IE Systems

Fang Li
What is IE system?

IE systems extract domain-specific information from natural language text. The domain and types of information to be extracted must be defined in advance. IE systems often focus on object identification, such as references to people, places, companies, and physical objects. [...] Domain-specific extraction patterns (or something similar) are used to identify relevant information.

(Riloff and Lorenzen, 1999, p. 169)

Some **limitations**:  
- Information defined in advance  
- Domain –specific
Types of IE systems introduced

Aim: Extract information from the Internet.

- Wrapper Systems
- NLP based extraction system
- Open-domain extraction system
Wrapper-based IE systems:

- lixTo system *(semi-automatically)*
- Roadrunner system *(automatically)*
- Never-Ending Learning (NELL)
Discussion

Given a page with many data in it

How to extract it automatically?
Lixto System

A system and method for the visual and interactive generation of wrappers for web pages under the supervision of a human developer, for automatically extracting information from Web pages using such wrappers, and for translating the extracted content into XML.

www.lixto.com (the company founded in 2001 as a spin-off of the Vienna Technical University.)
Tools & Middleware

Lixto provides enterprise-class development tools and middleware to rapidly develop maintainable and robust data extraction programmes and to effectively use these applications to gather and process data from the web on a large scale. Technology from Lixto is in particular well-suited to access, augment and deliver content and data from highly dynamic web applications which take advantage of client-side processing technologies such as JavaScript, AJAX and dynamic HTML in general.

Tools
Visual development environment

Middleware
Scalable web data extraction processes

Lixto provides an integrated development environment (IDE) for web data extraction programmes. This framework forms the basis for the Lixto Visual Developer and the Lixto Web Application Testing Suite.

Lixto Middleware enables enterprises to run extremely reliable web data extraction processes. Lixto Middleware is a highly scalable web data extraction infrastructure and supports cloud computing for instance deployment.
Features of Lixto

- Very high expressive power:
  - of defining sophisticated extraction patterns

- Excellent visual support
  - for marking extraction patterns

- Good learnability
  - No extraction language needs to be learned

- Sample parsimony
  - Very few sample pages are needed in order to define robust wrappers

- Simple and smooth XML translation mechanism
Architecture of Lixto System

Interactive Pattern Builder

Extractor (continual)

Logic control

Event

XML Generator / Simple Query System

Transformer

Pattern Instance Base
(hierarchically ordered)

work on XML

ELOG Program

work on XML

Set of structural similar pages

Example set (usually a single page)

User
Interactive pattern builder: provides the visual UI that allows a user to specify the desired extraction patterns and the basic algorithm for creating a corresponding Elog wrapper as output.

Extractor: Elog program interpreter that performs the actual extraction based on a given Elog program.

The controller of XML Generator: the user chooses how to map extracted information to XML.
About extraction language: *Elog*

*Elog*: system-internal datalog-like rule based language specially designed for hierarchical and modular data extraction.

**Datalog Rule:**

\[
\text{Happy}(d) \leftarrow \text{Frequents}(d, \text{bar}) \ \text{AND} \ \text{Likes}(d, \text{beer}) \ \text{AND} \ \text{Sells}(\text{bar}, \text{beer}, p)
\]

**Head predicate:**

\text{Happy}(d)

If and only if all atoms of the body are true, the head is true.
The head of a rule $r$ is of the form $p(S,X)$:

- $p$ is a pattern name,
- $S$ is a variable which is **bound in the body of the rule** to the parent-pattern instances of the filter corresponding to $r$,
- $X$ is the **target variable** which, at extraction time, is bound to some target pattern instance (a tree region or string) to be extracted.
Extraction language: *Elog (cont.*)

A standard extraction rule:
New(S,X) ← Par(_,S), Ex(S,X), Co(S,X,..)[a,b]

Par(_,S): parent pattern predicate
Ex(S,X): extraction definition predicate
Co(S,X,..): further imposed conditions
[a,b] are optional, range parameter.
record($S, X$) ← tableseq($\_\_\_, S$), subelem($S, :table, X$)

If $S$ is an instance in tableseq, and $X$ is a tree region contained in $S$ and the root of $X$ matches table then $X$ is a table contained in $S$.

The first atom: the parent pattern is an instance of <tableseq>. The second atom: looks for subelements that qualify as tables inside the unique tableseq instance and instantiates $X$ with them.
Extraction language: *Elog* (body of the rule)

- **Attribute conditions**: impose restrictions on matched elements. E.g. the value is *italics*.

- **Element characterizations**: the value is a concept like “isCity”.

- **Tree Extraction Definition Predicates**: a variable should be instantiated with a node in the HTML tree which matches an element path definition.
Extraction language: *Elog* (body of the rule)

String extraction definition predicates: every node $n$ of the parse tree by concatenating all strings corresponding to leaves of the subtree rooted in $n$.

Contextual conditions: some other elements must or must not appear either before or after some instances.

Internal conditions: some characteristic feature must or must not appear with an instance.
Extraction language: Elog

*body of the rule*

- **Concept conditions**: predicates like `isEmail(X)`, `isCurrency(X)`
- **Comparison conditions**: compare two dates,
- **Pattern References**: parent pattern defines the context of a rule.
- **Range conditions**: any rule a range condition such as “[3,7]” can be added.
Lecture of Web-based IE Technologies

Ebay Charity Fundraising
Find what interests you most. Browse by Theme pages

Top > Computers > Laptops, Notebooks > General

Current || New Today || Ending Today || Completed || Going, Going, Gone

Related Topics: PC Systems | Workstations and Servers | Video Equipment | Half.com Computers

2323 General items in All eBay

Compaq Armada M700 PIII 450mhz 128mb DVD
$499.00

Panasonic CF-35 P 150 LAPTOP 96MB CD 12.1"TF
$1.00

Acer Travelmate 366 MHz 160MB 4GB 12.1"CD\(\times\)M\(\times\)CDROM
$157.00

CTX 300 mhz/32 meg/\(\times\)20/\(\times\)64k/w98/TFT active!
$449.00

AT&T Gloabalyst 200 486 Color Laptop
$10.00

LAPLINK PURPLE USB CABLE NR
$10.00

Sell Laptops on Ebay, make $5000 a week w/CD
$0.01

DELL 7000 P2 366 LAPTOP 15"TFT 160MB 64CD DVD
$355.00

Sell Laptops on Ebay, make $5000 a week w/CD
$0.01

For more items in this category, click these pages:

\[ 1 = 2 3 4 5 6 \cdots 20 \cdots 40 \cdots 67 \quad \text{next page} \]
tablesq(S,X) ← document("www.ebay.com/",S), subsq(S,(.body, []),(.table, []),(.table, []),X),
before(S,X,(.table, [(elementtext, item, substr)]),0,0,-,after(S,X, .hr, 0,0,-,)
record(S,X) ← tableseq(_,S), subelem(S,.table,X)
itemnum(S,X) ← record(_,S), subelem(S, *.td,X), notbefore(S,X, .td, 100)
itemdes(S,X) ← record(_,S), subelem(S, *(td .* .content, [(a,. substr)],X)
price(S,X) ← record(_,S), subelem(S, *(td, [(elementtext, \var[Y], *,regvar)]),X), isCurrency(Y)
bids(S,X) ← record(_,S), subelem(S, *td,X), before(S,X, .td, 0, 30,Y, _), price(_,Y)
currency(S,X) ← price(_,S), subtext(S, \var[Y], X), isCurrency(Y)
pricewc(S,X) ← price(_,S), subtext(S, [0-9]+, [0-9]+,X)
How to build the extraction rules?

**Pattern**: A set of rules defining the same head.

**Rule**: A rule defines many extraction conditions, such as attribute condition, element characterization,...

**Filter**: like a rule.
How to Build Wrapper

New Pattern Generation

Input: Parent pattern S
Output: Child pattern T defined as a set of rules d

Select a suitable instance s of the parent pattern S containing an instance of the desired target pattern T. (I)

Highlight an instance t of T within s. (I) Select characteristic attributes of t. (I/A) System creates main rule goal of the desired filter. (A)

Test whether d extracts exactly the desired set of instances of T. (I/A)

User is shown currently matched instances of T within all instances of S and asked if satisfied (= no unwanted target matched). (A)

Extend filter by adding a constraint condition (I/A). See Condition Builder Figure for details.

Let d be the set of filters so far constructed for pattern T. Add current filter to d. System adds a corresponding rule to program. (A)

Add d to the program and remember all instances of T for future pattern generation steps. (A)

*I: interactive
*A: automatic

Interactively generate a new pattern

Lecture of Web-based IE Technologies
Recursive Wrapping

“$1$” is interpreted as a constant whose value is the URL of the start document.

document(S,X) ← getDocument($1$, X)

table(S,X) ← document(_,S), subelem(S,.\* .table,X)

table(S,X) ← table(_,S), subelem(S,.\* .table,X)

It extracts all nested tables within one page, starting with the outermost, and stores them in this hierarchical order in the pattern instance base. The second rule of \texttt{<table>} is iteratively called, until no further table can be extracted.
Recursive extract pages which are connected to each other via a “next page” link.
Results reported from Lixto

<table>
<thead>
<tr>
<th>Name</th>
<th>Website</th>
<th>Used Example Page</th>
<th>Testpages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td><a href="http://www.amazon.com/">http://www.amazon.com/</a></td>
<td>Lord of the Rings</td>
<td>10</td>
</tr>
<tr>
<td>CIA Factbook</td>
<td><a href="http://www.odci.gov/cia/publications/factbook/">www.odci.gov/cia/publications/factbook/</a></td>
<td>United Kingdom</td>
<td>12</td>
</tr>
<tr>
<td>Cinemachine</td>
<td><a href="http://www.cinemachne.com/">www.cinemachne.com/</a></td>
<td>The World is not enough</td>
<td>15</td>
</tr>
<tr>
<td>DBLP</td>
<td><a href="http://www.informatik.uni-trier.de/~ley/db/">www.informatik.uni-trier.de/~ley/db/</a></td>
<td>Michael Ley</td>
<td>10</td>
</tr>
<tr>
<td>Election Results / State</td>
<td><a href="http://www.cnn.com/ELECTION/2000/results/">www.cnn.com/ELECTION/2000/results/</a></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>eBay</td>
<td><a href="http://www.ebay.com/">www.ebay.com/</a></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Excite Weather</td>
<td><a href="http://www.excite.com/weather/forecast">www.excite.com/weather/forecast</a></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Jobs-Jobs-Jobs</td>
<td><a href="http://www.jobsjobsjobs.com/">www.jobsjobsjobs.com/</a></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Perl Module List</td>
<td><a href="http://www.cpan.org/modules/00modlist.long.html">www.cpan.org/modules/00modlist.long.html</a></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Travelnotes</td>
<td><a href="http://www.travelnotes.org/">www.travelnotes.org/</a></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Yahoo People Email</td>
<td>people.yahoo.com/</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Yahoo Weather</td>
<td>weather.yahoo.com/</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Some of the test-sites used for Lixto

<table>
<thead>
<tr>
<th>Name</th>
<th>wrapable?</th>
<th>Complexity</th>
<th>Correct</th>
<th>for 100%</th>
<th>Time/Pattern (mins)</th>
<th>Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>yes</td>
<td>16/9 = 1.78</td>
<td>95%</td>
<td>3</td>
<td>22/9 = 2.44</td>
<td>4</td>
</tr>
<tr>
<td>CIA Factbook</td>
<td>yes</td>
<td>17/5 = 3.4</td>
<td>80%</td>
<td>3</td>
<td>18/5 = 3.6</td>
<td>3</td>
</tr>
<tr>
<td>Cinemachne</td>
<td>yes</td>
<td>6/4 = 1.5</td>
<td>100%</td>
<td>1</td>
<td>16/4 = 4</td>
<td>2</td>
</tr>
<tr>
<td>DBLP</td>
<td>yes</td>
<td>27/9 = 3</td>
<td>90%</td>
<td>2</td>
<td>54/9 = 6</td>
<td>8</td>
</tr>
<tr>
<td>Election Results / State</td>
<td>yes</td>
<td>4/2 = 2</td>
<td>100%</td>
<td>1</td>
<td>6/2 = 3</td>
<td>3</td>
</tr>
<tr>
<td>eBay</td>
<td>yes</td>
<td>19/8 = 2.38</td>
<td>99.9%</td>
<td>2</td>
<td>21/8 = 2</td>
<td></td>
</tr>
<tr>
<td>Excite Weather</td>
<td>yes</td>
<td>22/7 = 3.14</td>
<td>100%</td>
<td>1</td>
<td>30/7 =</td>
<td></td>
</tr>
<tr>
<td>Jobs-Jobs-Jobs</td>
<td>yes</td>
<td>21/12 = 1.75</td>
<td>90%</td>
<td>3</td>
<td>40/12 =</td>
<td></td>
</tr>
<tr>
<td>Perl Module List</td>
<td>yes</td>
<td>22/5 = 4.4</td>
<td>(100 %)</td>
<td>(1)</td>
<td>60/5 =</td>
<td></td>
</tr>
<tr>
<td>Travelnotes</td>
<td>yes</td>
<td>11/4 = 2.75</td>
<td>95%</td>
<td>2</td>
<td>20/4 =</td>
<td></td>
</tr>
<tr>
<td>Yahoo People Email</td>
<td>yes</td>
<td>10/3 = 3.3</td>
<td>100%</td>
<td>1</td>
<td>24/3 =</td>
<td></td>
</tr>
<tr>
<td>Yahoo Weather</td>
<td>yes</td>
<td>22/10 = 2.2</td>
<td>100%</td>
<td>1</td>
<td>12/10 =</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Evaluation of wrapper generation

Lecture of Web-based IE Technologies

how many example pages are necessary to get 100 percent of correctly matched pattern instances

the time needed for constructing the initial wrapper based on one example page
Question

How does the company profit from the data extraction program?

http://www.lixto.com
RoadRunner System

Aim:
- Extract data-intensive web sites.
- Data is stored in a back-end DBMS, HTML pages are dynamically generated using scripts

Methods:
- Unsupervised wrapper generation
- Do not assume that sample pages are manually selected → the system is able to automatically cluster pages in a site into homogeneous classes
- Does not rely on user-specified labeled examples → wrappers are generated and data are extracted in a completely automatic way.
- Do not assume any a priori knowledge about the target schema → deal with flat records and also nested structures.
Overview of RoadRunner System

Given a set of HTML pages, find a **schema** for the content of these pages.

A set of extraction rules parse the HTML code and retrieve the data according to the discovered schema.

Pattern discovery can be based on the study of similarities and dissimilarities between the pages.
A running example

- **Fig. a:** a nested dataset by querying a database.
- **Fig. b:** each author’s book information with the same style.

**Method:** compares the HTML codes of the two pages, infers a common structure and a wrapper, and use that to extract the source dataset.
Result of the extraction in the example

c. Data Extraction Output

| A                      | B                        | C          | D    | E                         | F                                                                 
|------------------------|---------------------------|------------|------|----------------------------|----------------------------------------------------------------------
| Computer Systems       | First Edition, Paperback  | 1995       | $40  | An undergraduate level introduction to computer... |
|                        |                           |            |      |                                                                           |
| Paul Jones             | XML at Work               | First Edition, Paperback | 1999  | $30                        | A comprehensive description of XML and all related standards...        |
|                        |                           |            |      |                                                                           |
|                        | HTML and Scripts          | First Edition, Paperback | 1983  | $30                        | A useful HTML handbook with a good tutorial on the use of sc...        |
|                        | JavaScripts               | jawf       | 2000 | $50                        | A must in every Webmaster’s bookshelf...                              |
The Architecture of the System

- **Classifier:** analyzes pages from the target site and collect them into clusters with a homogeneous structure.

- **Aligner:** compares the HTML sources of some samples pages to infer a grammar to be used as a wrapper.
How to identify different pages classes in the target sites?

(Classifier: mapping a sample to the feature space)

- **Tag Probability**: it is reasonable to assume that pages complying the same grammar have a similar “distribution” of tags, i.e., tags appear in the pages with similar probability.

- **Tag Periodicity**: there are cases in which tag probabilities may be misleading, since they do not give information about the relative positions of tags. Tag frequency is used to complement tag probability.

- **Distance from the Home Page**: if navigation paths in the site are well organized, it is reasonable to assume that pages containing homogeneous information are approximately at the same distance from the home page in the site graph.

- **URL Similarity**
Expander: infer a wrapper for those singleton pages. Most singleton pages are indices or links to other pages.

Labeler: associates a semantic meaning to the data fields that can be extracted by running the wrappers generated by the above modules.

Discussion:
How to label the data item extracted from the page?
The Labeler (methods)

- To be done manually.
- Adoption of knowledge representation techniques, by some domain ontology.
- Based on a generalized notion of closeness between wrapper’s tokens and non-terminal symbols.
Check whether the pattern sub-tree is adjacent with some \textit{isomorphic sub-tree}. The leaves of the discovered tree can be selected as names for the non-terminals of the patterns tree. \textit{namely}, the strings "name" and "phone" – are candidate to be used as names for the non-terminals $A$ and $B$ respectively.
Richness of the Web itself:

- It is possible that in some page a given data item is associated with some information describing its meaning.
- It is reasonable that in some of the pages retrieved by the search engine, the input value is explicitly associated with some descriptive text.
Further Research:

Simultaneous Record Detection and Attribute Labeling in Web Data Extraction

Jun Zhu† Zaiqing Nie‡ Ji-Rong Wen‡ Bo Zhang† Wei-Ying Ma‡

†Department of Computer Science & Technology
Tsinghua University
Beijing, China
‡Web Search & Mining Group
Microsoft Research Asia
Beijing, China

†{jun-zhu, dcszb}@mails.tsinghua.edu.cn
‡{znie, jrwen, wyma}@microsoft.com

ABSTRACT
Recent work has shown the feasibility and promise of template-independent Web data extraction. However, existing approaches use decoupled strategies — attempting to do data record detection and attribute labeling in two separate phases. In this paper, we show that separately extracting data records and attributes is highly ineffective and propose a probabilistic model to perform these two tasks simultaneously. In our approach, record detection can benefit from the availability of semantics required in attribute labeling and, at the same time, the accuracy of attribute labeling can be improved when data records are labeled in a collective manner. The proposed model is called Hierarchical Conditional Random Fields. It can efficiently integrate all useful features by learning their importance, and it can also incorporate hierarchical interactions which are very important for Web data extraction. We empirically compare the proposed model with existing decoupled approaches for product information extraction, and the results show significant improvements in both record detection and attribute labeling.

1.1 Motivating Example
We begin by illustrating the problem with an example, drawn from an actual application of product information extraction. The goal of the application is to extract meta-data about real-world products from every product page on the Web. Specifically, for each crawled Web page, we first use a classifier to decide whether it is a product page and then extract the name, image, price and description of each product from detected product pages.

Our statistical study on 51K randomly crawled Web pages shows that about 12.6 percent are product pages. That is, there are about 1 billion product pages within a search index containing 9 billion crawled Web pages. If all of these pages or just half of them are correctly extracted, we will have a huge collection of meta-data about real-world products that could be used for further business applications and it is an important research area.
Summarization

- lixTo system (interactive wrapper generation, semi-supervised)
- Roadrunner system (data-intensive page extraction, unsupervised)
References


Never-Ending Learning (NELL)

Read the web 24 hours/day since Jan. 2010.

Acquired a knowledge base with 80 million confidence-weighted beliefs.

http://rtw.ml.cmu.edu
A set \( L = \{ L_i \} \) of learning tasks.  
where \( L_i = (T_i, P_i, E_i) \) performance metric \( P_i \), on a given performance task \( T_i \), through a given type of experience \( E_i \);  

A set of coupling constraints \( C = \{ \varphi_K, V_{ki} \} \)  

\( \varphi_K \) is a real-valued function over two or more learning tasks, specifying the degree of satisfaction of the constraint.  

\( V_{ki} \) a vector of indices over learning tasks.
Problem Statement (cont.)

\[ L = (L, C) \]
\[ L = \{ \langle T_i, P_i, E_i \rangle \} \]
\[ C = \{ \langle \phi_k, V_k \rangle \} \]

Above, each performance task \( T_i \) is a pair \( T_i \equiv \langle X_i, Y_i \rangle \) defining the domain and range of a function to be learned \( f_i^* : X_i \to Y_i \). The performance metric \( P_i : f \to \mathbb{R} \) defines the optimal learned function \( f_i^* \) for the \( i \)th learning task:

\[ f_i^* \equiv \arg \max_{f \in F_i} P_i(f) \]

where \( F_i \) is the set of all possible functions from \( X_i \) to \( Y_i \).
Input of the System

Input
- Ontology and binary relations (~800 categories and relations)
- 10-20 Labeled training examples for each category and relation
- The web and access to 100,000 Google API search queries.
- Occasional interaction with humans

System Doing
- read (extract) more beliefs from the web
- remove old incorrect beliefs
- populate a growing knowledge base containing a confidence and provenance for each belief
- learn to read better than the previous day.

Result: KB with +90,000,000 extracted beliefs (different levels of confidence)
Output of the system

Figure 1: Fragment of the 80 million beliefs NELL has read from the web. Each edge represents a belief triple (e.g., play(MapleLeafs, hockey), with an associated confidence and provenance not shown here. This figure contains only correct beliefs from NELL’s KB – it has many incorrect beliefs as well since NELL is still learning.
NELL architecture

Lecture of Web-based IE Technologies
Techniques used for learning tasks

- **Category classification**: NELL learns different boolean functions for each of the 280 categories in its ontology, allowing noun phrases to refer to entities in multiple semantic categories.

- **Relation classification**: NELL learns distinct boolean-valued classification functions for each of the 327 relations in its ontology.

- **Entity Resolution**: Functions that classify noun phrase pairs by whether they are synonyms.

- **Inference rules among belief triples**: Functions that map from NELL’s current KB, to new beliefs it should add to its KB.
Techniques used for coupling constraints

- Multi-view co-training coupling.
- Subset/superset coupling.
- Multi-label mutual exclusion coupling.
- Coupling relations to their argument types.
- Horn clause coupling.
Coupled semi-supervised training of many functions

hard
(underconstrained)
semi-supervised learning problem

much easier (more constrained)
semi-supervised learning problem

Lecture of Web-based IE Technologies
Type 1 Coupling Constraints in NELL
Type 2 Coupling Constraints in NELL

Learn functions with the same input, different outputs, where some constraint are known.

- athlete(NP) → person(NP)
- athlete(NP) → NOT sport(NP)
- NOT athlete(NP) ← sport(NP)
NP: NP text, context, distribution

NP morphology

NP HTML contexts

Over 2500 coupled functions in NELL
Type 3 Coupling: Argument Types

Constraint: f3(x1,x2) \rightarrow (f1(x1) \text{ AND } f2(x2))

playsSport(a,s)
playsForTeam(a,t)
teamPlaysSport(t,s)
coachesTeam(c,t)

NP1
NP2

playsSport(NP1,NP2) \rightarrow athlete(NP1), sport(NP2)
Advantages of NELL

- To achieve successful semi-supervised learning, couple the training of many different learning tasks.
- Allow the agent to learn additional coupling constraints.
- Learn new representations that cover relevant phenomena beyond the initial representation.
- Organize the set of learning tasks into an easy-to increasingly-difficult curriculum.