Searching for Hidden-Web Databases

Luciano Barbosa       Juliana Freire

University of Utah
The Hidden Web

- Web content *hidden* behind form interfaces
  - Not accessible through traditional search engines
- Databases on the Web
  - Lots of data (7,500--91,000 TB)
  - High-quality content: flight schedules, library catalogs, sensor readings, patent filings, genetic research data, ...
- Lots of *academic and commercial* interest in accessing and integrating hidden-Web data
  - MetaQuerier (UIUC), Metasearch (UIC), Tatu (U. Utah), ...
  - Yahoo!, Transformic, Dipsie, CompletePlanet...
- Issues considered: Crawling, form clustering, form matching
- Problem: How to find hidden-Web databases?
Searching for Hidden-Web Databases

- Find and identify the entry points to the hidden Web
  - Given the set of all Web pages $W$, find $W_f \subset W$ st $W_f$ contains searchable forms

- Very sparse space: the Web is huge, there are relatively few databases -- $|W_f| << |W|$
  - Google indexes 8 billion pages
  - ~300,000 hidden databases [Chang et al, 2004]

  Look for a few needles in a haystack

- Forms not precisely defined

- Requirements
  - Perform a broad search
  - Avoid visiting pages unnecessarily
Traditional Crawlers

- Start from a seed set of urls
- Recursively follow links to other documents
- Problems:
  - Too many documents must be retrieved before hitting the target
  - Inefficient: it takes too long to crawl the whole Web
Focused Crawlers

- Search and retrieve only subset of Web that pertains to a specific topic of relevance
  - consider the contents
- Retrieves a small subset of the documents on the Web
- Problem: still inefficient
  - Too many pages retrieved -- few forms even within a restricted domain
Form-Focused Crawler

- Focus on topic – just like a focused crawler
- Also focus on finding forms: better guide the search
- Goal: ensure a high harvest rate – fraction of pages fetched that contain forms
Form-Focused Crawler: Overview

Crawler \(\xrightarrow{\text{Page}}\) Page Classifier \(\xrightarrow{\text{Forms}}\) Form Classifier

Page Classifier \(\xrightarrow{\text{Links}}\) Link Classifier

Link Classifier \(\xrightarrow{(\text{Link, Relevance})}\) Frontier

Frontier \(\xrightarrow{\text{Seeds}}\) Form Database

Searchable Forms

Most Relevant Link
Page Classifier

- Goal: identify pages that belong to a given topic – keep the crawler on a specific topic

- Similar to the best-first focused crawler [Chakrabarti et al, 1999]

- Implementation: used Rainbow to build as Naïve Bayes classifier
Link Classifier

Crawler → Page Classifier → Forms Classifier → Form Database

Page Classifier → Frontier

Frontier → Most Relevant Link

Link Classifier

Links → (Link, Relevance)

Searchable Forms
Link Classifier

- Goal: further focus the crawl – prioritize links that are likely to lead (or that are close) to pages $P_f$ containing forms

- Problem: Learn patterns of links that lead to $P_f$ in 1 or more steps
  - Estimate how far is a link $l$ from $P_f$

- Building the classifier:
  - Repeatedly crawl sample sites to learn features of good paths [Rennie and McCallum, 1999]
  - May require crawling a substantial portion of the sites to build a good sample of the Web graph:
    - Ok for well-defined search problems
    - Too inefficient for a broad search
Learning by Crawling Backwards

- Build a sample of the paths by crawling backwards
  - Find a representative set of pages $P_0$ that contain searchable forms
  - Use Google or AltaVista “link:” facility
    $P_1 = \text{link}(P_0); \ P_2 = \text{link}(P_1); \ ...$

- Extract features from links in
  level $n$ ($P_n, P_{n-1}$), level $n-1$ ($P_{n-1}, P_{n-2}$), ..., level $2$ ($P_2, P_1$), level $1$ ($P_1, P_0$)
  - Contexts: anchor, URL, text in the proximity of the URL

- Build a Naïve Bayes classifier
  - Obtain the probabilistic class membership for links in a level
Feature Space for Jobs

- Words related to topic domain and “search” in all features
- Frequency of *relevant* words decreases as the distance from target page increases, but many relevant words are present in lower levels
- Important to combine focused crawler with link classifier
Form Classifier

Crawler → Page Classifier → Forms Classifier → Form Database

Page Classifier → Form Classifier

Crawler → Page Classifier → Links → Link Classifier → Frontiers

Frontier → Most Relevant Link

Searchable Forms

Form Database
Form Classifier

- Not all forms are entry points to hidden-Web databases
  - E.g., login, discussion groups, mailing list subscriptions
- Goal: Identify searchable forms

Quick Job Search

Enter Keyword(s): [Help]
(i.e. job title, company name)

Enter a City: [City List]

Select a State:
- All United States

Select a Category:
- All Job Categories
- Or select multiple categories

Advanced Search: Search>>

Did you like my survey?

- Yes
- Sort Of
- No

Submit Vote

Searchable form

Non-searchable form
Form Classifier

- Positive examples: UIUC repository
- Negative examples: manually collected
- 14 features: number of checkboxes, textboxes etc.
- Tried different classifiers
- Chosen: C4.5 – lowest test error rate

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error test rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>C4.5</td>
<td>8.02%</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>14.19%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>10.49%</td>
</tr>
<tr>
<td>Multiayer Perceptron</td>
<td>9.87%</td>
</tr>
</tbody>
</table>
Managing the Frontier

- Frontier
  - Links to be visited ordered by *importance*
  - Determines the crawler’s steps

- Link classifier sets the priority
  - One queue per level

- Root pages have the highest priority within a queue
  - Searchable forms are close to the root page [Chang et al, 2004]
Stopping Criteria

- Some sites may not have searchable forms
- Sites have very few searchable forms
  - Average of 4.2 query interfaces per deep-Web site [Chang et al, 2004]
- Avoid unnecessarily crawling a given site for too long
- Stop crawling a site $S$ if
  - Enough *distinct* forms are retrieved, or
  - A maximum number of pages is visited in $S$
Experiments

- Crawlers:
  - Baseline: Variation of the best-first crawler [Chakrabarti et al, 1999]
  - Fixed depth [Chang et al, 2005]
  - Baseline SC: Baseline + stop criteria
  - Form Crawler with 1-3 levels
  - Form Crawler with 1-3 levels without prioritizing root pages

- Measure the number of distinct relevant forms in relation to number pages visited

- Domains: cars, jobs, books
Results: Cars

- Fixed-depth retrieved only 254 forms after crawling 30,000 pages
- Baseline crawler similar to fixed-depth
- Baseline SC is much better than baseline: 1,168 forms after crawling 30,000 pages
- Multiple levels are better than baseline SC: 2,883 forms retrieved
Results: Jobs

Number of forms vs Number of pages

- form crawler (3 levels)
- form crawler (2 levels)
- form crawler (1 level)
- fixed depth
- baseline SC

WebDB 2005 – Searching for Hidden-Web Databases
L. Barbosa, J. Freire
Results: Books

![Graph showing the number of forms vs number of pages for different crawling methods: form crawler (3 levels), form crawler (2 levels), form crawler (1 level), fixed depth, and baseline SC. The graph illustrates how the number of forms increases with the number of pages for each method.]

WebDB 2005 – Searching for Hidden-Web Databases
L. Barbosa, J. Freire
Prioritizing Root Pages: Jobs

Number of forms vs. Number of pages for:
- fc 3-level
- fc 3-level no root
- priority
- baseline SC

Graph shows the comparison of these methods with respect to the number of forms detected as the number of pages increases.
Experiments: Summary

- Learning link/path features + considering delayed benefit is effective
  - Best performance: multi-level Form Crawler
  - 3 vs 1 level leads to 20-110% improvement
- Too many levels does not help
  - $\geq 4$, no improvement
- Stopping criteria are critical
  - Baseline SC is much better than baseline
- Fixing the crawling depth is not enough
- High priority to root pages helps
Related Work

- Focused crawlers:
  - Avoid off-topic pages (Chakrabarti et al, 1999 and 2002)
    - Only consider links that give *immediate benefit*
    - May miss relevant pages linked from irrelevant pages
  - Consider delayed benefit (Rennie and McCallum, 1999 and Diligenti et al, 2000)

- MetaQuerier Database Crawler (Chen et al, 2005)
  - No focus on topic, fixes depth of breadth-first crawl
Conclusion and Future Work

- New efficient crawling strategy to automatically discover hidden-Web databases
  - Focus on topic, prioritize promising links, use appropriate stopping criteria

- Issue: are the retrieved forms good?
  - Form classifier is effective, but not perfect
  - Can’t check +3 thousand forms manually!
  - Further analyze forms in repository

- Hidden-Web database directory
  - Build form crawlers for different topics
  - Organize them in a hierarchy (a la DMOZ)