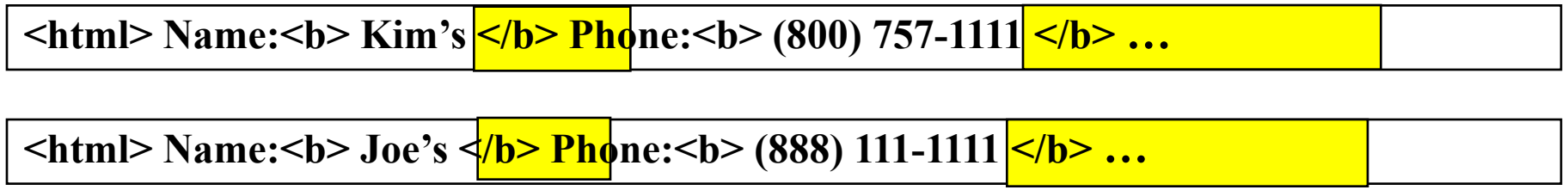


Learning LR extraction rules

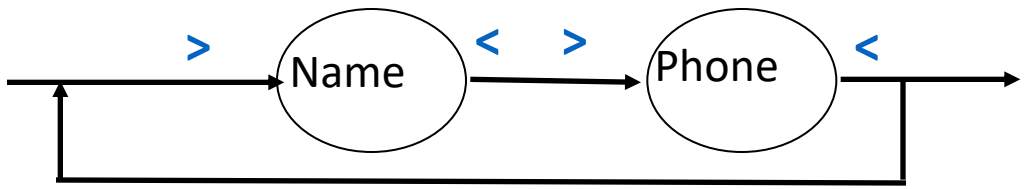


- Admissible rules:
  - prefixes & suffixes of items of interest
- Search strategy:
  - start with shortest prefix & suffix, and expand until correct

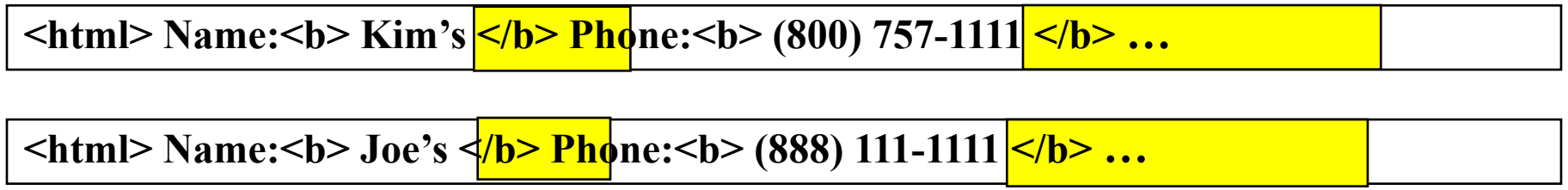
Learning LR extraction rules



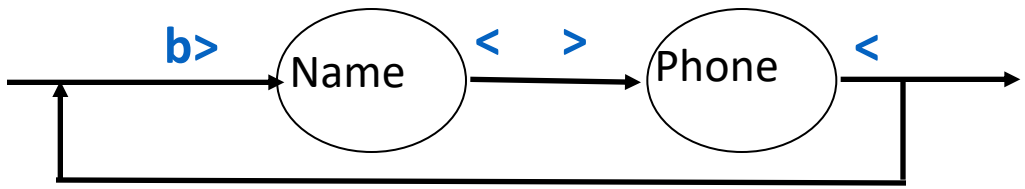
- Admissible rules:
  - prefixes & suffixes of items of interest
- Search strategy:
  - start with shortest prefix & suffix, and expand until correct



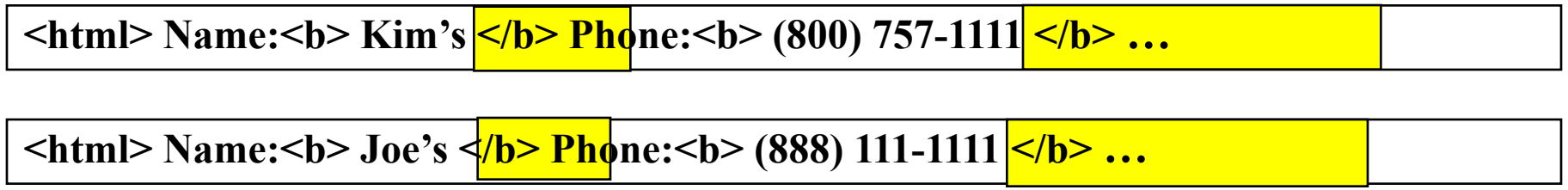
Learning LR extraction rules



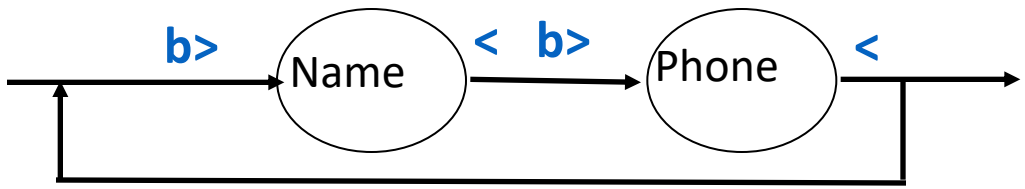
- Admissible rules:
  - prefixes & suffixes of items of interest
- Search strategy:
  - start with shortest prefix & suffix, and expand until correct



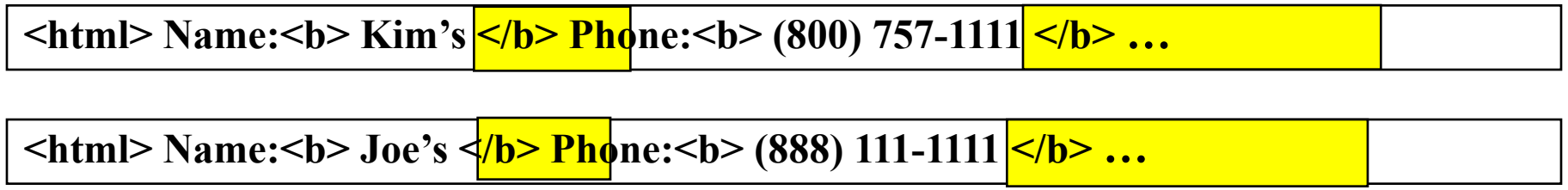
Learning LR extraction rules



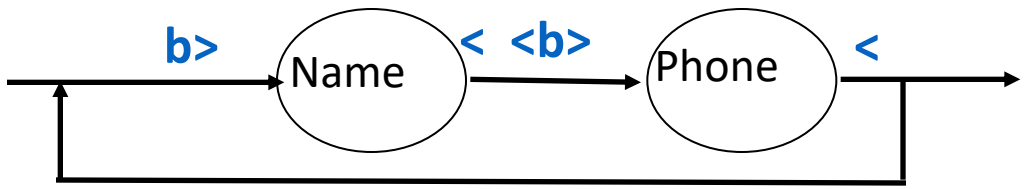
- Admissible rules:
  - prefixes & suffixes of items of interest
- Search strategy:
  - start with shortest prefix & suffix, and expand until correct



Learning LR extraction rules



- Admissible rules:
  - prefixes & suffixes of items of interest
- Search strategy:
  - start with shortest prefix & suffix, and expand until correct



Labeling Data

- Instead of labeling all of the data, use *recognizers* to find instances of a particular attribute
- Recognizers may be:
  - Perfect
    - Accept all positive instances and reject all negatives
  - Incomplete
    - Reject all negative instances but reject some positives
  - Unsound
    - Accept all positive, but accept some negatives
  - Unreliable
    - Reject some positive instances and accept some negatives

Summary

- Advantages:
  - Fast to learn & extract
  - Some sources could be labeled automatically given an appropriate set of recognizers
- Drawbacks:
  - Cannot handle permutations and missing items
  - Entire page must be labeled
  - Requires large number of examples