



Information Extraction

CS 510 Winter 2007

1



Topics

- History
- IE vs. IR
- Evaluation metrics
- Wrappers
 - Construction tools
 - Automatic generation

CS 510 Winter 2007

2



Some questions

- What SONY digital cameras sell for under \$500?
- What are parasites of northern flying squirrels?
- What are titles of books that talk about PSOCOPTERA?

CS 510 Winter 2007

3



Applying IR to the questions

- Could IR techniques distinguish web pages relevant to Q1?
- Can IR techniques use the third document to answer Q2 and Q3?

CS 510 Winter 2007

4



Need something in addition to IR

Would like to do “databasey” stuff with the info.

- Selection: `price < $500`
- Sorting, Aggregation: `group parasites by host`
- Join: `taxonomic orders with publications by author`

CS 510 Winter 2007

5



Complementary technologies

- IR retrieves relevant documents
- IE extract relevant information from documents

*Information extraction from World Wide Web
-- a survey, Line Eikvil, July 1999.*

Extracting text vs. extracting structured or semi-structured data

CS 510 Winter 2007

6



History

- Roots in Natural Language community
- Message Understanding Conferences (MUCs)

Look at small documents (email, news) and extract facts

- joint ventures
- company management changes

Mainly using NL approaches

CS 510 Winter 2007

7



Evaluation metrics

- Similar to IR, but one twist: IR doesn't "make up" documents, but IE might "make up" facts.

- Relevant = relevant + correct

- Precision:

`#relevant facts/#facts returned`

- Recall:

`#relevant facts/#total possible correct`

CS 510 Winter 2007

8

Influence of the Web

- Semi-structured text: between free text and rigidly formatted documents
Can often make IE easier

Ideal



CS 510 Winter 2007

9

Issues

1. Document not broken into info elements
2. They aren't in the DB's data model
3. Database model too rigid for web content
4. Database query language not flexible enough

Many of these problems go away with web services

CS 510 Winter 2007

10

For issues 1 & 2

- Create *wrappers* to parse web pages and structure into appropriate model
 - many sites, change often
 - Do you want to write 10,000 parsers by hand?
 - wrapper construction tools
 - automated wrapper generation

CS 510 Winter 2007

11

Wrapper Function



CS 510 Winter 2007

12



For issues 3 & 4

- A lot of variation within and between documents
 - some references have # of pages, some have page numbers
 - one has reprint info
 - some annotated with taxonomic info (?)
 - nested data structures
 - `Androlaelaps casilis`
`{mole, shrew, deermouse, flying squirrel}`

CS 510 Winter 2007

13



Semi-structured data models

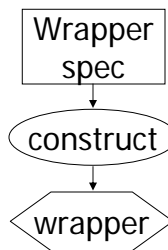
- Self-describing
 - Flexible structure
 - Repetition and nesting
- XML is the current prime example; OEM and ACEdb are others.

CS 510 Winter 2007

14

Wrapper approaches

- Hand coding
- Construction tools

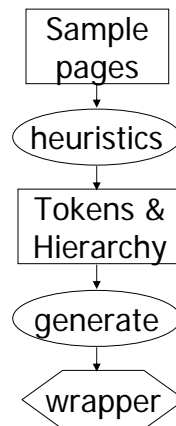


CS 510 Winter 2007

15

Wrapper approaches 2

- Heuristic tools

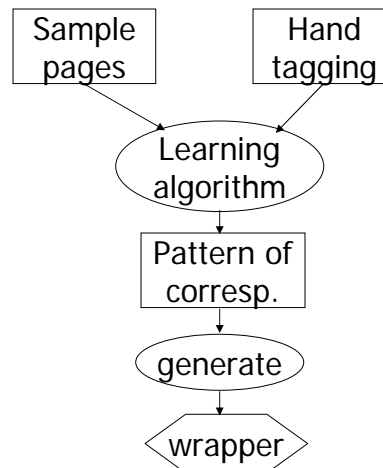


CS 510 Winter 2007

16

Wrapper approaches 3

- Learning tools



CS 510 Winter 2007

17

Example of construction tool

- Extracting structured information from the web, J. Hammer, et al.
<http://www-db.stanford.edu/pub/1997-38>
- Specification for HTML parser and translators
- Part of TSIMMIS project at Stanford and IBM

CS 510 Winter 2007

18



Intellicast European weather

		Tue. 28 Jan	Wed. 29 Jan
country	city	hi/lo	hi/lo
Austria	<u>Vienna</u>	-2/-7	-2/-7
Belgium	<u>Brussels</u>	3/-4	3/-4
Czech Rep.	<u>Prague</u>	-1/-7	-1/-7
England	<u>Birmingham</u>	9/-3	7/3
England	<u>Liverpool</u>	8/2	6/2
...			

CS 510 Winter 2007

19



HTML is a table

```

<HTML>
<HEAD>
<TITLE>INTELLICAST: europe weather</TITLE>
<HR>
<TABLE cellspacing=0 cellpadding=0 width=514>
  <TR align=left>
    <TH colspan=2><BR></TH>
    <TH colspan=1><B>Tue. 28 Jan</B></TH>
    <TH colspan=1><B>Wed. 29 Jan</B></TH>
  </TR>
  <TR align=left>
    <TH><B>country</B></TH>
    <TH><B>city</B></TH>
    <TH><B>hi/lo</B></TH>
    <TH><B>hi/lo</B></TH>
  </TR>

```

CS 510 Winter 2007

20



HTML continued

```

<TR align=left>
  <TD>Austria</TD>
  <TD><A
href=http://www.intellicast.com/vie>Vienna</A></TD>
  <TD>-2/-7</TD>
  <TD>-2/-7</TD>
</TR>
<TR align=left>
  <TD>Belgium</TD>
  <TD><A
href=http://www.intellicast.com/bru>Brussels</A></TD>
  <TD>2/-2</TD>
  <TD>3/-1</TD>
</TR>
...
</TABLE>
</HTML>

```

CS 510 Winter 2007

21



Wrapper specification

- List of rules
 - [variables, source, pattern]
- Variables: fields in final result
 - underscore means temp
 - can be sequences
- Source: previous variable, get(), other operation (split)
- Pattern: text to match
 - * means discard
 - # means keep

CS 510 Winter 2007

22



Sample specification

```
[ [ "root",
    "get('http://www.intellicast.com/europe/')",
    "#"],
  [ "temperatures",
    "root",
    "**<TITLE*<TABLE*</TR>#</TABLE>* " ]
  [ "_citytemp",
    "split(temperatures,'<TR align=left>')",
    "#"]
  [ "city_temp",
    "_city_temp[1:0]",
    "#"]
  [ "country,c_url,city,h_tdy,l_tdy,h_tom,l_tom",
    "city_temp"
    "**<TD>#</TD>*href=#>#</A>*<TD>#</TD>
    <TD>#/#*" ] ]
```

CS 510 Winter 2007

23



Remaining process

- Processed with Python
- Builds an OEM (Object Exchange Model) structure
 - self-describing
 - tuples and sequences
 - can be nested

CS 510 Winter 2007

24



OEM Output

```

root complex {
  temperatures complex {
    city_temp complex {
      country string "Austria"
      c_url url "http://www.int..."
      city string "Vienna"
      h_tdy string "-2"
      l_tdy string "-7"
      h_tom string "-2"
      l_tom string "-7"
    }
    city_temp complex {
      country string "Belgium"
      ...
    }
  }
}

```

CS 510 Winter 2007

25



LOREL – Stanford

- Query language for OEM
- Like SQL + Path expressions

```

select C.city
from root.temperatures.city_temp C
where C.l_tom < 0

```

CS 510 Winter 2007

26



LOREL – paths

- Handle variability, uncertainty in structure
- Optional
`(root)?.temperatures.city_temp.l_tom`
- Regular expressions
`root.(temperatures|weather).city_temp.l_tom`
- Wildcard
`root.#.l_tom`

CS 510 Winter 2007

27



Wrapper generation

- *Wrapper generation for semistructured Internet sources*, N. Ashish and C. Knoblock.
<http://www.isi.edu/sims/naveen/research.html>

CS 510 Winter 2007

28

CIA Factbook page (old style)

France

Geography

Location: Western Europe, bordering the Bay of Biscay and English Channel, between Belgium and Spain southeast of the UK; bordering the Mediterranean Sea, between Italy and Spain

Map references: Europe

Area:

total area: 547,030 sq km

land area: 545,630 sq km

comparative area: slightly more than twice the size of Colorado

note: includes Corsica and the rest of metropolitan France, but excludes

CS 510 Winter 2007

29

Structuring the source: Heuristics

- Finding tokens that begin a section
 - bold tags, headings `<h3></h3>`, sequence with colon, italicized words


Introduction Geography

Location: Area: Land boundaries:

total: land: water:

CS 510 Winter 2007

30



Nesting of sections: Heuristics

- Font size and indentation

```
CIAPage -> Geography People ...
Geography -> Location Map_references Area ...
Area -> total_area land_area comparative_area
```

- Use Lex and YACC to build parser that will find text associated with tokens
- Used with Ariadne integration system