BibTeX Bibliography Index Maker

(Bibim)

Primavera 2010

Índex

Objectius

Motivació

Guió típic d'un treball de recerca:

1. Lectura d'articles

2. S'acumulen els fitxers (PDF) en un directori

3. A l'hora d'escriure, fa falta tenir un índex bibliogràfic per poder citar

Objectius

Exemple d'un article

Closed and Maximal Tree Mining Using Natural Representations

José L. Balcázar balqui@lsi.upc.edu
Albert Bifet abifet@lsi.upc.edu
Antoni Lozano antoni@lsi.upc.edu
Universitat Politècnica de Catalunya, Departament de Llenguatges i Sistemes Informàtics

1. Introduction

Mining frequent trees is becoming an important task, with broad applications including chemical informatics, computer vision, text retrieval, bioinformatics, and Web analysis. Many link-based structures may be studied formally by means of unordered trees [...]

the closed graph patterns discovered.

In the case of trees, there are two broad kinds of subtrees considered in the literature: subtrees which are just induced subgraphs, called induced subtrees, and subtrees where contraction of edges is allowed, called embedded subtrees[...]

Objectius

Exemple d'un article

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Discrete Applied Mathematics 147 (2005) 43–55
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Pseudo-models and propositional Horn inference

Bernhard Ganter*, Rüdiger Krauße

Institut für Algebra, Technische Universität Dresden, Zellescher Weg 12-14, D-01062 Dresden, Germany

Received 4 May 2001; received in revised form 6 January 2003; accepted 21 June 2004
Available online 29 December 2004

Abstract

A well-known result is that the inference problem for propositional Horn formulae can be solved in linear time. We show that this remains true even in the presence of arbitrary (static) propositional background knowledge. Our main tool is the notion of a cumulated clause, a slight generalization of the usual clauses in Propositional Logic. We show that each propositional theory has a canonical irredundant base of cumulated clauses, and present an algorithm to compute this base.

Objectius

Exemple de referència

```
@article{1063619,
  author =
    {Ganter, Bernhard and Krau\sse, R\"{u}diger},

  title =
    {Pseudo-models and propositional Horn inference},

  journal =
    {Discrete Appl. Math.},

  volume =
    {147},

  year =
    {2005},
  ...
 }
```

Objectius

Objectiu

Un únic objectiu principal:

- La creació d'índexs bibliogràfics de forma automàtica

Però per aconseguir-lo necessitarem:

- Cercar referències a Internet
  (Biblioteques digitals)
- Extreure la informació de les pàgines HTML
- Crear regles d'extracció automàticament

Objectius

Objectiu

Un únic objectiu principal:

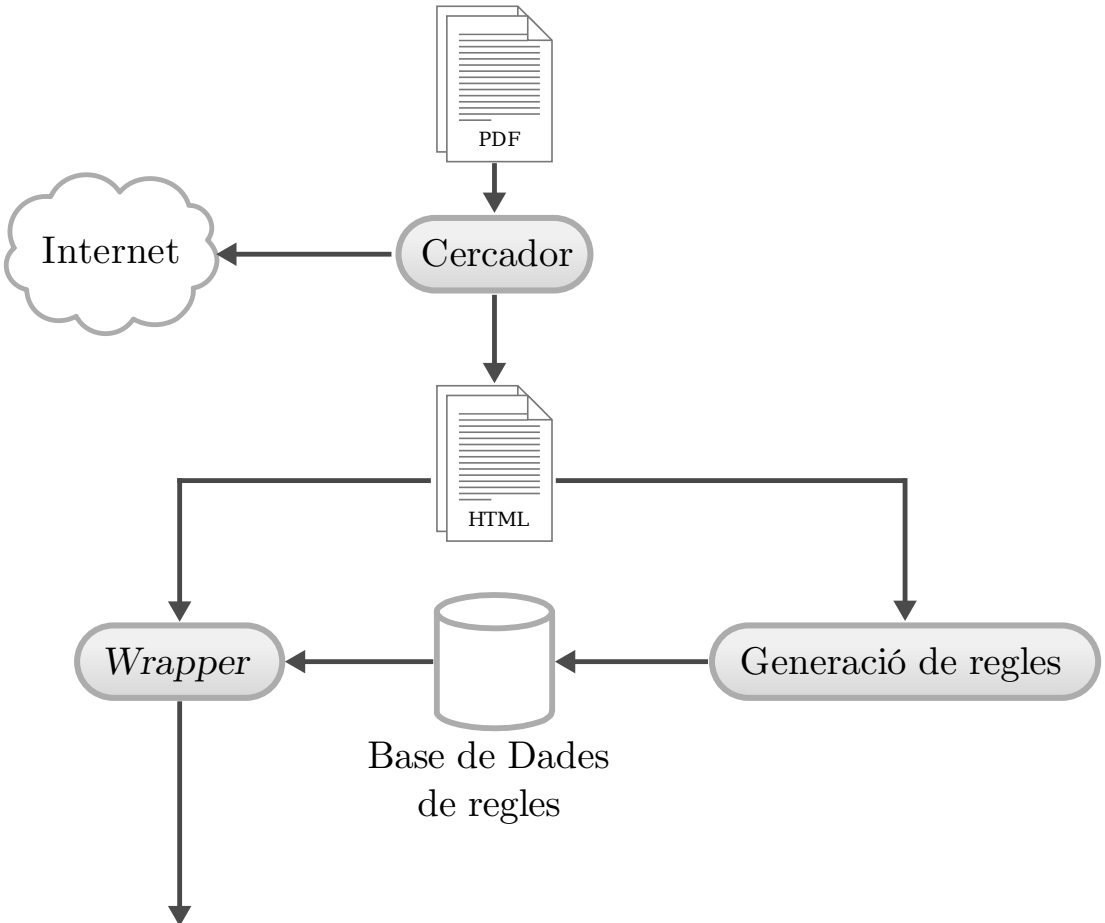
- La creació d'índexs bibliogràfics de forma automàtica

Però per aconseguir-lo necessitarem:

- Cercar referències a Internet
  (Biblioteques digitals)
- Extreure la informació de les pàgines HTML
- Crear regles d'extracció automàticament

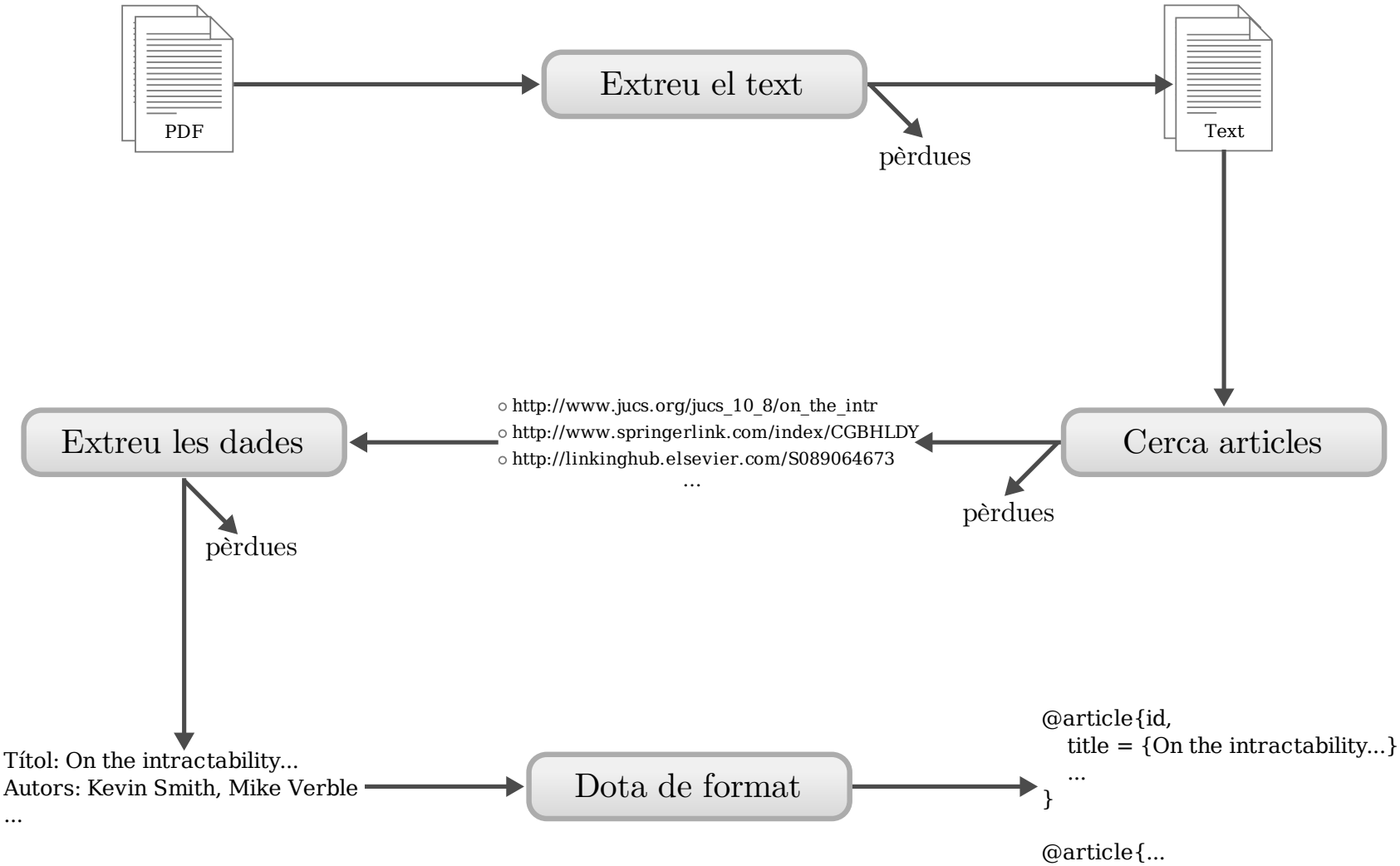
Objectius

Esquema del sistema



Extracció de referències

Esquema d'extracció



Extracció de referències

PDF to text

Quant a l'extracció de text dels fitxers PDFs:

- S'utilitza l'eina *xPDF*
- Resultats relativament bons...
- ...però continua tenint força problemes

Extracció de referències

Cerca de referències

Passos a seguir per cercar referències:

- Genera consultes a partir del text
  `([\w()?!]+\ ){min,max}`
  "and slurry methods and have been tested in the selective"

- Obté els resultats de les consultes (*Google*, *Bing*, *Yahoo*)
  - `www.springerlink.com/index/G4588X...`
  - `www.ingentaconnect.com/content/klu/ca...`

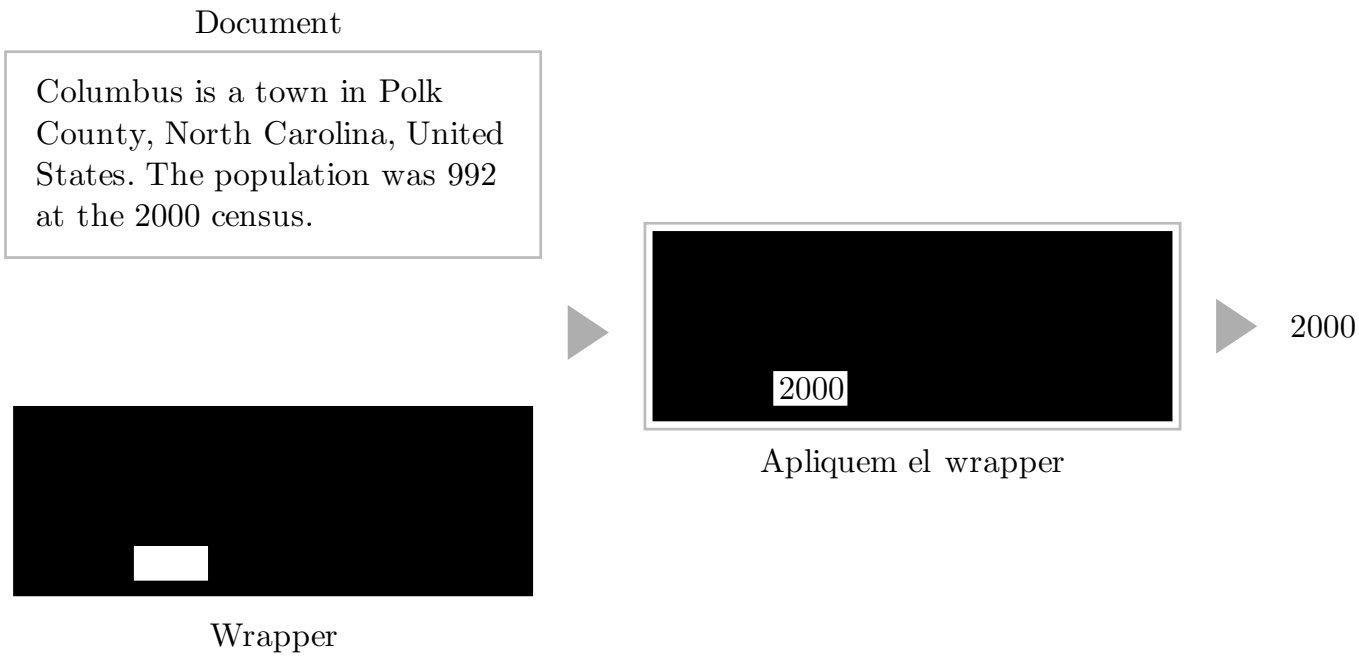
- Filtra i ordena els resultats

Extracció de referències

Extracció d'informació

Dos tipus de regles d'extracció d'informació:

- *Reference Wrappers*
- *Field Wrappers*

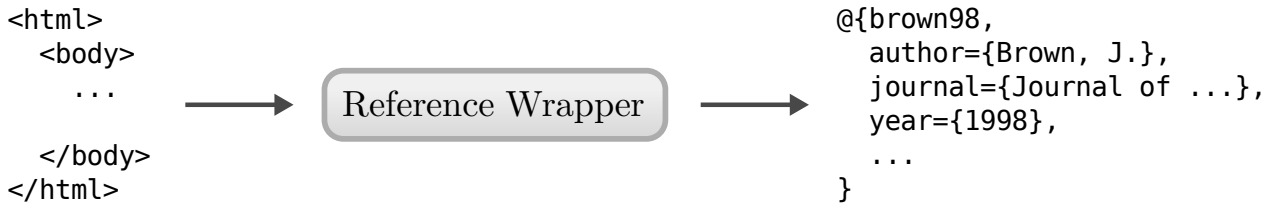


Extracció de referències

Reference Wrappers

Característiques:

- Extreuen referències senceres (BIBTEX)
- Només se'n necessita un per cada biblioteca digital
- Els resultats solen ser bons
- Implementats manualment

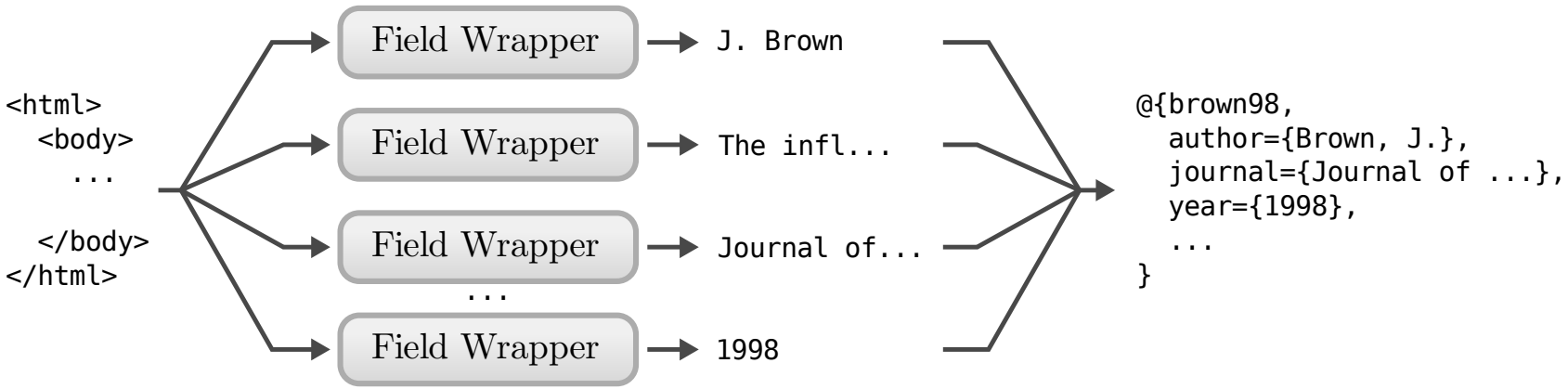


Extracció de referències

Field Wrappers

Característiques:

- Especialitzats en extreure un sol camp
- Se'n necessita un per cada camp i per cada biblioteca
- Implementats automàticament



Extracció de referències

Validació

Per què:

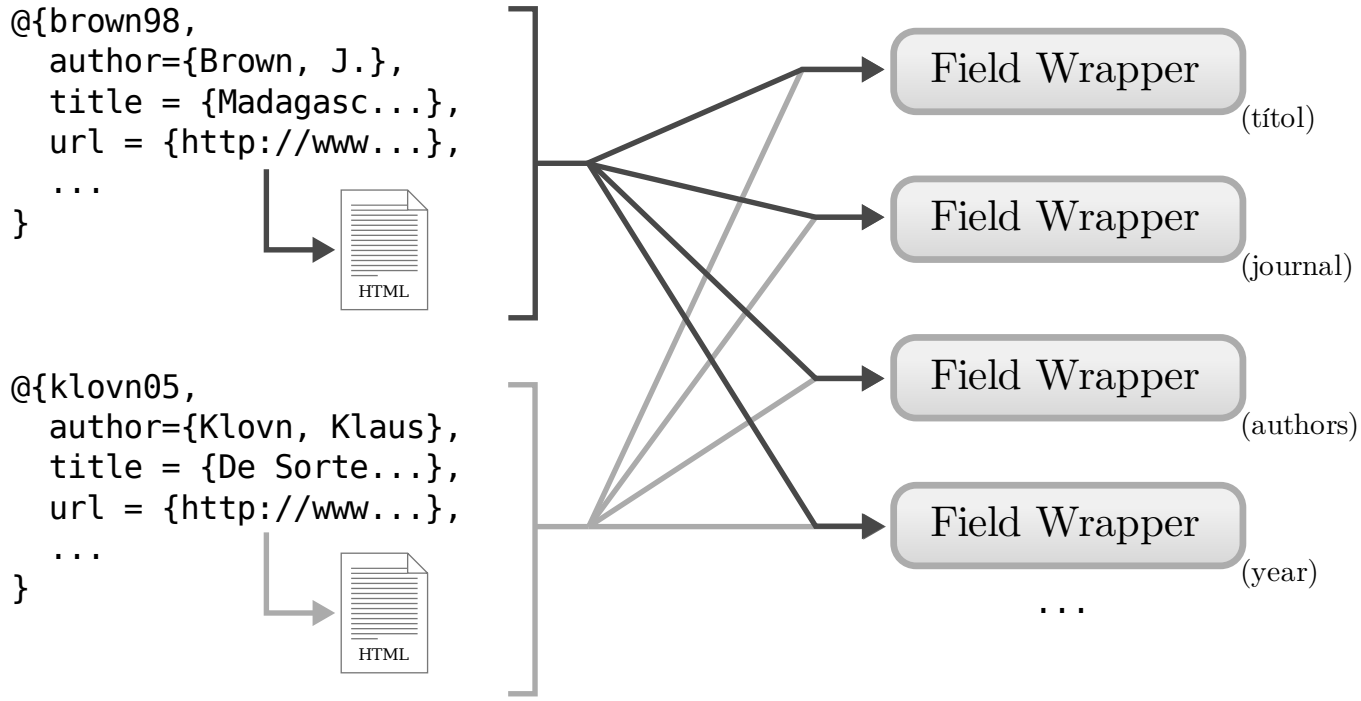
- Informar a l'usuari
- Utilitzar les referències per re-generar regles

Com ho fem:

- Camps com ara el títol i autors: es comprova que es troben dins del document PDF
- Camps com ara el número de pàgines o any: es mira que compleixin una expressió regular.
- Cada camp té un pes sobre la validesa final

Generació de *wrappers*

Esquema



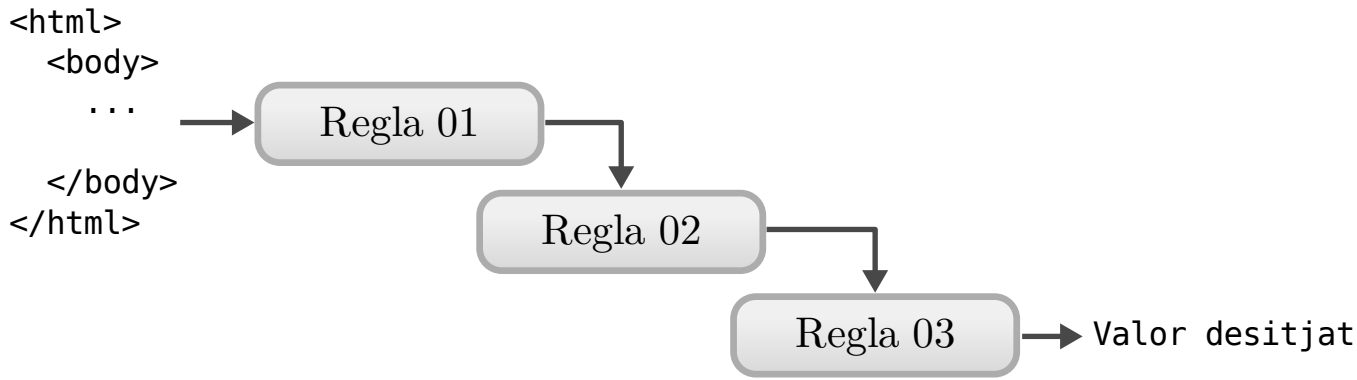
- S'utilitzen (poques) referències com a exemples.
- Corresponen a la mateixa biblioteca digital

Generació de *wrappers*

Estructura dels *Field Wrappers*

Els *field wrappers* es caracteritzen per:

- Consisteixen en una seqüència ordenada de regles
- Cada regla està formada per un procediment guiat per un patró
- Les regles estan connectades en cascada

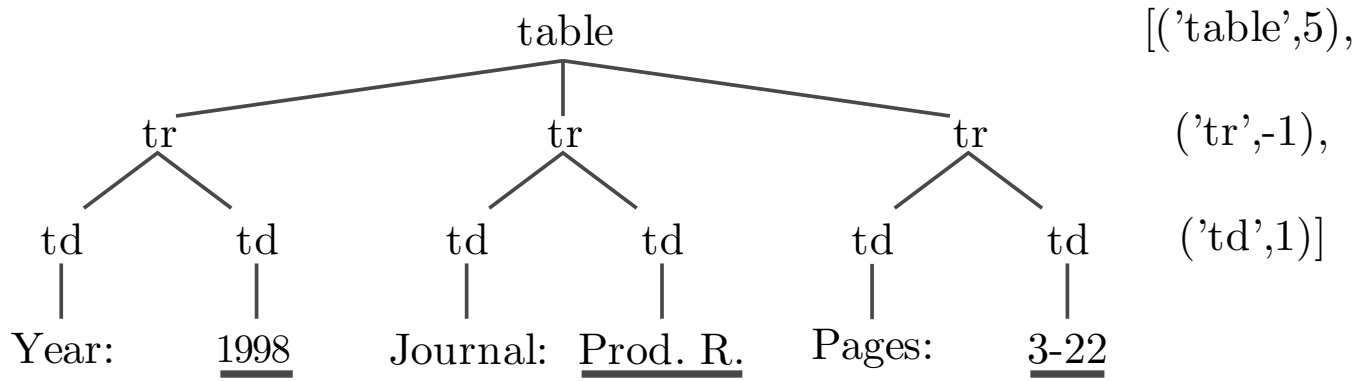


Generació de *wrappers*

Tipus de regles

Path Rule:

- Localitza elements del document HTML
- El patró consisteix en una ruta dins de l'arbre HTML



Generació de *wrappers*

Tipus de regles

Regex Rule:

- Localitza valors dins d'una cadena de caràcters.
- El patró és una expressió regular per extreure el valor desitjat

Text: `Vol. 27, pp. 1204–1209.`
Patró: `(?:.*), pp. (.*).`

Resultat: `1204–1209`

Generació de *wrappers*

Tipus de regles

Exemple de generació d'un patró:

- Dos elements de pàgines HTML d'exemple:

```
Vol.  27, pp.  1204–1209.
Vol.  14, pp.  33–43.
```

- Extraiem la informació que ens interessa:

```
Vol.  27, pp.  (.*).
Vol.  14, pp.  (.*).
```

- Obtenim la similaritat de les dues cadenes:

```
0.88
```

- Fusionem els blocs de caràcters no coincidents:

```
Vol.  (?:.*), pp.  (.*).
```

Generació de *wrappers*

Tipus de regles

Exemple de generació d'un patró:

- Dos elements de pàgines HTML d'exemple:

```
Vol.  27, pp.  1204–1209.
Vol.  14, pp.  33–43.
```

- Extraiem la informació que ens interessa:

```
Vol.  27, pp.  (.*).
Vol.  14, pp.  (.*).
```

- Obtenim la similaritat de les dues cadenes:

```
0.88
```

- Fusionem els blocs de caràcters no coincidents:

```
Vol.  (?:.*), pp.  (.*).
```

Generació de *wrappers*

Tipus de regles

Exemple de generació d'un patró:

- Dos elements de pàgines HTML d'exemple:
  ```
  Vol.  27, pp.  1204–1209.
  Vol.  14, pp.  33–43.
  ```

- Extraiem la informació que ens interessa:
  ```
  Vol.  27, pp.  (.*).
  Vol.  14, pp.  (.*).
  ```

- Obtenim la similaritat de les dues cadenes:
  ```
  0.88
  ```

- Fusionem els blocs de caràcters no coincidents:
  ```
  Vol.  (?:.*), pp.  (.*).
  ```

Generació de *wrappers*

Tipus de regles

Exemple de generació d'un patró:

- Dos elements de pàgines HTML d'exemple:

```
Vol.  27, pp.  1204–1209.
Vol.  14, pp.  33–43.
```

- Extraiem la informació que ens interessa:

```
Vol.  27, pp.  (.*).
Vol.  14, pp.  (.*).
```

- Obtenim la similaritat de les dues cadenes:

```
0.88
```

- Fusionem els blocs de caràcters no coincidents:

```
Vol.  (?:.*), pp.  (.*).
```

Generació de *wrappers*

Altres regles

Separators Regex Rule:

- Separa valors continguts en un mateix text
- El patró és un conjunt de cadenes que actuen de separadors

Text: **`Value A, Value B and Value C`**
Patró: `“, ”` i `“ and ”`

Resultat: **`[Value A, Value B, Value C]`**

Generació de *wrappers*

Altres regles

MultiValue Regex Rule:

- *Regex rule* que actua sobre diferents valors

Person Rule:

- Separen noms en els camps: `last_name`, `middle_name` i `first_name`. Sense patró.

Nom: **`Anders And`**

Resultat: **`["And", "","Anders"]`**

Generació de *wrappers*

Altres regles

MultiValue Regex Rule:

- *Regex rule* que actua sobre diferents valors

Person Rule:

- Separen noms en els camps: `last_name`, `middle_name` i `first_name`. Sense patró.

  Nom: **`Anders And`**

  Resultat: **`["And", "","Anders"]`**

Generació de *wrappers*

Avaluació de *Wrappers*

Quant a l'avaluació de *wrappers*:

- Els millors *wrappers* s'han de provar primer
- Per escollir un ordre inicial, s'apliquen sobre els mateixos exemples
- Cada vegada que s'utilitza un *wrapper* se li dóna un vot
- Els bons pugen i els dolents baixen

Demo

Què veurem:

- Extracció utilitzant només *Reference Wrappers*
- Importació de referències
- Generació de *Field Wrappers*
- Extracció utilitzant tots els *wrappers*
- Correcció manual de regles
- Extracció amb les regles corregides

Tasques i Estimació Temporal

Tasca	**Temps estimat**
Recerca i proves de concepte	20 dies
Configuració de la infraestructura	4 dies
Extracció del text dels PDF	8 dies
Obtenció de pàgines amb la referència	13.5 dies
Extracció d'informació	14 dies
Generació automàtica de *wrappers*	32.5 dies
Tractament de referències	5 dies
Base de dades	5 dies
Mòdul principal	11 dies
Interfície d'usuari	4 dies
Proves	12 dies

Conclusions

- Pèrdues a cada pas
- L'èxit depèn completament de la qualitat dels *wrappers* disponibles
- Errors fàcilment corregibles
- La part de generació i extracció, aplicable a qualsevol altre domini
- Treball futur: afegir millores i noves funcionalitats

Q&A