Learning to Construct Knowledge Bases from the World Wide Web

Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal, Neigam, Sean Slattery

(2000)

Oliver Rohr

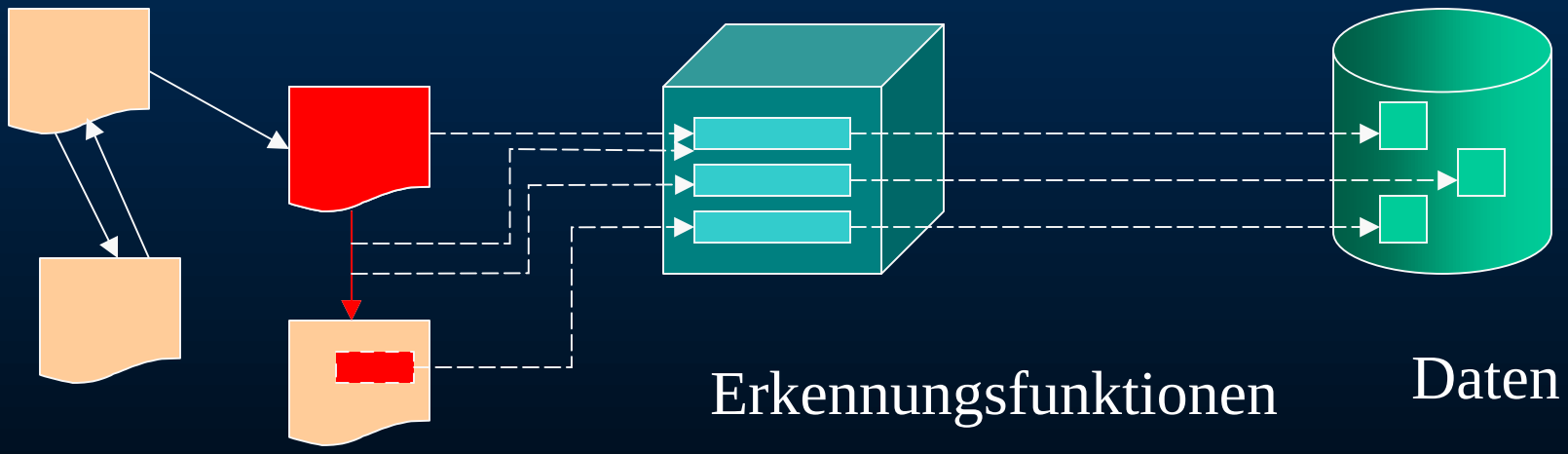
14.1.2003

Gliederung

- Überblick
- Problemspezifikation
- Erkennen von Klasseninstanzen
- Erkennen von Relationsinstanzen
- Extraktion aus Textsegmenten
- Fazit, Ausblick

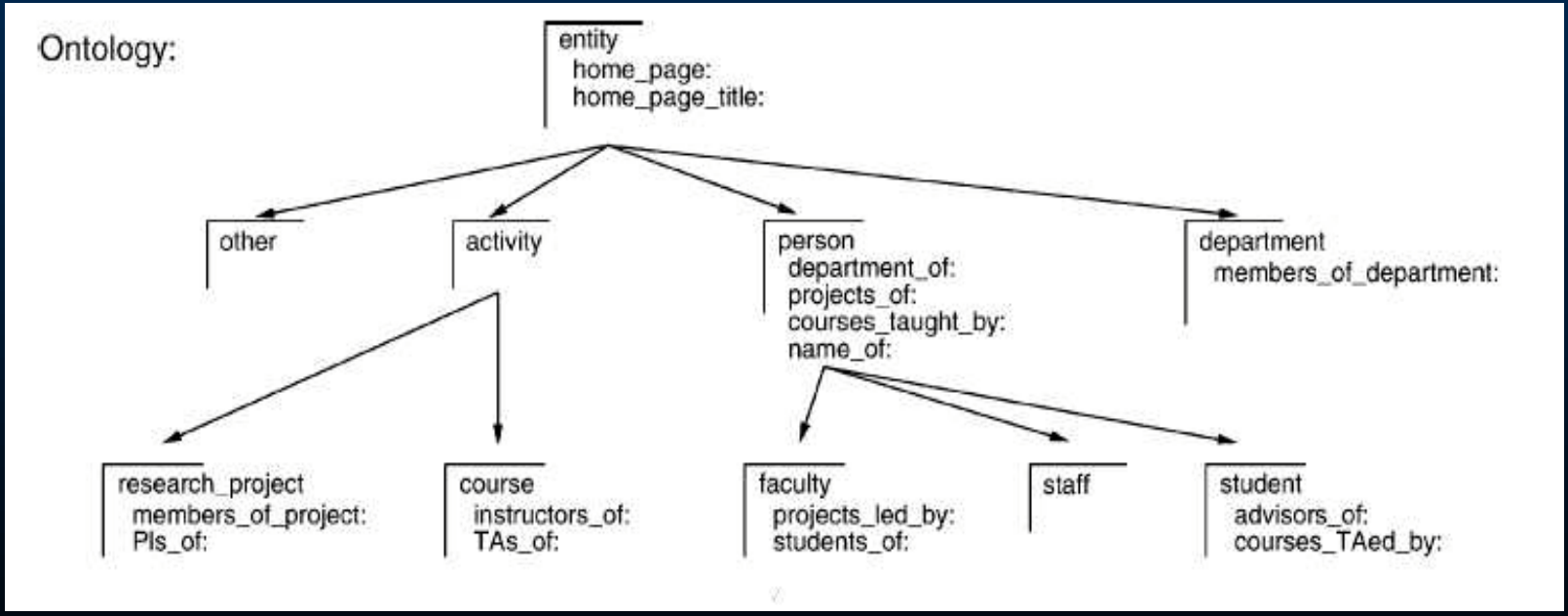
Überblick

Funktion des Web-KB Systems



Überblick
Ontologie

- Ontologie: Klassen und Relationen, die Informatikfachbereichsseiten beschreiben

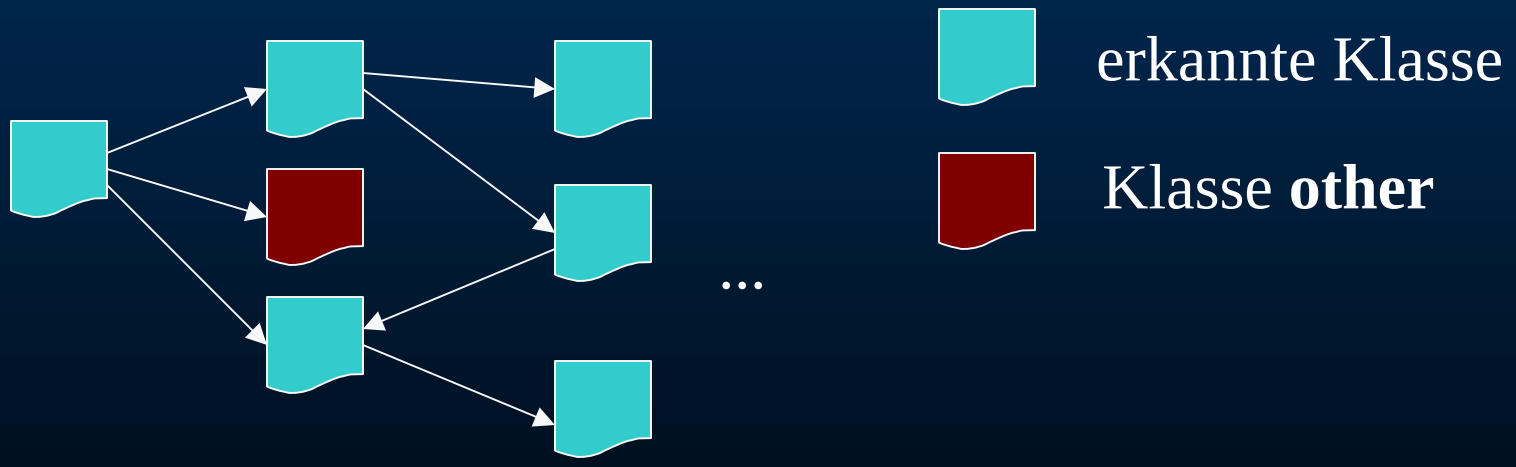


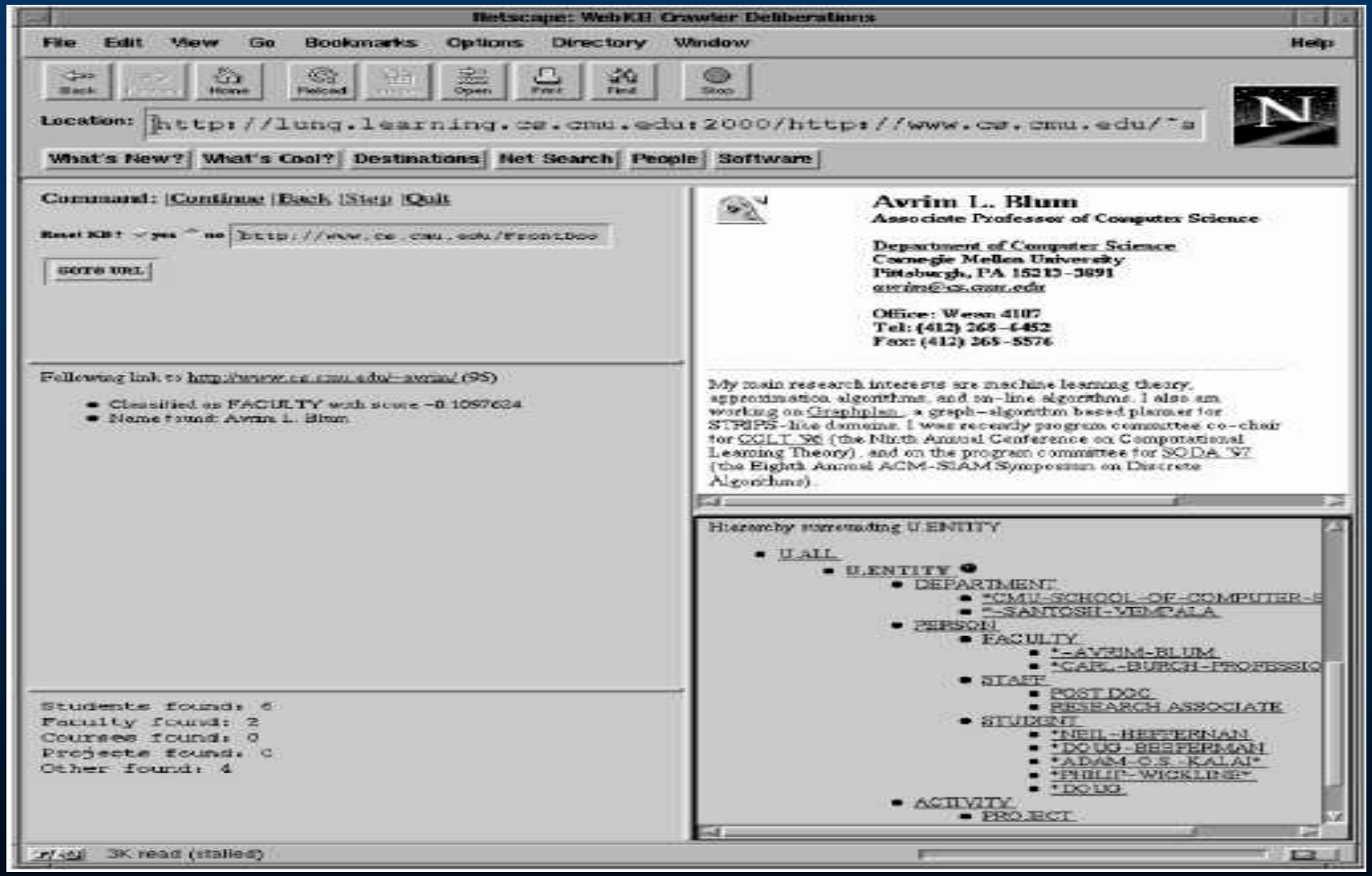
Überblick
Training / Testing

- Training: es werden Erkennungsfunktionen zur Extraktion gelernt (ML). Trainingsmenge: gelabelte Webseiten von 4 Fachbereichen an versch. Universitäten (8000 Seiten + 1400 Seitenpaare)
- Testing: System durchsucht selbsttätig das Web und extrahiert neue Instanzen von Klassen und Relationen (4127 Seiten + 10945 Hyperlinks)

Überblick
Anwendung

- Breitensuche auf dem Graphen aus Seiten und Hyperlinks:





Das Webinterface des vorgestellten Prototyps

Problemspezifikation

- gegeben:
  - initiale Wissensbasis bestehend aus Ontologie (und optional Instanzen)
  - Trainingsbeispiele, die Instanzen von Klassen und Relationen beschreiben
- gesucht:
  - Erkennungsfunktionen zur selbständigen Extraktion von Instanzen aus dem Web (Erweiterung der Wissensbasis)

Problemspezifikation

- Annahmen:
  - Klassen werden durch Hypertextsegmente beschrieben (1 Textabschnitt, 1 Seite oder zusammenhängende Seiten), hier: nur 1 Seite
  - Relationen werden durch Hyperlinkpfade beschrieben (Weg aus Seiten+Hyperlinks) oder durch Textsegmente oder durch andere gelernte Regeln, hier: nur binäre Relationen

Problemspezifikation

- offene Aufgaben
  - Erkennen von Klasseninstanzen
  - Erkennen von Relationsinstanzen
  - Extraktion aus Textsegmenten

Erkennen von Klasseninstanzen
statistische Textklassifikation

- Classifier
  - *full-text*
  - *title / heading*
  - *hyperlink*

Erkennen von Klasseninstanzen
statistische Textklassifikation

- Bewertung
  - probabilistisches Model (*unigram*, *bag-of-words*): Wörter sind unabhängig voneinenander
  - ordne Seite der Klasse zu, die - bzgl. der vorkommenden Wörter - am wahrscheinlichsten ist

student		faculty		course	
my	0.0247	*DDDD*	0.0138	course	0.0151
page	0.0109	of	0.0113	*DD:DD*	0.0130
home	0.0104	and	0.0109	homework	0.0106
am	0.0085	professor	0.0088	will	0.0088
university	0.0061	computer	0.0073	*D*	0.0080
computer	0.0060	research	0.0060	assignments	0.0079
science	0.0059	science	0.0057	class	0.0073
me	0.0058	university	0.0049	hours	0.0059
at	0.0049	*DDD*	0.0042	assignment	0.0058
here	0.0046	systems	0.0042	due	0.0058

Top-Ten Wörter

Erkennen von Klasseninstanzen

statistische Textklassifikation

Verwendete Score-Funktion:

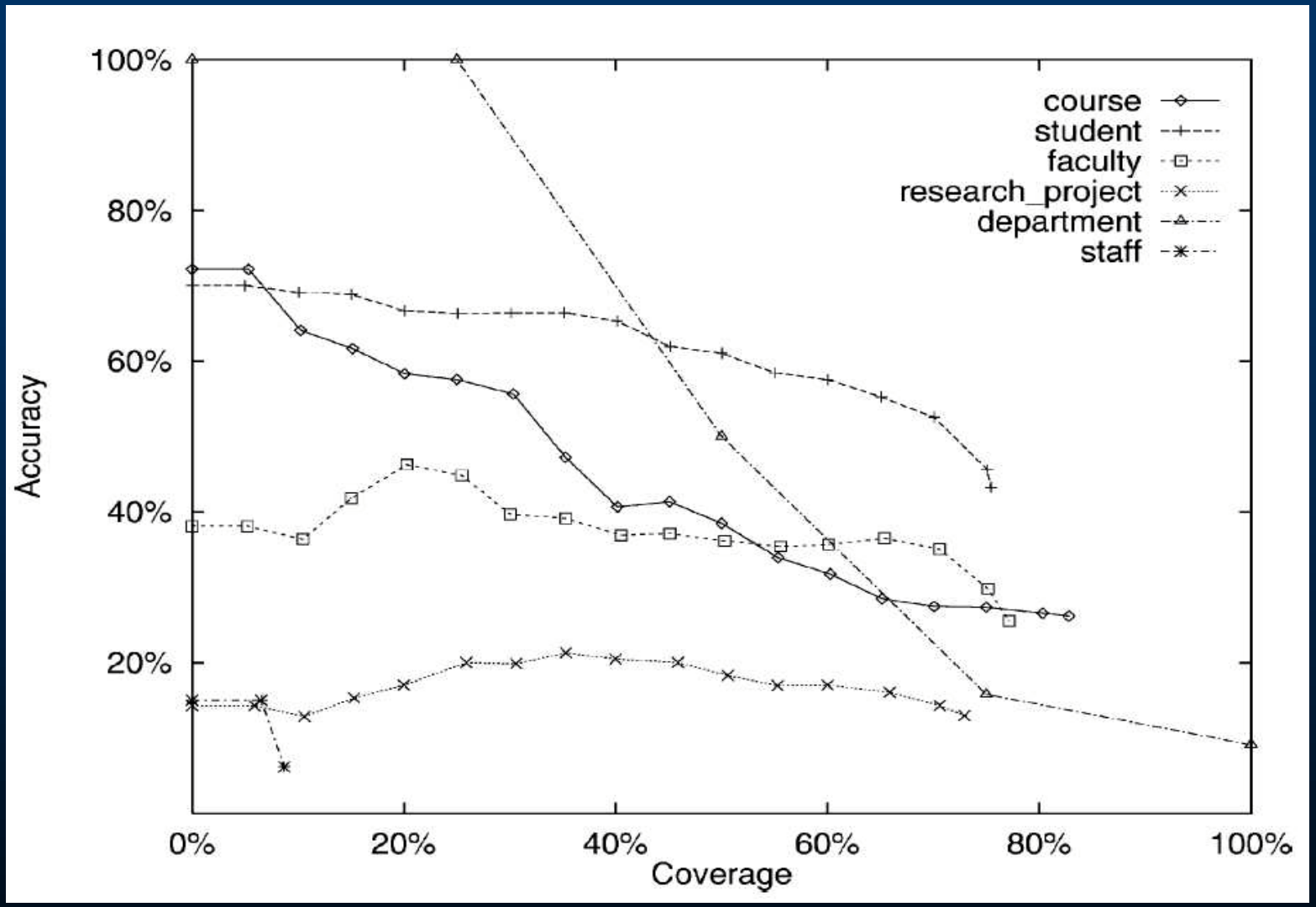
$$Score_c(d) = \frac{\log \Pr(c)}{n} + \sum_{i=1}^{T} \Pr(w_i|d) \log\left(\frac{\Pr(w_i|c)}{\Pr(w_i|d)}\right)$$

enstanden aus:

$$\Pr(c|w_1, \ldots, w_n) \simeq \Pr(c) \prod_{i=1}^{n} \Pr(w_i|c).$$

	Actual							
Predicted	course	student	faculty	staff	research_project	department	other	**Accuracy**
course	**202**	17	0	0	1	0	552	26.2
student	0	**421**	14	17	2	0	519	43.3
faculty	5	56	**118**	16	3	0	264	17.9
staff	0	15	1	**4**	0	0	45	6.2
research_project	8	9	10	5	**62**	0	384	13.0
department	10	8	3	1	5	**4**	209	1.7
other	19	32	7	3	12	0	**1064**	93.6
Coverage	82.8	75.4	77.1	8.7	72.9	100.0	35.0	

Konfusionsmatrix (*full-text*)



Accuracy / Coverage Tradeoff (*full-text*)

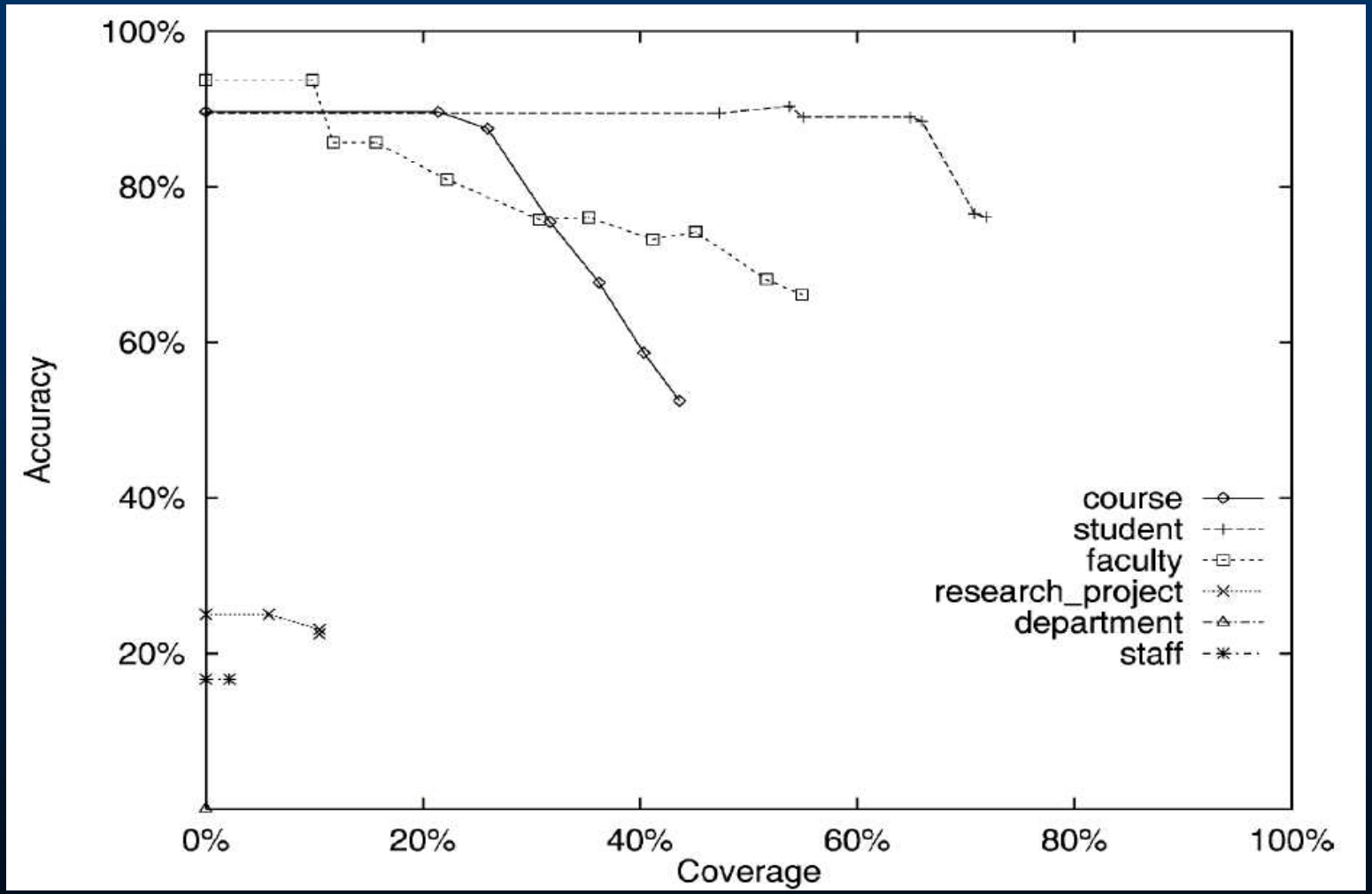
Erkennen von Klasseninstanzen
logische Klassifikation

- Classifier werden als logische Regeln (Klauseln) gelernt
- FOIL: beginne mit leerer Klausel und füge per hill-climbing Suche neue Literale hinzu bis die Regel nur noch positive Instanzen abdeckt

Erkennen von Klasseninstanzen
logische Klassifikation

- Hintergrundrelationen, z.B.
  - has_*word*(Page)
  - link_to(Page, Page)
- Beispiel-Regel:

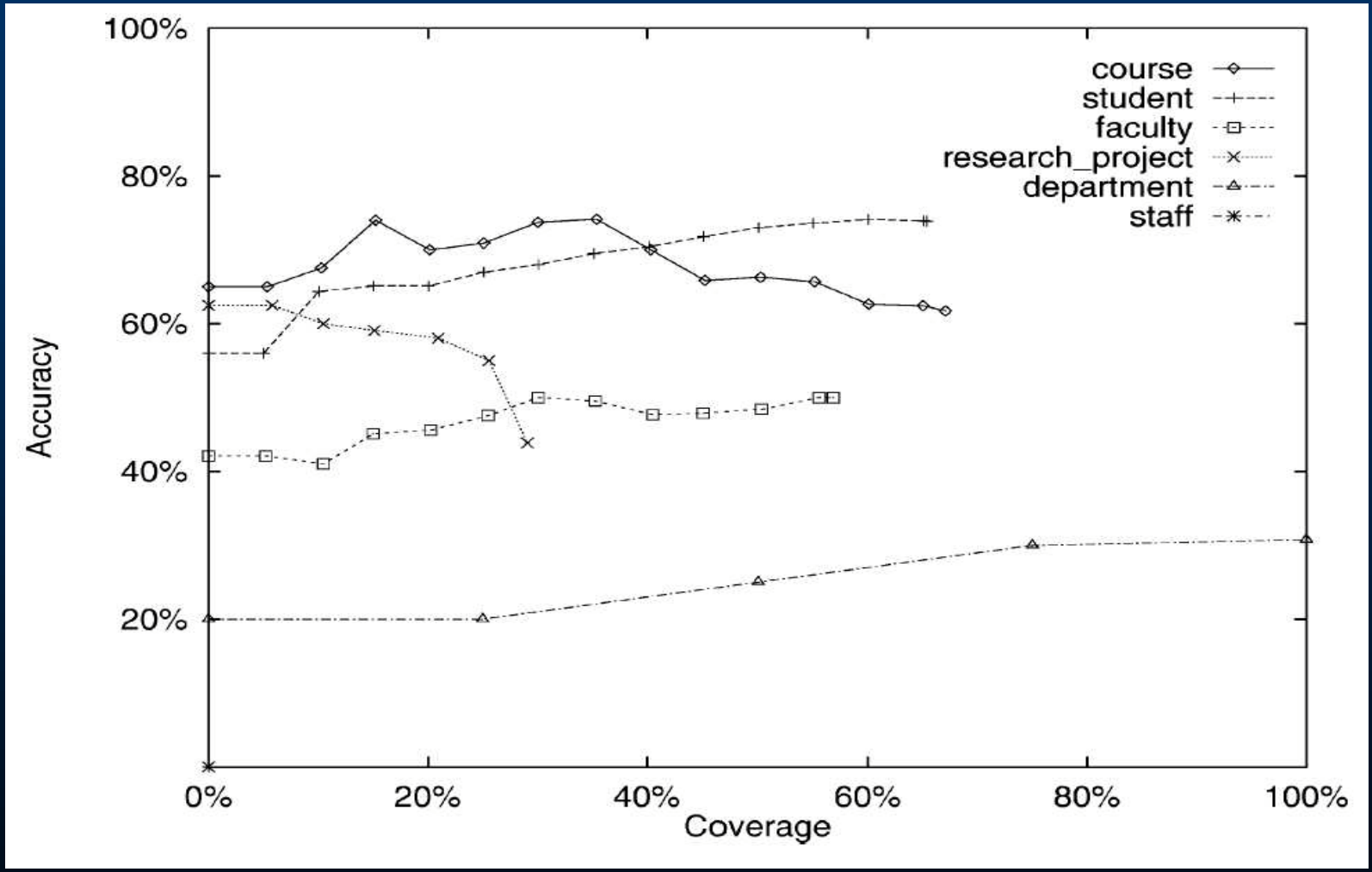
```
course(A) :- has_instructor(A), not(has_good(A)), link_to(A,B), not(link_to(B,_1)),
             has_assign(B).
Training Set: 31 Pos, 0 Neg; Test Set: 31 Pos, 3 Neg
```



Accuracy / Coverage Tradeoff (logische Regeln)

Erkennen von Klasseninstanzen
kombinierte Klassifikation

- benutze alle 4 bisherigen Classfier (*full-text, title/heading, hyperlink, logische Regeln*) und nehme die Klasse mit den meisten Stimmen (bei Stimmgleichheit entscheidet die Zuverlässigkeit der Vorhersagen)
- Ergebnis etwas deprimierend (keine einheitliche Verbesserung i.V. zu einzelnen Classifiern), zu einfache Kombination?



Accuracy / Coverage Tradeoff bei kombinierten Classifiern (200 Vokabeln)

Erkennen von Relationsinstanzen

- Hyperlinkpfade
- Hintergrundrelationen
  - *class*(Page)
  - link_to(Hyperlink, Page, Page)
  - ...

Erkennen von Relationsinstanzen

- Algo so ähnlich wie FOIL:
  (1) Pfadteil der Klausel wird gelernt
  (2) Literale werden hinzugefügt (hill-climbing)

Input: training set of negative and uncovered positive instances

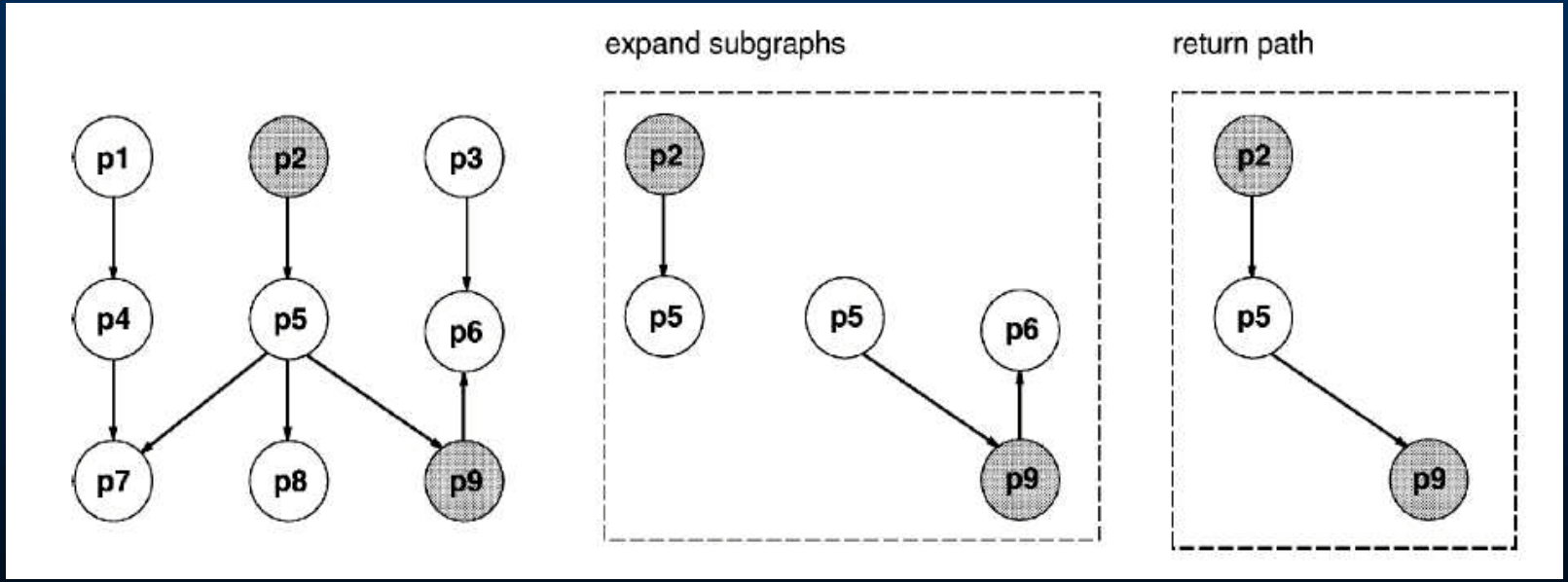
(1)
1. for each uncovered positive instance
2. find a path (up to bounded length) using the background relations
3. select the most common path prototype for which clause search hasn't yet failed
4. generalize the path into an initial clause

(2)
5. do hill-climbing to refine the clause
6. if hill-climbing fails to find an acceptable clause, backtrack to step 3.

Return: learned clause

Erkennen von Relationsinstanzen

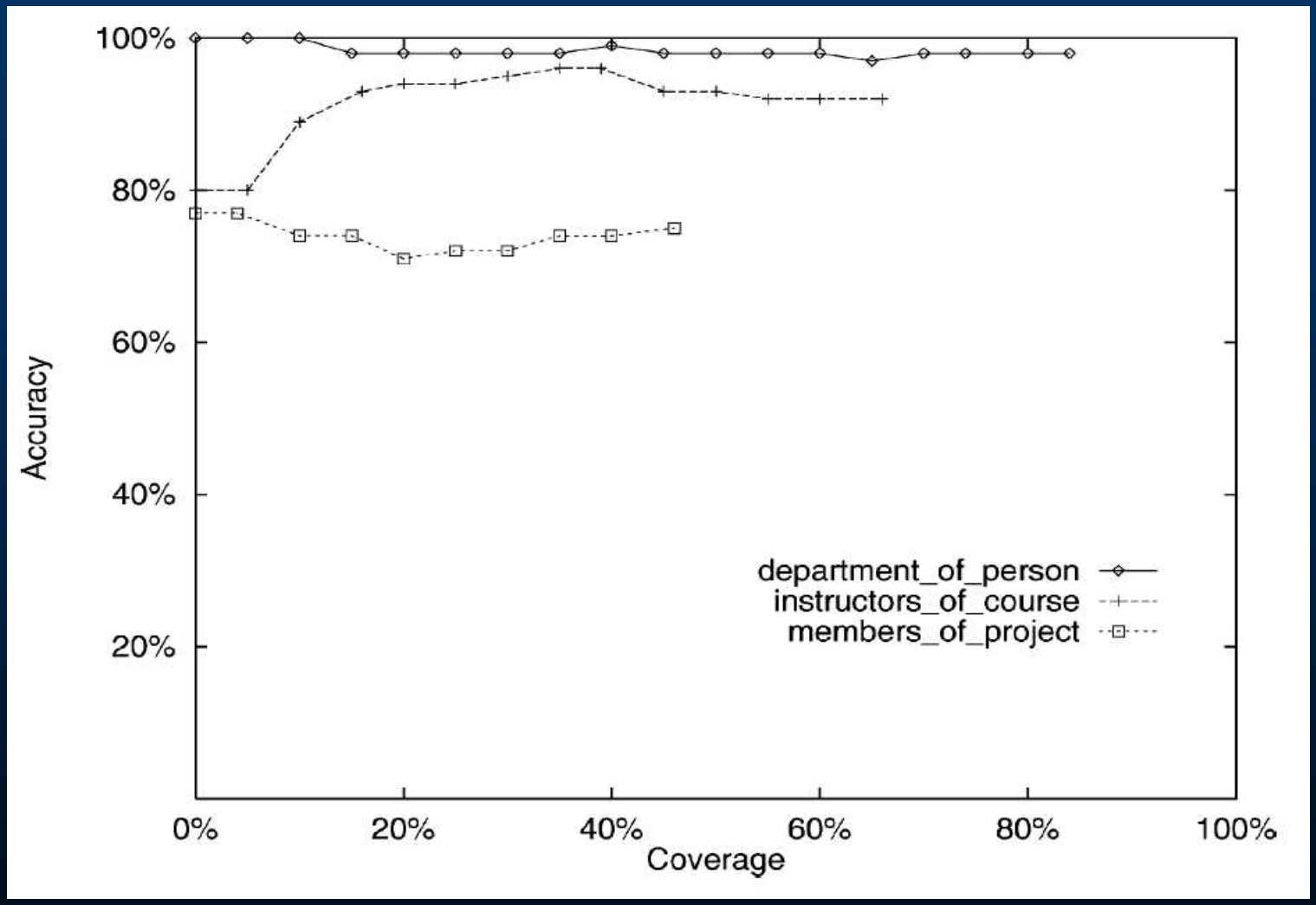
- Pfadfinder bei der Arbeit:



Erkennen von Relationsinstanzen

- Beispiel:

```
members_of_project(A,B) :-  research_project(A), person(B), link_to(C,A,D),
                            link_to(E,D,B), has_neighborhood_word_people(C).
Test Set: 18 Pos, 0 Neg
```



Accuracy / Coverage Tradeoff bei gelernten Relationen

Extraktion aus Textsegmenten

- typische IE - Aufgabe
- IDL Lerner im Sinne FOILs: SRV (Sequence Rule with Validation)
- Eingabe für SRV:
  - gelabelte Seiten (bzgl. der gesuchten Instanzen)
  - Features
- Ausgabe: IE Regeln

Extraktion aus Textsegmenten

- hill-climbing: Füge ein Literal hinzu, das soviel positive Instanzen (gelabelte Segmente) wie möglich abdeckt und viele negative Instanzen (ungelabelte Segmente) ausschliesst
- beende Suche, wenn Regel gut genug ist

Extraktion aus Textsegmenten

- Beispiel:

```
ownername(Fragment) :- some(B, [ ], in_title, true),
                       length(<, 3),
                       some(B, [prev_token], word, "gmt"),
                       some(A, [ ], longp, true),
                       some(B, [ ], word, unknown),
                       some(B, [ ], quadrupletonp, false)
```

Extraktion aus Textsegmenten

- Erkannter Name:

```
Last-Modified: Wednesday, 26-Jun-96 01:37:46 GMT

<title> Bruce Randall  Donald</title>

<h1>
<img src="ftp://ftp.cs.cornell.edu/pub/brd/images/brd.gif">
<p>
Bruce Randall Donald<br>
Associate Professor<br>
```

- Regel trotzdem gut, Accuracy bei 77%

Fazit

- Prototyp hat Accuracy von über 70% bei Coverage von ca. 30%
- Methoden lassen sich vermutlich in neuen Kombination in anderen Domänen anwenden
- Aber: geignetste Methode hängt von der Darstellung der Instanzen ab

Fazit

- Prototyp gut an Webseiten angepasst:
  - Sicht nach 'Außen': Betrachtung der Umgebung von Webseiten (logische Regeln)
  - Sicht nach 'Innen': Betrachtung einzelner Textabschnitte von Webseiten (SRV-Textextraktion)

Ausblick

- viele potentielle Verbesserungen
  - hierarchische Relation zwischen Klassen
  - kombiniere ungelabelte und gelabelte Trainingsmenge
  - mehr linguistische Struktur
  - multiple Strategien zum Extrahieren von Textsegmenten
  - ...

Links

- http://www.biostat.wisc.edu/~craven/papers/aij00.ps
- http://www.cs.cmu.edu/~webkb/

Erkennen von Klasseninstanzen

statistische Textklassifikation

- Wortwahrscheinlichkeiten (Abschätzung)

$$\Pr(w_i|c) = \begin{cases} \dfrac{N(w_i,c)}{T_c + \sum_j N(w_j,c)}, & \text{if } N(w_i,c) \neq 0, \\ \dfrac{T}{T_c + \sum_j N(w_j,c)} \dfrac{1}{T - T_c}, & \text{if } N(w_i,c) = 0, \end{cases}$$