

#21, Applying Pattern Mining to Web Information Extraction

PS Web Information Extraction, SS 2005

by Jeremy Solarz

Approaches

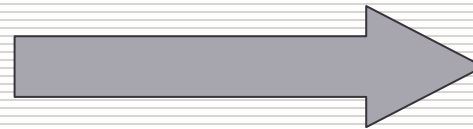
- Use training data to produce extraction rules.
- Pattern mining - no human intervention

Example (1)

```
<html>
  <head></head>
  <body>
    <table>
      <tr>
        <td>Text1</td>
      </tr>
      <tr>
        <td>Text2</td>
      </tr>
    </table>
  </body>
```

Example (2)

```
<html>
  <head></head>
  <body>
    <table>
      <tr>
        <td>Text1</td>
      </tr>
      <tr>
        <td>Text2</td>
      </tr>
    </table>
  </body>
```

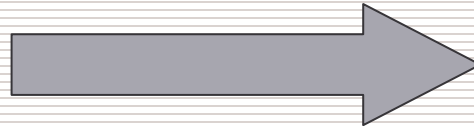


replace text with Text(_)

```
<html>
  <head></head>
  <body>
    <table>
      <tr>
        <td>Text (_)</td>
      </tr>
      <tr>
        <td>Text (_)</td>
      </tr>
    </table>
  </body>
```

Example (3)

```
<html>
  <head></head>
  <body>
  <table>
    <tr>
      <td>Text (_)</td>
    </tr>
    <tr>
      <td>Text (_)</td>
    </tr>
  </table>
</body>
```



replace HTML with Html(<tag>)

```
Html (<html>)
  Html (<head>) Html (</head>)
  Html (<body>)
  Html (<table>)
    Html (<tr>)
      Html (<td>) Text (_) Html (</td>)
    Html (</tr>)
  Html (<tr>)
    Html (<td>) Text (_) Html (</td>)
  Html (</tr>)
  Html (</table>)
  Html (</body>)
```

Example (4)

```
Html (<html>
  Html (<head> Html (</head>
  Html (<body>
  Html (<table>
```

```
    Html (<tr>
      Html (<td> Text (_) Html (</td>
    Html (</tr>
    Html (<tr>
      Html (<td> Text (_) Html (</td>
    Html (</tr>
```

```
  Html (</table>
  Html (</body>
```

1

2

Repeated Patterns

for instance:

```
„Html (<td>) Text (_) “  
“Html (<td>) Text (_) Html (</td>) “  
„Html (<tr>) Html (<td>) Text (_) Html (</td>) “
```

```
Html (<html>)  
  Html (<head>) Html (</head>)  
  Html (<body>)  
    Html (<table>)  
      Html (<tr>)  
        Html (<td>) Text (_) Html (</td>)  
      Html (</tr>)  
      Html (<tr>)  
        Html (<td>) Text (_) Html (</td>)  
      Html (</tr>)  
    Html (</table>)  
  Html (</body>)
```

Maximal repeats

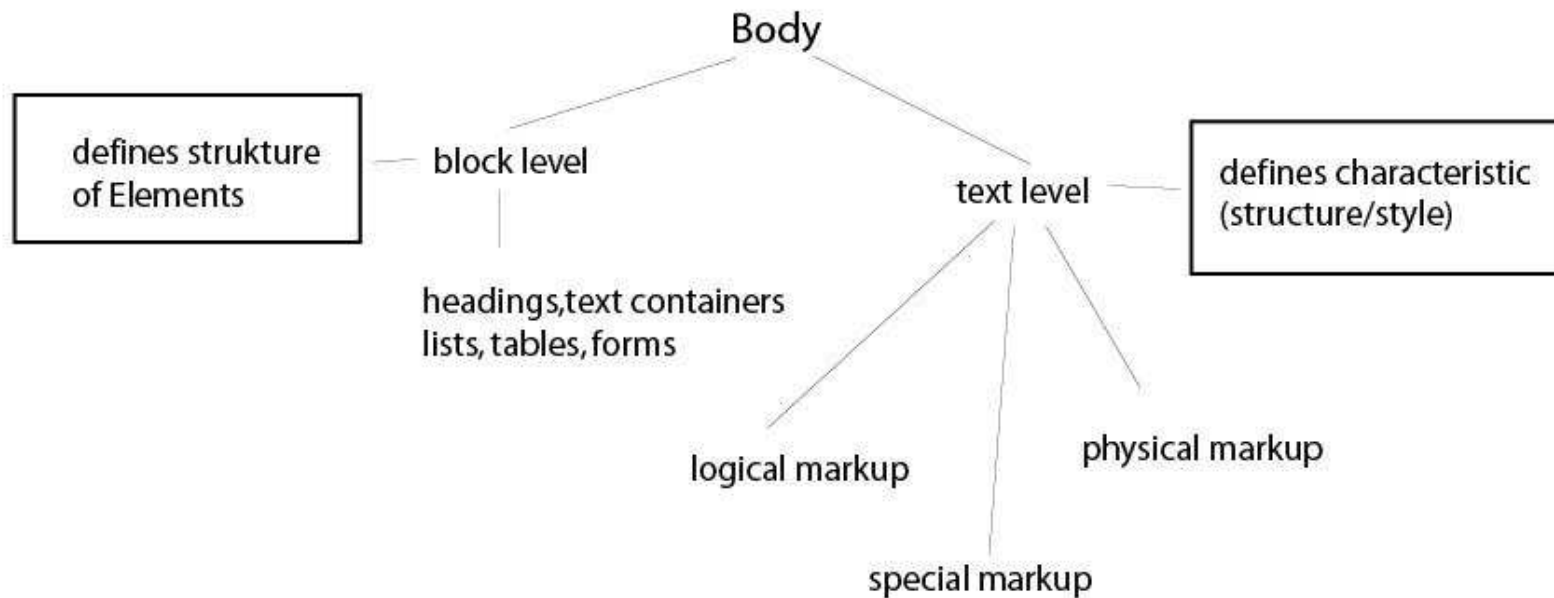
Least one pair of Tokens must be different in:

p_{i-1}, p_{j-1} th ($1 \leq i < j \leq \text{numOccur}$) Token
(left maximal)

and

$p_x - |a|, p_y - |a|$ ($1 \leq x < y \leq \text{numOccur}$)
(right maximal)

Structure levels



Using structure levels

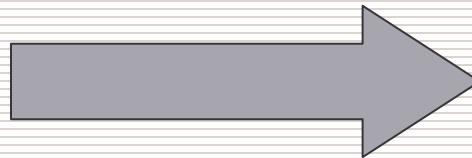
using the block-level ...

```
Html (<table>
  Html (<tr>
    Html (<td> Text (_) Html (</td>)
  Html (</tr>)
  Html (<tr>
    Html (<td> Text (_) Html (</td>)
  Html (</tr>)
Html (</table>)
```

Transformation structure level -> bit string

```
Html (<table>)  
  Html (<tr>)  
    Html (<td>)  
      Text (_)  
    Html (</td>)  
  Html (</tr>)  
  Html (<tr>)  
    Html (<td>)  
      Text (_)  
    Html (</td>)  
  Html (</tr>)  
Html (</table>)
```

Html(<table>)	000	Html(<td>)	100
Html(</table>)	001	Html(</td>)	101
Html(<tr>)	010	Text(_)	110
Html(</tr>)	011		



```
000  
  010  
    100  
      110  
        101  
          011  
            010  
              100  
                110  
                  101  
                    011  
                      001
```

Build suffixes

first suffix:

000 010 100 110 101 011 010 100 110 101 011 001

second suffix:

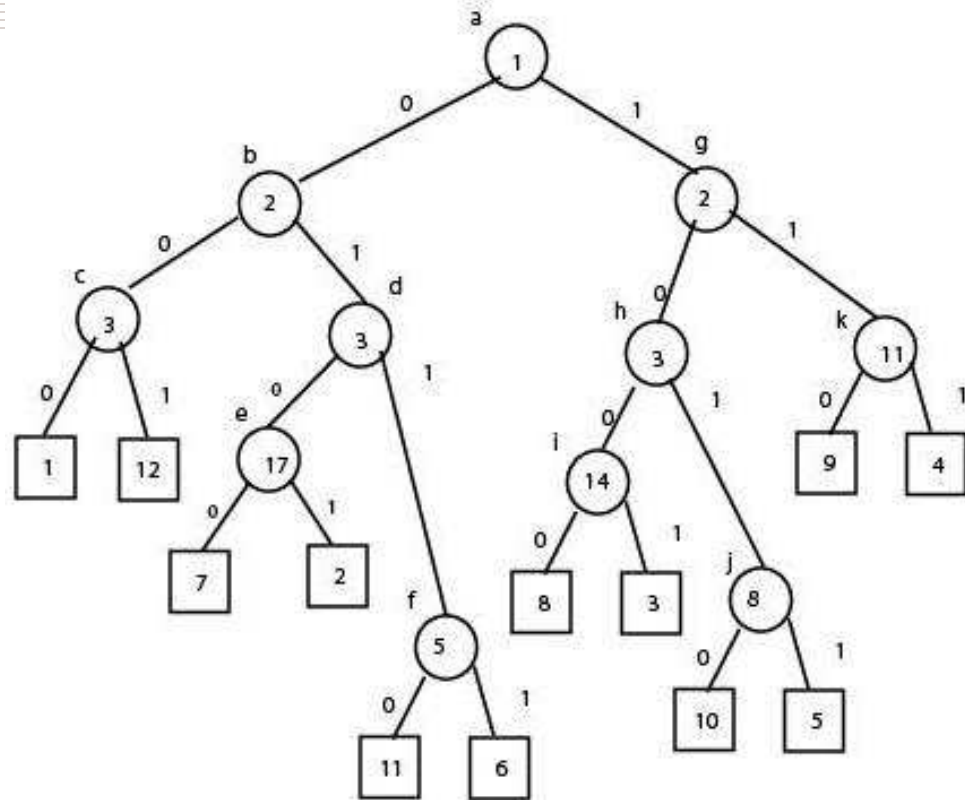
010 100 110 101 011 010 100 110 101 011 001

third suffix:

100 110 101 011 010 100 110 101 011 001

...

The PAT tree



Indexing position:

suffix 1: 000 010 100 110 101 011 010 100 110 101 011 001

suffix 2: 010 100 110 101 011 010 100 110 101 011 001

suffix 3: 100 110 101 011 010 100 110 101 011 001

suffix 4: 110 101 011 010 100 110 101 011 001

suffix 5: 101 011 010 100 110 101 011 001

suffix 6: 011 010 100 110 101 011 001

suffix 7: 010 100 110 101 011 001

suffix 8: 100 110 101 011 001

suffix 9: 110 101 011 001

suffix 10: 101 011 001

suffix 11: 011 001

suffix 12: 001

Validate maximal repeats

- **Regularity:**

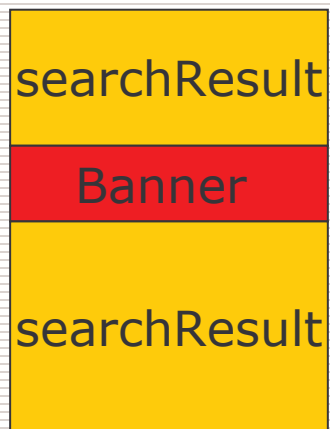
 - standard deviation of two adjacent occurrences

- **Compactness:**

 - density of max. repeats

Partitioning (1)

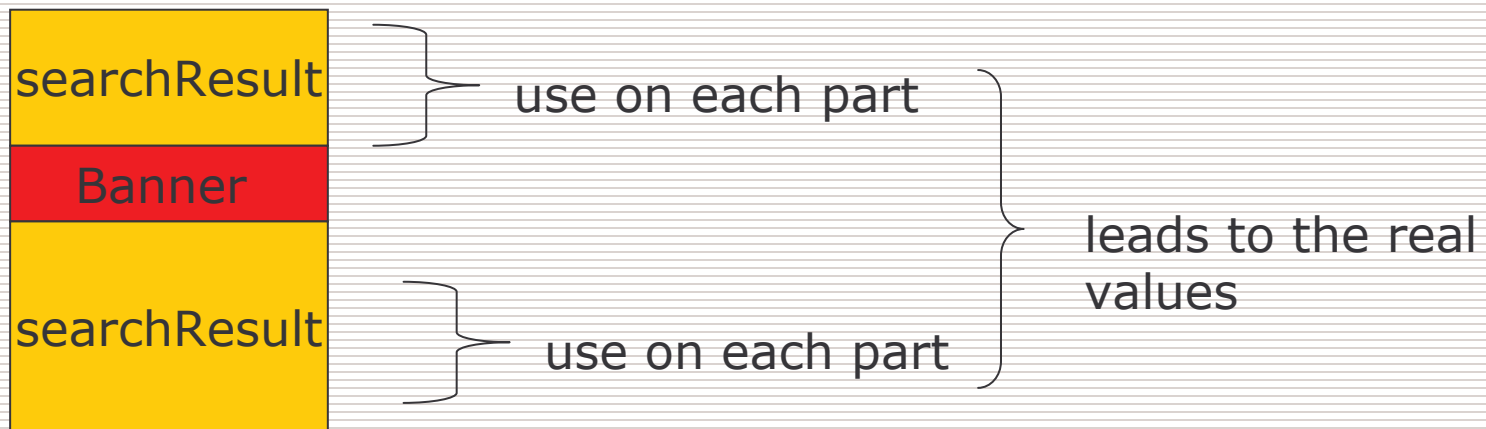
Searchresult:



the values for regularity
and density are falsified

Partitioning (2)

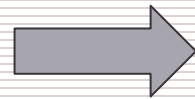
Searchresult:



Multiple String Alignment

- generalize a search pattern

```
1. f g h i k j  
2. f g h x k l  
3. g g h i k -
```



```
[g|f]gh[i|X]k[j|l|-]
```

IE PAD

- rule generator (uses this technique)
- pattern viewer
- extractor module

Thanks for your attention.

Any questions?