# The Wrapper Induction Environment

**Nicholas Kushmerick**
Dublin City University
nick@compapp.dcu.ie

**Brett Grace**
Excite, Inc.
bgrace@excite.com

## Abstract

There is much interest in systems that automatically interact with Internet information sites. Such systems are hard to build, partly because they use hand-crafted *wrappers* to extract a site's content. We advocate *wrapper induction*, a technique for automatically learning wrappers. Our wrapper induction environment (WIEN) enables users to quickly capture a set of example page; our wrapper learning algorithm then handles the low-level details of constructing the wrapper.

**Introduction.** The Internet presents numerous sources of information: telephone directories, airline schedules, retail product catalogs, *etc.* There has been tremendous interest in information-integration systems that automatically manipulate such sites' content on a user's behalf (*e.g.* (Etzioni & Weld 1994; Kirk *et al.* 1995)).

Unfortunately, these sites are often formatted for people rather than machines, and no provision is made for automating the process. Specifically, the content is often embedded in an HTML page, and an information-integration system must extract the relevant text, while discarding irrelevant material such as HTML tags or advertisements.

Information-integration systems typically use hand-coded *wrappers* to perform this information extraction process. But as the Internet grows, maintaining a large wrapper repository becomes very challenging.

To simplify the wrapper-construction process, we advocate *wrapper induction* (Kushmerick 1997; Kushmerick, Weld, & Doorenbos 1997), a technique for automatically generating wrappers. Our wrapper induction environment (WIEN) helps wrapper developers rapidly gather and label the examples needed by our wrapper induction algorithm.

**Wrapper induction.** As an example, suppose an information-integration system must extract the content shown in Fig. 1(a) from the page in (b), which was rendered from the HTML in (c). The 'ccwrap' wrapper in (d) can perform this extraction task. 'ccwrap' operates by scanning an HTML page for particular strings ('<B>', '</B>', *etc.*) that identify the parts of the page to be extracted. In a nutshell, our learning algorithms constructs wrappers like 'ccwrap', from sets of page/label pairs such as (a)/(c).

'ccwrap' is very simple, and most Internet site are more complicated that our fictitious example. But our empirical results indicate that our techniques are appropriate for numerous actual Internet sites: we find that 70% of surveyed sites can be handled by our techniques, and our algorithm usually requires just a handful of examples and a few CPU seconds of processing.

**WIEN.** Fig. 1(e–i) illustrates the use of WIEN to build a wrapper for the Lycos search engine, which involves the following steps:

*Domain specification (e):* The user can specify the attributes to be extracted. WIEN uses distinct colors to highlight the fragments of each page to be extracted for each attribute.

*Gathering & labelling examples (f):* Using a standard Internet browser, the user gives WIEN a set of example pages from the site, as well as the text to be extracted from each. Using the mouse, the user drag-selects the fragments of the example page to be extracted.

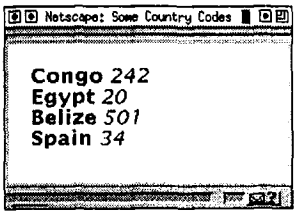
*Building the wrapper:* Once the examples have been gathered, the user simply invokes a 'Build Wrapper' command. Once learned, the wrapper can be tested on additional examples; if it makes mistakes, then the user need only correct them and re-invoke the learner.

*Source mode (g):* In some circumstances (*e.g.*, when extracting URLs), the text fragments to be extracted are not rendered by the browser; WIEN handles such cases with its 'HTML source mode'.

*Recognizers (h):* To simplify the task of labeling the examples, WIEN provides an extensible facility to automate this process. When started, WIEN loads a dynamically maintained library of *recognizers*. A recognizer is a procedure for examining a page and identifying "interesting" text fragments. We have built recognizers that identify URLs, email addresses, dates, times, US ZIP codes, ISBN num-

(a) $\left\{ \begin{array}{c} \langle \text{'Congo', '242'} \rangle, \\ \langle \text{'Egypt', '20'} \rangle, \\ \langle \text{'Belize', '501'} \rangle, \\ \langle \text{'Spain', '34'} \rangle \end{array} \right\}$

(b)



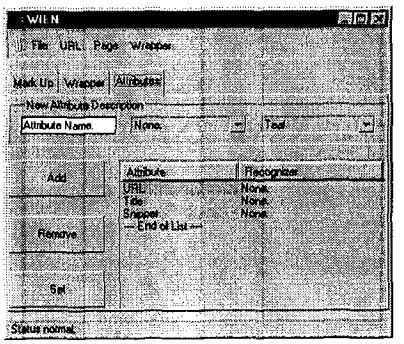

(c)

```
<HTML><TITLE>Some Country Codes</TITLE><BODY>
<B>Congo</B> <I>242</I><BR>
<B>Egypt</B> <I>20</I><BR>
<B>Belize</B> <I>501</I><BR>
<B>Spain</B> <I>34</I><BR>
</BODY></HTML>
```

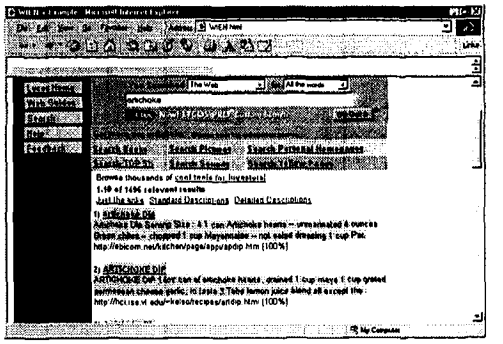
(d)

procedure $\text{ccwrap}_{\text{LR}}$(page $P$)
  while there are more occurrences in $P$ of '<B>'
    for each $\langle \ell_k, r_k \rangle \in \{\langle \text{'<B>', '</B>'} \rangle, \langle \text{'<I>', '</I>'} \rangle\}$
      scan in $P$ to next occurrence of $\ell_k$
      scan in $P$ to next occurrence of $r_k$
  return extracted $\{\ldots, \langle \text{country, code} \rangle, \ldots\}$ pairs

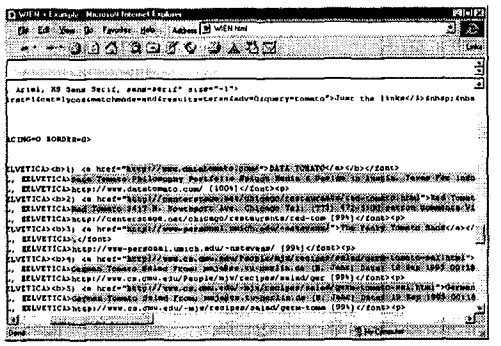
(e)




(f)



(g)



(h)

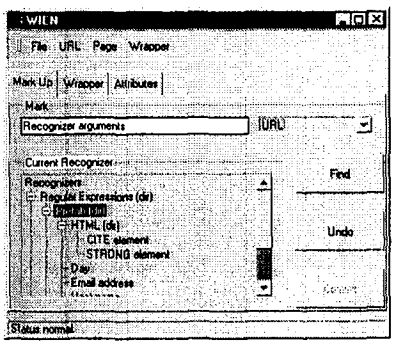



Figure 1: An example (a–d), and the WIEN application (e–h).

bers, and so forth; more sophisticated recognizers might try to locate company or personal names.

**Discussion.** WIEN was developed in the specific context of our work on wrapper induction, but we believe that our tool is applicable more generally, and we hope that WIEN is useful to the wrapper induction community at large (*e.g.*, (Ashish & Knoblock 1997; Hsu 1998)). To this end, we have designed WIEN so that its learning module is cleanly decoupled from the labeling and recognizer facilities. WIEN is available at www.compapp.dcu.ie/~nick/research/wrappers/wien.

**Acknowledgements.** This research was conducted in collaboration with Dan Weld at the Univ. of Washington, and was funded by ONR Grant N00014-94-1-0060, NSF Grant IRI-9303461, ARPA/Rome Labs grant F30602-95-1-0024, and Rockwell International Palo Alto Research.

## References

Ashish, N., and Knoblock, C. 1997. Semi-automatic wrapper generation for Internet information sources. In *Proc. Cooperative Information Systems.*

Etzioni, O., and Weld, D. 1994. A softbot-based interface to the Internet. *C. ACM* 37(7):72–6.

Hsu, C. 1998. Initial Results on Wrapping Semistructured Web Pages with Finite-state Transducers and Contextual Rules. In *Workshop on AI and Information Integration, AAAI-98.*

Kirk, T.; Levy, A.; Sagiv, Y.; and Srivastava, D. 1995. The Information Manifold. In *AAAI Spring Symposium: Information Gathering from Heterogeneous, Distributed Environments*, 85–91.

Kushmerick, N.; Weld, D.; and Doorenbos, R. 1997. Wrapper Induction for Information Extraction. In *Proc. 15th Int. Joint Conf. AI.*

Kushmerick, N. 1997. *Wrapper Induction for Information Extraction.* Ph.D. Dissertation, Univ. of Washington.