
data
IE
systems
task
extraction
pages
input
attribute
documents
Web
tools
rules
information
wrappers
HTML
different
wrapper
e.g
attributes
various
tasks
classified
dimension
used
may
approaches
example
system
support
structured
techniques
dimensions
semi-structured
survey
class
structures
researchers
require
online
source
proposed
Section
process
first
generated
TRANSACTIONS
Wrapper
mining
Extraction
pages
input
domain
free
automation
data-object
learning
use
KNOWLEDGE
domains
finite-state
written
usually
logic
need
values
XML
single
categories
col
major
list
However
targets
degree
token
patterns
tion
paper
training
sites
missing
Information
integration
criteria
object
unstructured
labeling
languages
relational
designed
server
structure
records
output
page
text
relevant
Finally
using
second
set
type
features
based
tags
form
taxonomy
processes
across
site
WI
three
ing
database
examples
program
sources
format
page-level
user
site-level
third
re
whereas
including
zero com
non-HTML
E-mail
multiple
IEEE
semi
called
regular
made
extract
one
two
evaluating
formats
process
first
generated
TRANSACTIONS
mation
might
elements
natural
main
many
also
etc
ENGINEERING Fig

[1]

Klassifizierungsmöglichkeiten für Information Extraction

Freie Universität Berlin
Institut für Informatik
Wintersemester 2011/2012
Seminar zu Programmiersprachen

Friedrich Große
Hernando Saenz Sanchez
Sebastian Schulz
Sebastian Starroske

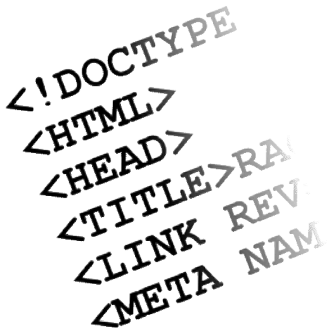
1. Einführung
2. Überblick verschiedener Klassifizierungsansätze
3. Klassifizierung nach Chang et.al.
4. Zusammenfassung

Klassifizierungen des **Inputs**:

- **unstrukturiert**
  - linguistisches Wissen nötig
- **(semi)strukturiert**
  - Extraktion mit syntaktischen Regeln

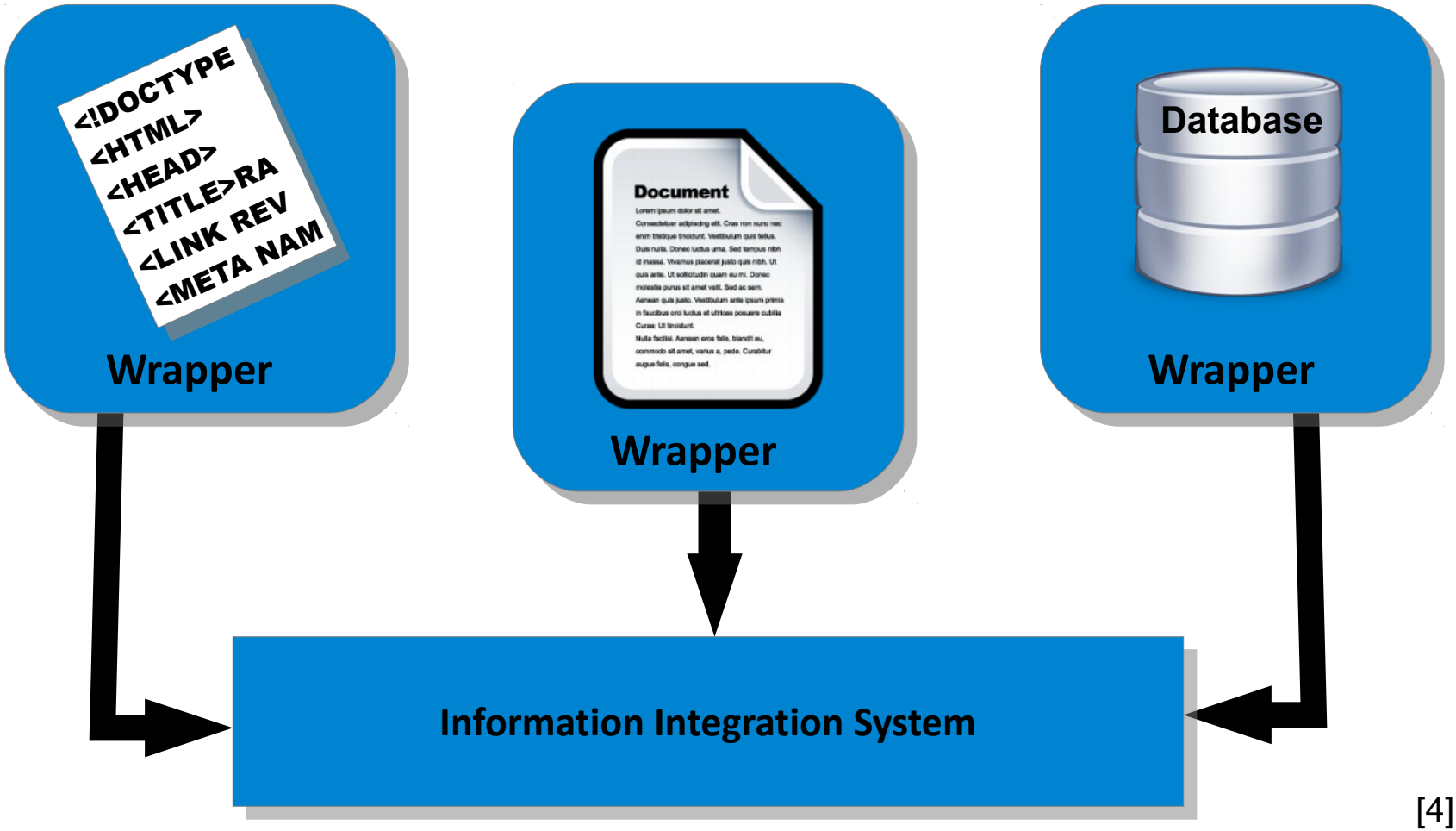
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam in scelerisque mauris. Cras nec magna vel lacus pretium feugiat sit amet in nulla. Donec fringilla ligula eu mi gravida vitae porttitor neque suscipit. Pellentesque ac mauris ligula.

Aliquam erat volutpat. Integer gravida semper nunc, nec condimentum arcu sollicitudin vitae. Mauris in ligula sapien, ac ultricies massa. Vivamus in enim vitae erat tempor auctor. Donec porttitor sem hendrerit elit lobortis id lobortis felis vestibulum. Nullam hendrerit hendrerit dignissim. Nunc eu justo tellus, non porta nunc.



[2], [3]

Was ist ein Wrapper?



[4], [5]

1. Einführung
2. **Überblick verschiedener Klassifizierungsansätze**
3. Klassifizierung nach Chang et.al.
4. Zusammenfassung

Klassifizierung nach Leander, da Silva, Ribeiro-Neto und Teixeira

Paper:

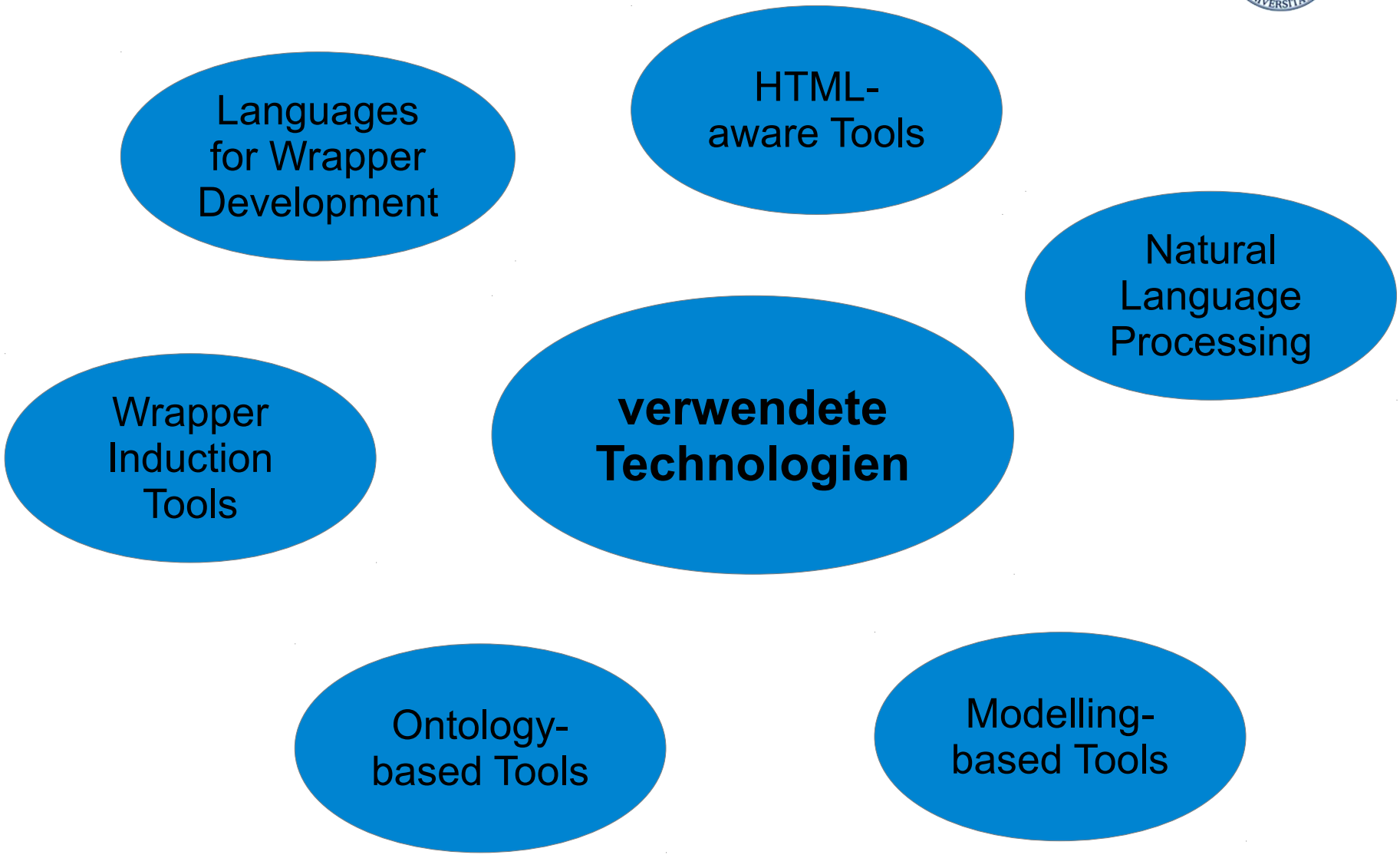
A brief **Survey** of Web Data **Extraction Tools**

2002

Klassifizierung anhand der eingesetzten Technologien

[6]

2. Klassifizierung nach Laender et.al.



Languages for Wrapper Development

- Einsatz einer Sprache, die das Erzeugen von Wrappern erleichtern
- dient als Alternative zu Mehrzwecksprachen wie Java oder C
- Beispiele:
  - Minerva
  - TSIMMIS
  - Web-OQL

select [*y.Title*, *y'.Url*]
from *x* **in** *csPapers*, *y* **in** *x'*
where *y.Authors* ~ "Smith"

[7], S.4

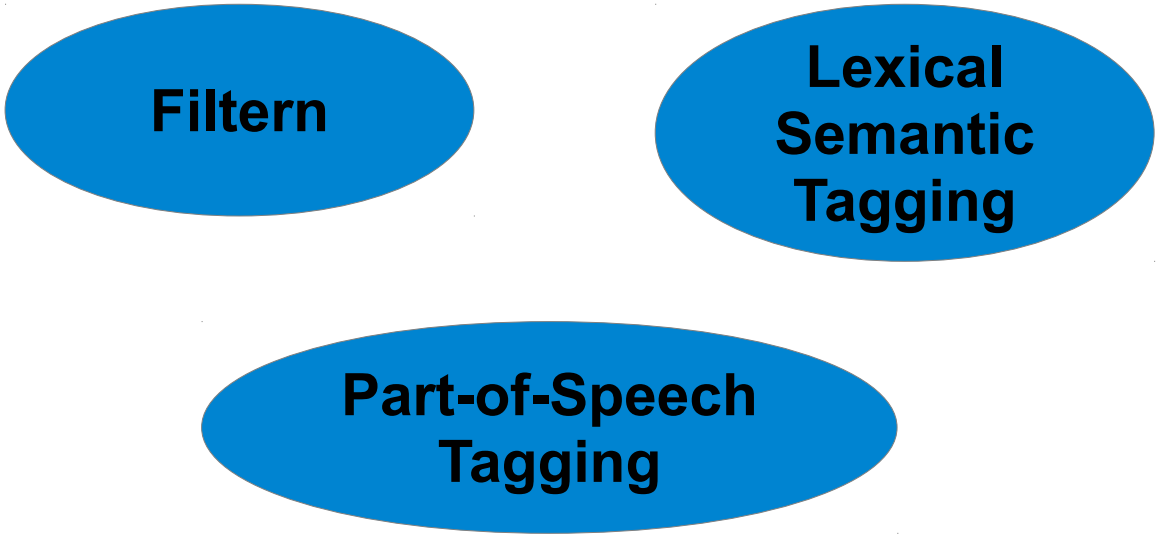
HTML-aware Tools

- Ausnutzen der Struktur von HTML-Seiten
- Darstellen der Tag Hierarchie in einem Parsebaum
- Beispiele:
  - W4F
  - XWRAP
  - RoadRunner

NLP-based Tools

- Texte in natürlicher Sprache

- Beispiele:
  - RAPIER
  - WHISK



Wrapper Induction Tools

- Erzeugen von trennzeichen-basierten Extraktions Regeln anhand von Trainingsdaten
- Basiert auf Format und Struktur der Dokumente
- Beispiele:
  - WIEN
  - STALKER
  - SoftMealy

Modeling-based Tools

- Lokalisieren von Daten anhand einer Gegeben Struktur
- Struktur wird durch Listen, Tupeln, usw. gegeben
- Beispiele:
  - NoDoSe
  - DEByE

Ontology-based Tools

- Verwendung von domänenspezifischen Ontologien
- Vorteil: unabhängig von Format und Struktur der Quelldaten
- Beispiel: BYU (Brigham Young University)

Qualitative Analyse

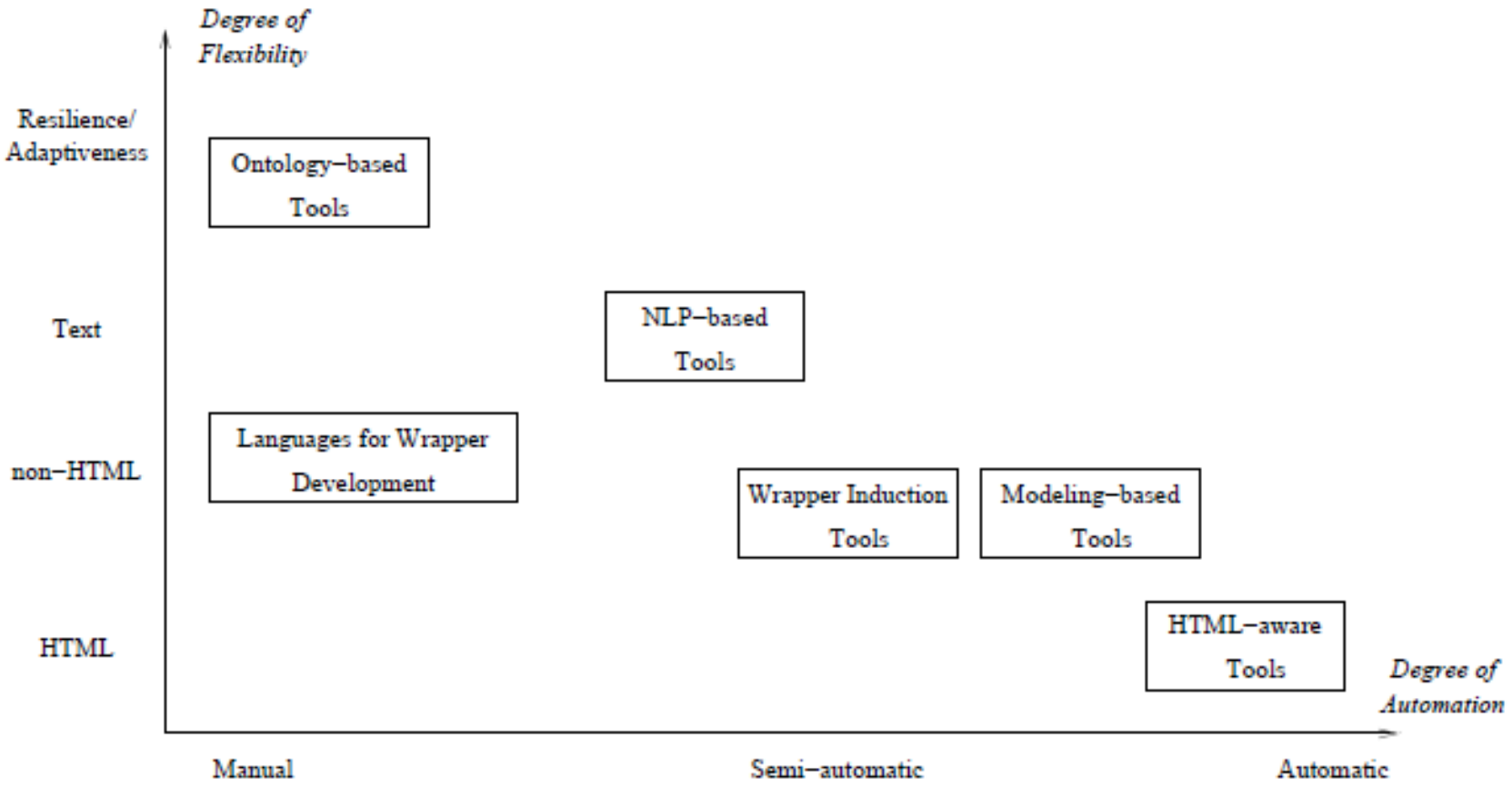
- Automatisierungsgrad
- Objekte mit komplexer Struktur
- Inhalte (semistrukturierter Text oder Daten)
- Graphische Benutzerschnittstelle
- XML Ausgabe
- Unterstützung von nicht-HTML Ressourcen
- Belastbarkeit und Anpassungsfähigkeit

2. Klassifizierung nach Laender et.al

Tools		Degree of Automation	Support for Complex Objects	GUI	XML Output	Support for Non-HTML Sources	Type of Page Contents
Languages	Minerva	Manual	Coding	No	Yes	Partial	SD
	TSIMMIS	Manual	Coding	No	No	Partial	SD
	Web-OQL	Manual	Coding	No	No	None	SD
HTML-aware	W4F	Semi-Automatic	Coding	Yes	Yes	None	SD
	XWRAP	Automatic	Yes	Yes	Yes	None	SD
	RoadRunner	Automatic	Yes	Yes	No	None	SD
NLP-based	WHISK	Semi-Automatic	No	Yes	No	Full	ST
	RAPIER	Semi-Automatic	No	Yes	No	Full	ST
	SRV	Semi-Automatic	No	Yes	No	Full	ST
Induction	WIEN	Semi-Automatic	No	Yes	No	Partial	SD
	SoftMealy	Semi-Automatic	Partial	Yes	No	Partial	SD
	STALKER	Semi-Automatic	Yes	Yes	No	Partial	SD
Modeling-based	NoDoSE	Semi-Automatic	Yes	Yes	Yes	Partial	SD
	DEByE	Semi-Automatic	Yes	Yes	Yes	Partial	SD
Ontology-based	BYU	Manual	Coding	Yes	No	Full	ST/SD

[6] S.9, Tabelle 1

2. Klassifizierung nach Laender et.al



[6] S9, Abbildung 3

Klassifizierung nach Sawaragi

Tutorial:

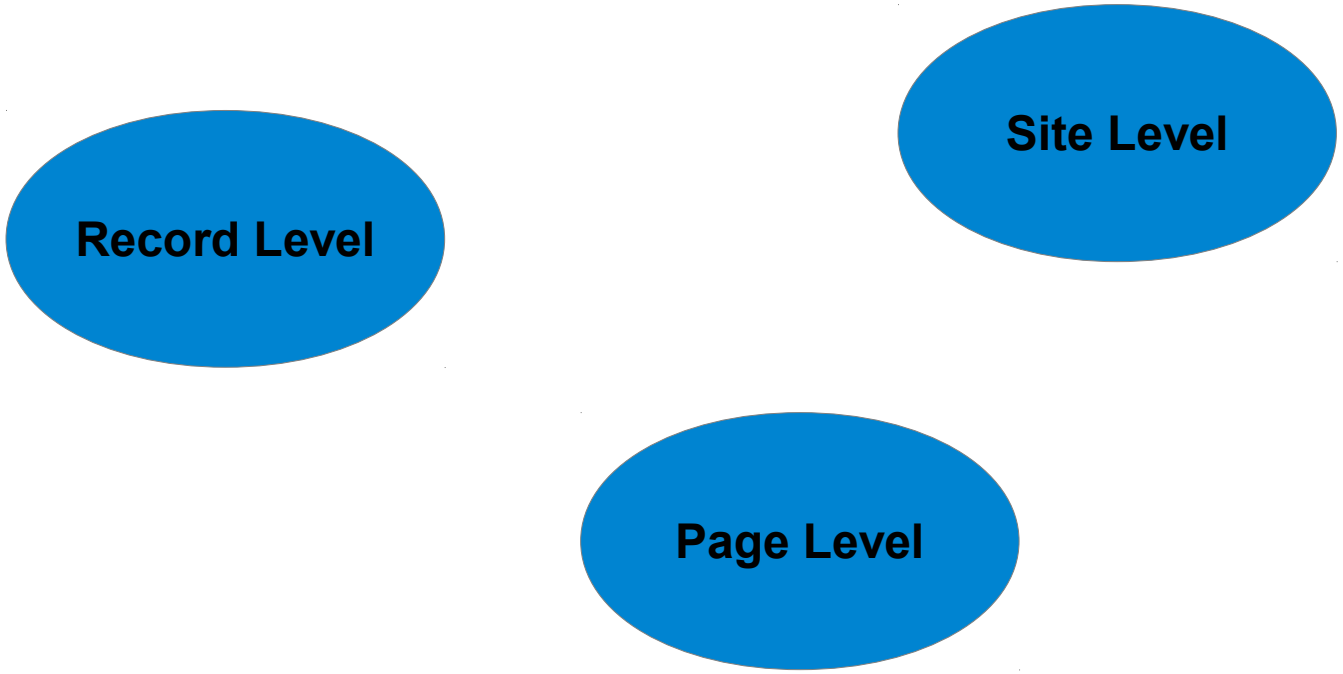
Automation in **Information Extraction** and Integration

2002

Klassifizierung von **HTML Wrappers**

[8]

Klassifizierung von HTML-Wrapper



Klassifizierung nach Hsu und Dung

Paper:

Generating Finite-State transducers for

semi-structured data extraction from the web

18. Februar 1998

SoftMealy: Ein Algorithmus zur **induktiven Wrapper** Erzeugung

[9]

Klassifizierung der **Wrapper Construction**

- **handgeschrieben** in allgemeiner Programmiersprache
  - schwer zu aktualisieren / warten
  - benötigt Experten
- **programmiert** mit speziellen Werkzeugen
  - schnell zu programmieren
  - ausdrucksstark in Problemdomäne
  - Benötigt spezialisierte Experten

Klassifizierung der **Wrapper Construction**

- **automatisch generiert** (Heuristiken)
  - Schnell
  - keine Experten notwendig
  - kann nur Bruchteil zuverlässig abdecken

Klassifizierung der **Wrapper Construction**

- **Wrapper Induktion**
  - „State of the Art“
  - Wrapper werden hergeleitet von Trainingsdaten
  - endliche deterministische Automaten
  - Übersetzung der Eingabe

Klassifizierung nach Hsu und Chang

Paper:

Automatic information extraction from semi-structured Web pages by **pattern discovery**

2002

IEPAD: Ansatz der **Wrapper Erzeugung** per Mustererkennung

[10]

Klassifizierung der **Wrapper**

- **Trennzeichenbasiert** (delimeter-based)

  Cannon S40: **$**129****

- **Musterbasiert** (pattern-based)

  Cannon S40: **$***[number]*****

- **Grammatikbasiert** (grammar-based)

  Webpage → Title Cameras

  Cameras → Model: Price

  Model → String

  **Price → $ String **

- **Heuristiken**

Klassifizierung aus **Sicht des Benutzers**

- Grad der Automatisierung:
  - benötigt **Programmierer**
  - benötigt **Anmerkungen** und **Beispiele**
  - **annotationsfreie** Systeme
  - **halbüberwachte** Systeme

Klassifizierung nach Kushmerick und Thomas

Paper:

Adaptive information extraction:

Core **technologies** for information **agents**

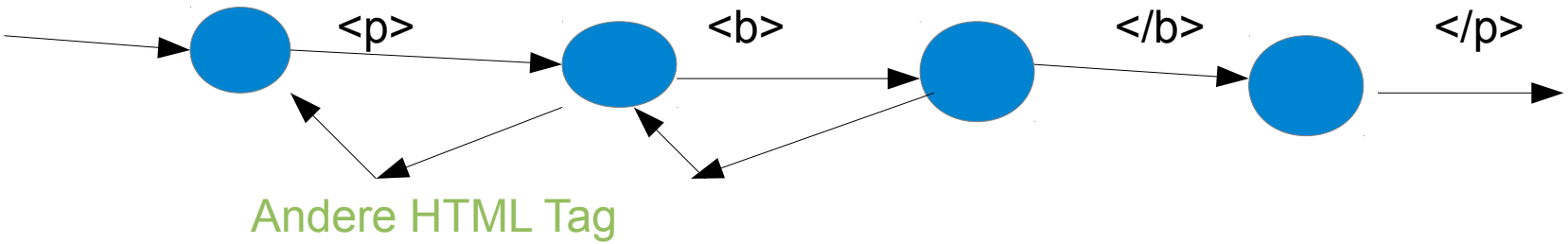
2003

Klassifizierung anhand der Art der Logik (finite-state vs. relational)

[11]

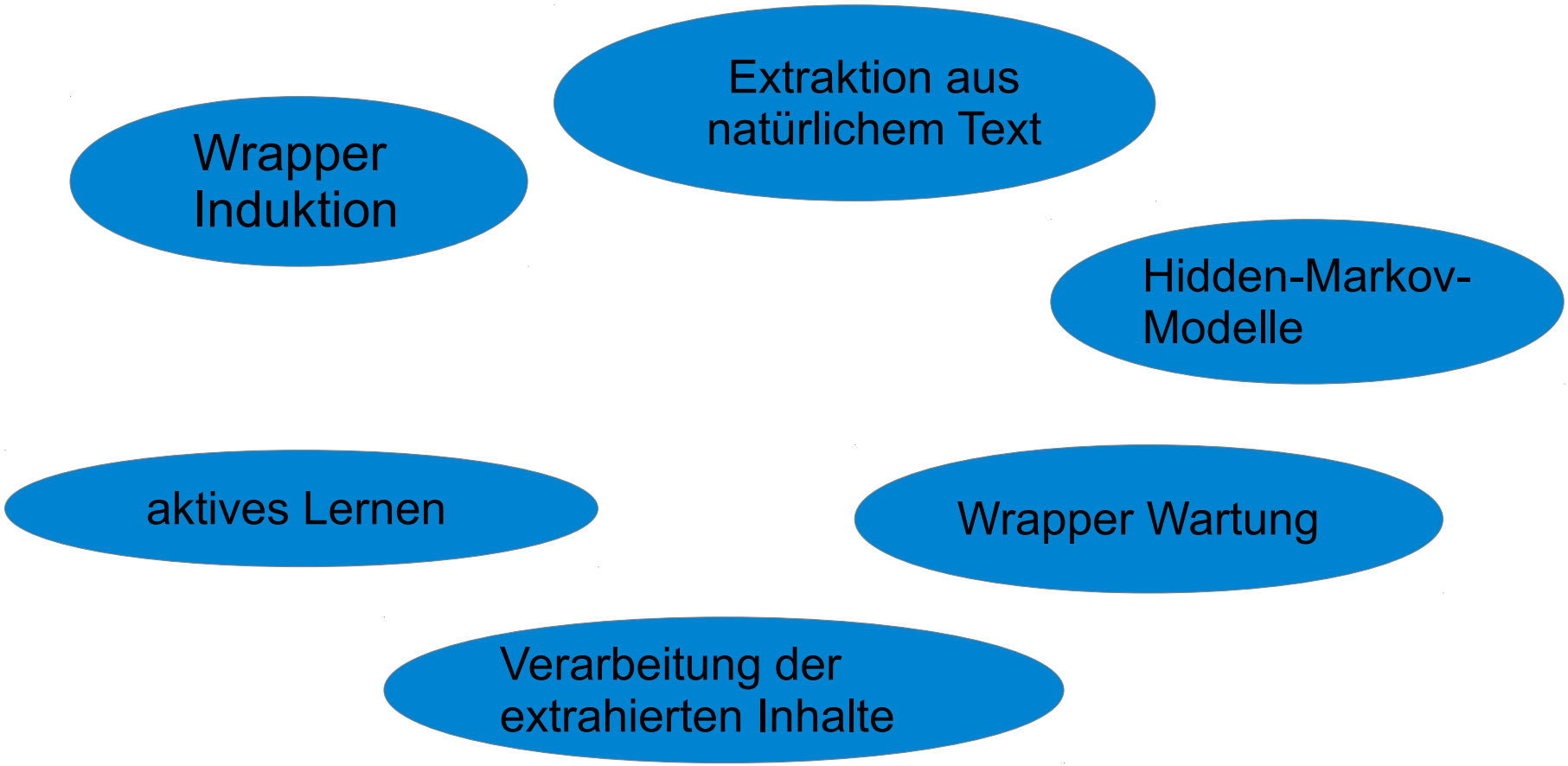
Einteilung von IE in zwei Kategorien:

- Finite-state
  - Äquivalent zum regulären Grammatiken oder Automaten (WIEN, STALKER, SoftMealy)



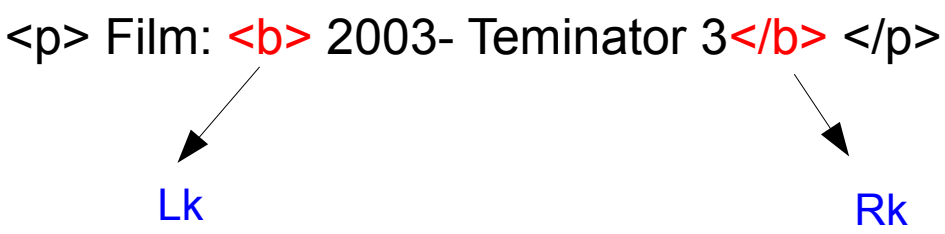
- Relational learning Techniken

Was gehört zur Text Extraktion?



Wrapper Induktion

- Relativ verallgemeinerbar und effizient
- Basiert auf Trennzeichen (delimiter)



- L_k = Spalte links delimiter
- R_k= Spalte rechts delimiter

6 verschiedene Induktion-Wrappers klassen

- LR (Left-Right)

  Set { (l1,r1),(l2,r2),.....,(lk,rk)

- HLRT (Head-Leaf-Right-Tail)
- OCLR (Open-Close-Left-Right)
- HOCLRT (Head-Open-Close-Left-Right-Tail)
- NLR (Nested-Left-Right)
- NHLRT(Nested-Head-Left-Right-Tail)

Schwierigkeiten

- Fehlende Attribute

   Film: 2003- Teminator 3

   Film: Thor (N)

   Film: 2004- Kill bill vol 2

- Mehrwertige Attribute

   Stadt: Madrid(Angebot)

   Stadt: Paris

   Stadt: Berlin 50% reduziert

Schwierigkeiten

- Multi Attribut Sortierung

  Top 10

  1 Moves Like Jagger-Maroon 5 ▲

   2▼Paradise-Coldplay

- Disjunktive Trennzeichen

  Preisvergleich

  <td> amazon.de </td><td> €51.24</td>

  <td> redcom.de </td><td> €49.84 </td>

  <td> conrad.de </td><td> €60.20 </td>

Schwierigkeiten

- Nicht vorhandene Trennzeichen

  INF4016

  MATE5124

- Typographische Fehler und Ausnahmen

  <td> redcom.de </td><td> €49.84 </td>

  <td> conrad.de </td><td> €60.20 </td>

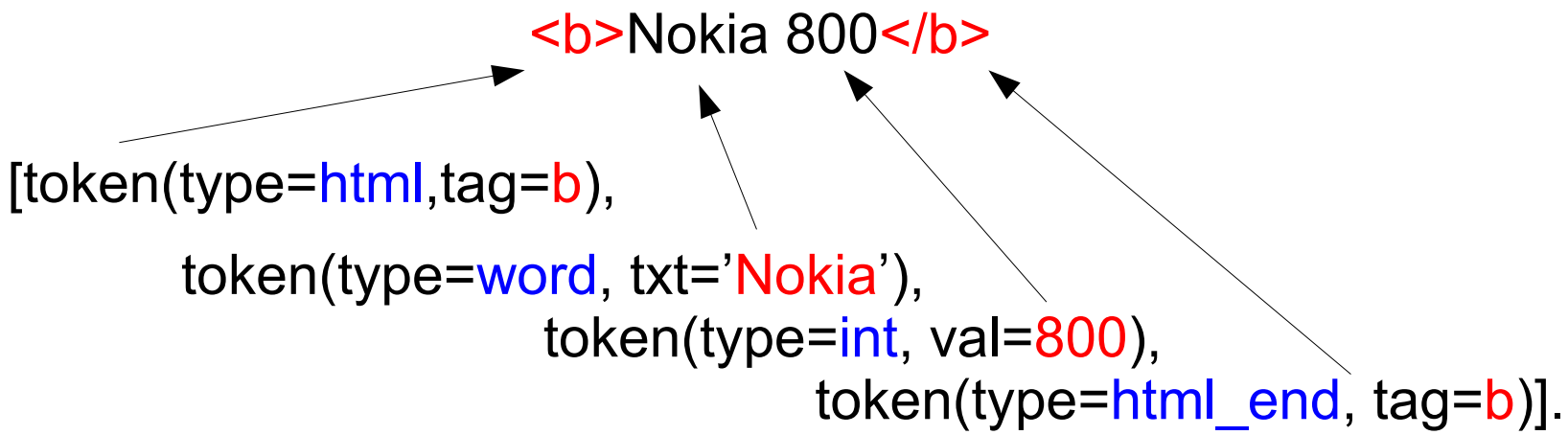
  <td> amazon.de </td><td> €51.24 ___</td>

„Relational learning“ Techniken

- Prolog-ähnliche Logik Regeln (SRV,Cristal, Webfoot)

- Regel lernen
  - One-Shot
  - Sequentiell Abdeckung
  - First order rules. (FOIL)
    - Aussagenlogik

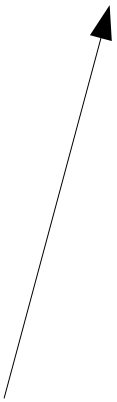
Thomas Dokument Umformung



token(type=html,tag=X)

- Beispiel Prolog Regel

```
link(Description, Url) :- pos(P, token(type=html,tag=a, href=Url)),
                          sequence(P, E, TokenSeq),
                          not in(token(type=html_end, tag=a), TokenSeq),
                          next(E, token(type=html_end, tag=a))
```



Ziele

Klassifizierung nach Muslea

Paper:

A **Hierarchical Approach** to **Wrapper Induction**

1999

Regelbasierte Wrapper und deren Umsetzung in STALKER

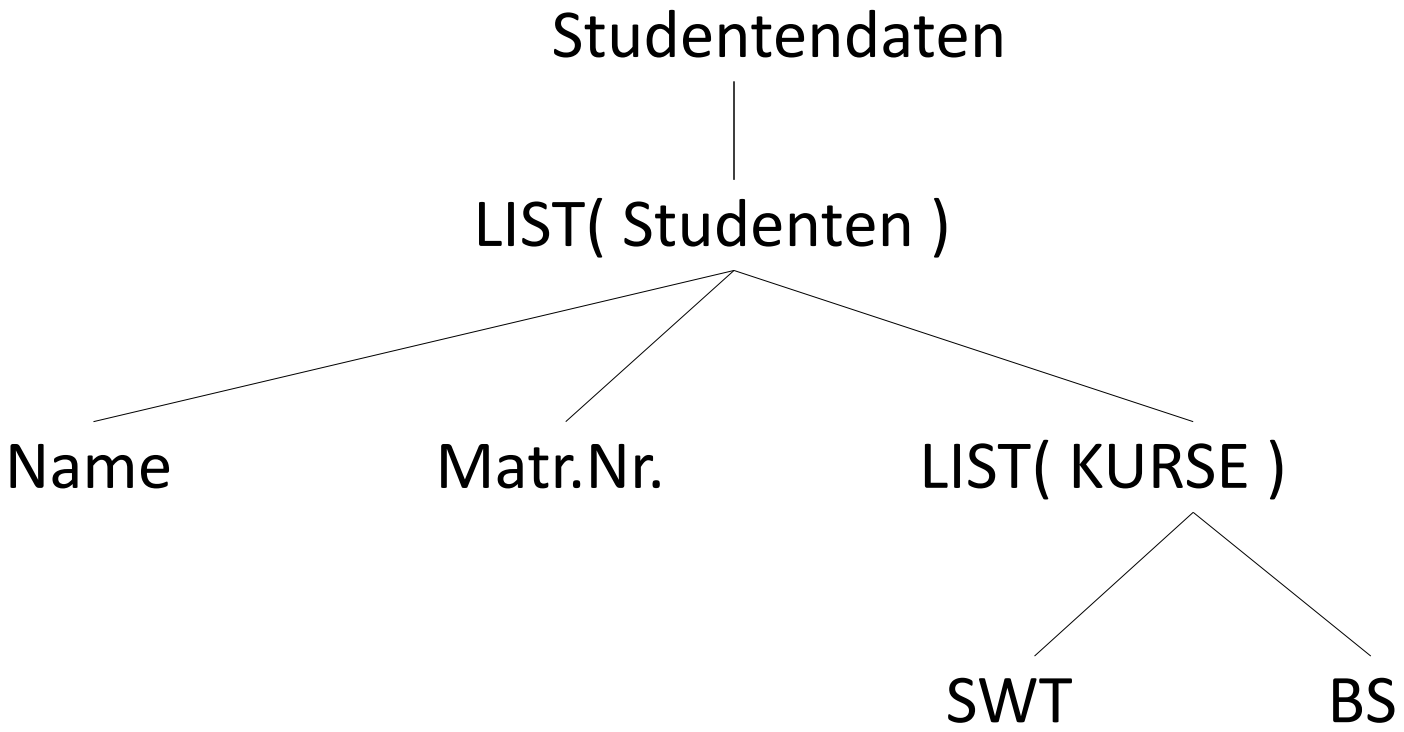
[12]

3 verschiedene **Klassen**: Art des **Input**s

- **Freier Text** (unstrukturiert)
  - semantische & syntaktische Beschränkung

- **Online Dokument** (semi-strukturiert)
  - semantische & syntaktische Beschränkung
  - **Regelsysteme** (Wrapper Induction Systems)

Embedded Catalog (EC)



Dokument = Sequenz von Token

1: <p> Name: Sebastian
2: <p> Matr.Nr: 123456
3: <p> Veranstaltungen:
4: Seminar Webtechnologien, PCQ 0345
5: Betriebssysteme, ADL 3123

Regeln für IE

- R1 = *SkipTo(*Name*) SkipTo(*<b>*)*

- R2 = *SkipTo(*AllCaps*) NextLandmark(*Number*)*

```
1:  <p> Name: <b> Sebastian </b>
2: <p> Matr.Nr: 123456
3: <p> Veranstaltungen: <i>
4: Seminar Webtechnologien, PCQ 0345 </i> <br>
5: <i> Betriebssysteme, ADL 3123 </i>
```

Wrappen von Dokumenten mit tiefer Schachtelung

Flexibler Ansatz (Geschwisterknoten getrennt behandelt)

[13]

Klassifizierung nach Kuhlins und Tredwell

Paper:

Toolkits for Generating Wrappers

A Survey of Software Toolkits for Automated Data Extraction from Websites

2002

Vorstellung und Vergleich von kommerziellen und nicht-kommerziellen Toolkits

[14]

Toolkit stellt Wrapper **automatisiert** her

- **Parameter** vom Benutzer

Unterscheidung:

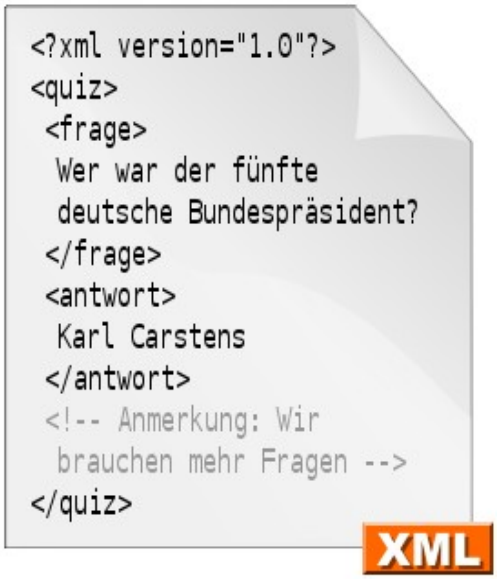
vs.

(kommerzielle Wrapper-Generating Toolkits)

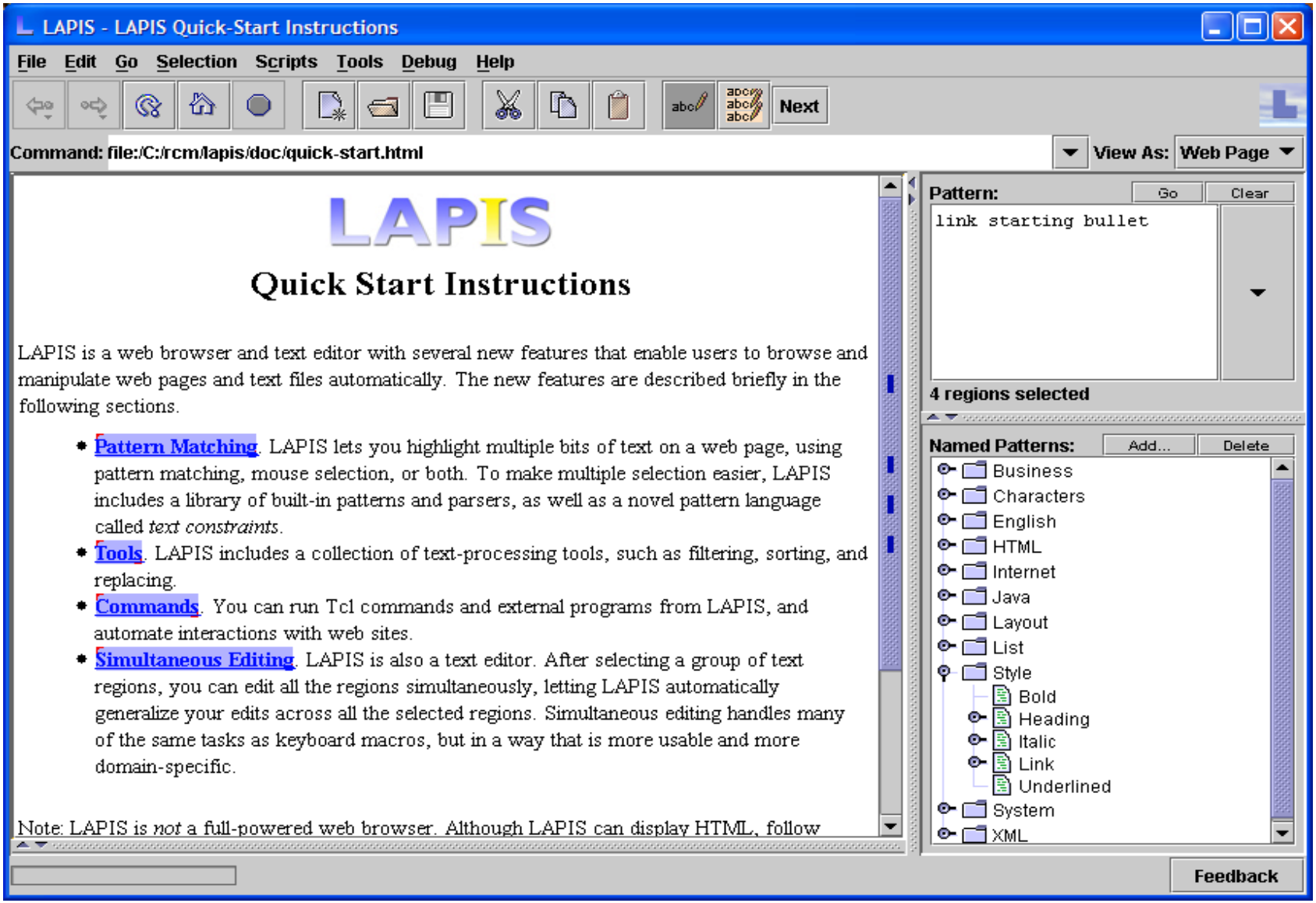
(nicht-kommerzielle Wrapper-Generating Toolkits)

[15], [16]

- Weitere wichtige Abgrenzungen
  - Shell vs. GUI
  - Web Crawling
  - Output



[17]


LAPIS - LAPIS Quick-Start Instructions
File Edit Go Selection Scripts Tools Debug Help
Next
Command: file:/C:/rcm/lapis/doc/quick-start.html
View As: Web Page
LAPIS
Quick Start Instructions
LAPIS is a web browser and text editor with several new features that enable users to browse and manipulate web pages and text files automatically. The new features are described briefly in the following sections.
Pattern Matching. LAPIS lets you highlight multiple bits of text on a web page, using pattern matching, mouse selection, or both. To make multiple selection easier, LAPIS includes a library of built-in patterns and parsers, as well as a novel pattern language called text constraints.
Tools. LAPIS includes a collection of text-processing tools, such as filtering, sorting, and replacing.
Commands. You can run Tcl commands and external programs from LAPIS, and automate interactions with web sites.
Simultaneous Editing. LAPIS is also a text editor. After selecting a group of text regions, you can edit all the regions simultaneously, letting LAPIS automatically generalize your edits across all the selected regions. Simultaneous editing handles many of the same tasks as keyboard macros, but in a way that is more usable and more domain-specific.
Note: LAPIS is not a full-powered web browser. Although LAPIS can display HTML, follow
Pattern:
Go
Clear
link starting bullet
4 regions selected
Named Patterns:
Add...
Delete
Business
Characters
English
HTML
Internet
Java
Layout
List
Style
Bold
Heading
Italic
Link
Underlined
System
XML
Feedback

[18]

2. Klassifizierung nach Kuhlins & Tredwell

Toolkit	Output Data	Java API	Open Source	Source Code	Web Crawling	GUI	Editor	Scripting Language	Tool Support
Araneus	XML, Text	✓	only API	Java	—	—	—	EDITOR	—
BYU	Text, Tabular	✓	—	Java	—	—	—	ontologies	—
COMMIX	HTML, XML, Text	No explicit API	—	Java	✓	✓	—	—	(✓) Chinese toolkit
DEByE	XML, Text	No explicit API	—	Java	✓	✓	—	—	✓
Info Extractor	DOM type structure	✓ JDBC	—	Java	—	—	—	reg. expr., grammar rules	—
Jedi	XML, Text	✓	—	Java	✓	—	—	JEDI	—

[14] S.4, Tabelle 1

2. Klassifizierung nach Kuhlins & Tredwell

Company	Toolkit	Demo Version	Output Data	Web Crawling	Interface for External Programs	GUI	Editor	Scripting Language
Caesius Software	WebQL (Web Query Language)	—	Text, HTML	✓	✓	✓	✓	WebQL
Connotate Technologies	vTag	—	XML	✓	✓	✓	—	—
Crystal Software	TextPipe	✓	XML, Text	—	✓	✓	✓	Perl, VBScript, JScript, Rexx, Python
Data Junction	Content Extractor	✓	XML, Text	✓	✓	✓	✓	CXL
Extradata Technologies	Unwwwrap	✓	HTML, Text, Table	(✓) based on "Bookmarks"	Plug-In for MS Internet Explorer	✓	—	—
Fetch Technologies	Agent Builder	—	XML, Text	✓	ODBC Database	✓	—	—
Fire Spout	ETL Engine	✓	HTML, XML, Text	✓	✓	✓	—	—

[14] S.5, Tabelle 2

- **Leander** et.al.:
  - **Wie?** - Technologien (z.B. HTML-aware tools, NLP-based tools, Modeling-based tools, etc.)

- **Hsu** & **Dung**:
  - **Wie?** - Herstellung:
    - handgeschrieben
    - programmiert (spezielle Sprache)
    - handgeschrieben (Heuristiken)
    - Wrapper Induktion

- **Hsu** & **Chang**:
  - **Wie?** - Grad der Automatisierung
    - benötigt Programmierer
    - benötigt Anmerkungen und Beispiele
    - annotationsfreie Systeme
    - halbüberwachte Systeme

- **Kushmerick**:
  - **Wie?** - „finite state“ & „relation learning“

- **Muslea**:
  - **Wie?** - 3 Klassen (Input & Extraktionsmuster):
    - Freier Text (Semantische & Syntaktische Beschr.)
    - Online Dokument (Semantische & Syntaktische Beschr.)
    - Online Dokument (Regelssystem)

- **Sarawagi**:
  - **Wie?** - Extraktionsaufgabe
    - record-level
    - page-level
    - site-level

- **Kuhlins** & **Tredwell**:
  - **Wie?** - kommerziell vs. nicht-kommerziell

1. Einführung
2. Überblick verschiedener Klassifizierungsansätze
3. **Klassifizierung nach Chang et.al.**
4. Zusammenfassung

Klassifizierung nach Chang, Kayed, Girgis und Shaalan

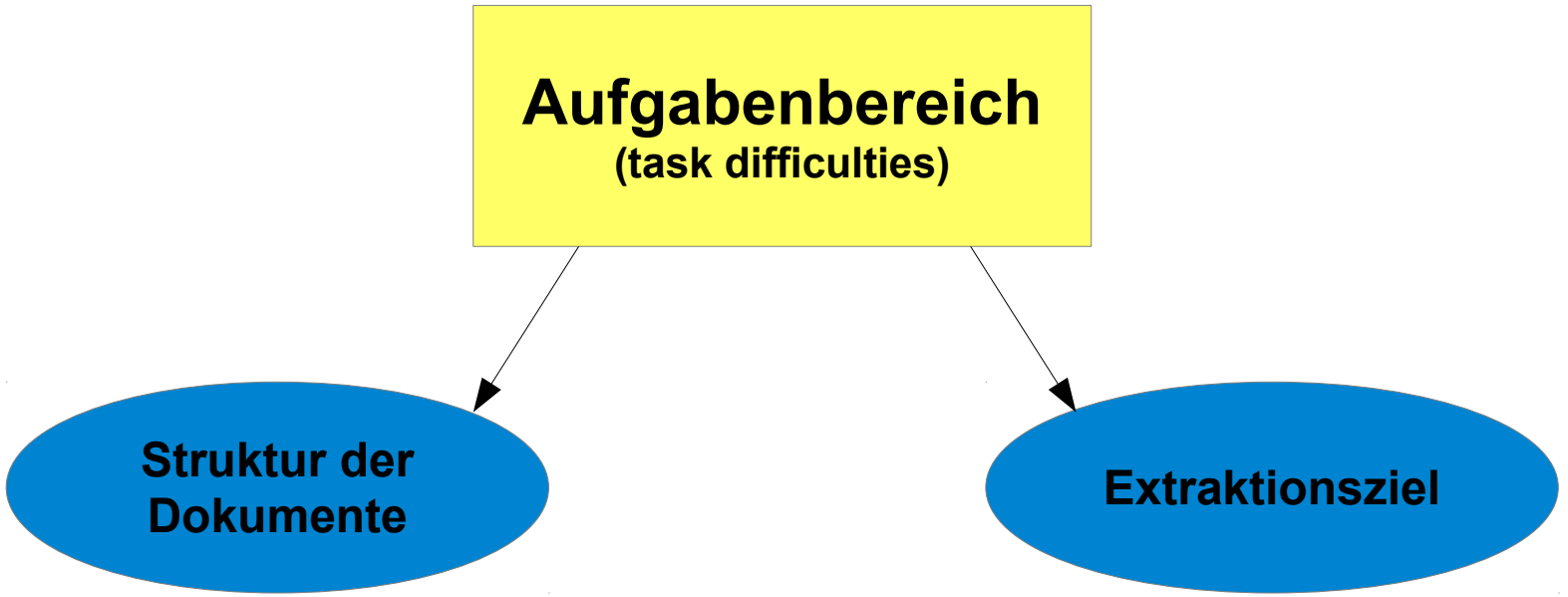
Paper:

A Survey of Web Information Extraction Systems

2006

Klassifizierung von IE nach **Aufgabenbereich**, **Automatisierungsgrad** und **verwendeten Technologien**

[19]


Aufgabenbereich
(task difficulties)
Struktur der Dokumente
Extraktionsziel

verwendete Techniken

- Teile
  - Trennung der Eingänge (Tokenize)
    - Tag-level, word-level encoding
  - Anwendung von Extraktionsregeln
  - Verbindung der Werte in Datensatz

Extraktionsregeln

- Verallgemeinerung , Pattern Mining, Logik-Programmierung
- 2 Arten von Ausdrücke
  - Reguläre Grammatiken
  - Logik Regeln
- Pfad-Ausdrücke (html.head.title),Syntaktik ,Semantik oder Delimiter-basiert

Grad der Automatisierung

- Benutzererfahrung
- Verallgemeinerung
- Einschränkungen
- Ausgabeformat (XML, DB, etc.)
- API Integration Unterstützung

1. Einführung
2. Überblick verschiedener Klassifizierungsansätze
3. Klassifizierung nach Chang et.al.
4. **Zusammenfassung**

Referenzierte Artikel

- Merkmale:
  - Herstellung
  - Eingabe
  - Automatisierungsgrad
  - Arbeitsweise
  - eingesetzte Technologien

Chang, Kayed, Girgis und Shaalan

- 3 Bereiche:
  - Task difficulty
  - Technique used
  - Automation Degree

Vielen Dank für die Aufmerksamkeit!

Fragen ?

[1] http://www.wordle.net

[2] http://www.irise.com/img/iblocs/irise/300/loremipsum_300.png

[3] http://i3.squidoocdn.com/resize/squidoo_images/-1/lens18139053_1310397437untitled.jpg

[4] http://icons.iconarchive.com/icons/deleket/sleek-xp-basic/256/Document-icon.png

[5] http://images.findicons.com/files/icons/725/colobrush/256/database.png

[6] A. H. F. **Laender**, B. **Ribeiro-Neto**, **DA Silva** und **Teixeira**, „A brief survey of Web data extraction tools.“ 2002.

[7] G.O.**Arocena**, A.O.**Mendelzon**, „WebOQL: Restructuring Documents, Databases and Webs“ In Proceedings of the 14th IEEE International

[8] S. **Sarawagi**, „Automation in information extraction and integration“, Tutorial of The 28th International Conference on Very Large Data Bases (VLDB). 2002

[9] C.-N. **Hsu** und M. **Dung**, „Generating finite-state transducers for semi-structured data extraction from the web.“ Journal of Information Systems 23(8): 521-538. 1998.

[10] C.-H. **Chang**, C.-N. **Hsu** und S.-C. **Lui**,
„Automatic information extraction from semi-Structured Web Pages by pattern discovery.“ Decision Support Systems Journal, 35(1): 129-147. 2003.

[11] N. **Kushmerick**,
„Adaptive Information Extraction: Core technologies for Information agents.“ In Intelligent Information Agents R&D in Europe: An AgentLink perspective (Klusch, Bergamaschi, Edwards & Petta, eds.). Lecture Notes in Computer Science 2586, Springer. 2003.

[12] I. **Muslea**,S. **Minton** und C. **Knoblock**,
„A hierarchical approach to wrapper induction.“
Proceedings of the Third International Conference on Autonomous Agents (AA-99). 1999.

[13] http://www.iconarchive.com/show/button-icons-by-deleket/Button-Add-icon.html

[14] S. **Kuhlins** und R. **Tredwell**,
„Toolkits for generating wrappers.“
Net.ObjectDays 2002: Objects, Components, Architectures, Services and Applications for a Networked World,
http://www.netobjectdays.org/, LNCS 2591. 2002.

[15] http://www.elec-intro.com/EX/05-14-06/makkelijk_geld_verdienen.jpg

[16] http://www.computers-home.de/userbilder/frei.jpg

[17] http://upload.wikimedia.org/wikipedia/commons/e/e9/XML_%28de%29.svg

[18] http://groups.csail.mit.edu/uid/lapis/images/browser.png

[19] C.-H. **Chang**, M. **Kayed**, M. R. **Girgis** und K. **Shaalan,**
„A Survey of Web Information Extraction Systems.“
IEEE transactions on knowledge and data engineering, TKDE-0475-1104.R3. 2006.