DISCOVERING RECORD BOUNDARIES

based on 1999’s paper by Embley / Jiang / Ng
ABOUT RENE KIESLER

- Consultant, focusing on Portals / CMS
- Bachelor for Software Engineering
- Master study for Information / Knowledge Management
- see www.kiesler.at
MOTIVATION

Why would someone want to detect record boundaries?

- tons of lists online (classifieds, member listings, ...)

- mostly in the web, not in a database
HEURISTIC 1/3

Embley / Liang / Ng chose the heuristic approach to finding boundaries.

- Origin: Greece

- „Eureka“ – I find

- different Meanings in Psychology, Philosophy, Law and Computer Science
- Psychology: Rules of Thumbs

- Philosophy: Describe something with something else

- Law: if case-by-case basis not feasible
HEURISTICS 3/3

- Computer Science: gain speed at cost of precision

- heuristics give good – but not perfect results (estimates)

- popular examples: shortest-path and Artificial Intelligence
CONCEPT OF DRB

- parse single page
- generate tag tree
- use heuristics
- build consensus
- finished
HTML TO TREE

- HTML has an inherent tree structure
- root element HTML
- nested tags
USED HEURISTICS 1/2

- **HT**: Highest Count Tag
  count fan-outs

- **IT**: Identifiable Separator Tags
  hr, tr, td, a, table, p, br, ...

- **SD**: Standard Deviation
  how many chars are between occurrences of hr / b / br / ...?
USED HEURISTICS 2/2

- RT: Repeating-Tag Pattern
  can we find reoccuring constructs like
  <hr><b>title</b></hr>?

- OM: Ontology Matching
  can we detect words that fit in our given ontology?
  can be skipped, if no Ontology available
EXAMPLE: TUWIS 1/2

- after searching for „informatik“, we get this list

- traditional table

- let’s look at the html code.
- large amount of TDs
- not-so-large amount of TRs
- only one big table
- a couple of meta-tags
- a bit of JavaScript
- perfect for our algorithm

```html
<tr>
  <td>106.009</td>
  <td>PR</td>
  <td title="Status der LVA: A wenn abgesagt, sonst leer">"</td>
  <td>2005S</td>
  <td>10.0</td>
  <td>WEINMÜLLER</td>
</tr>
```
EXAMPLE: CV 1/2

http://www.astro.umd.edu/~kzh/cv.html

- same structure, right?

- wrong!

- unstructured, no reoccurring patterns

- a bunch of tables, a bunch of divs

- heuristics would get confused
EXAMPLE: CV 2/2

- 11 differently nested DIVs

- 33 (!) completely different tables

- almost no common structure

- a couple of BRs (maybe these would work?)

- not ideal for our algorithm

<table>
<tr>
<td>
<table>
<tr><td></td></tr>
<tr><td></td></tr>
<tr><td></td></tr>
</table>
</td>
</tr>
<tr><td></td></tr>
</table>

<table>
<tr>
<td><b>Computing Skills</b></td>
</tr>
<tr><td></td></tr>
<tr><td></td></tr>
</table>

</table>
NOW WHAT?

- find consensus

- present record separator

- we don`t provide extraction to the database

- no crawling either