



ProFoUnd: Program-analysis–based Form Understanding

(joint work with M. Benedikt,
T. Furche, A. Savvides)

PIERRE SENELLART





The Deep Web

Definition (Deep Web, Hidden Web, Invisible Web)

All the content on the Web that is not directly accessible through **hyperlinks**. In particular: HTML forms, Web services.



Size estimate: 500 times more content than on the **surface Web!**
[BrightPlanet, 2001]. Hundreds of thousands of deep Web databases
[Chang et al., 2004]



Example

- *Yellow Pages* and other directories;
- Library catalogs;
- Weather services;
- US Census Bureau data;
- etc.

Discovering Knowledge from the Deep Web

[Varde et al., 2009]

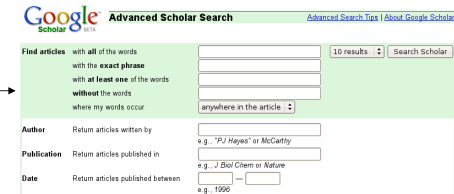
- Content of the deep Web hidden to classical Web search engines (they just follow links)
- But very valuable and high quality!
- Even services allowing access through the surface Web (e.g., e-commerce) have more semantics when accessed from the deep Web
- How to **benefit** from this information?

Focus here: Automatic, unsupervised, methods

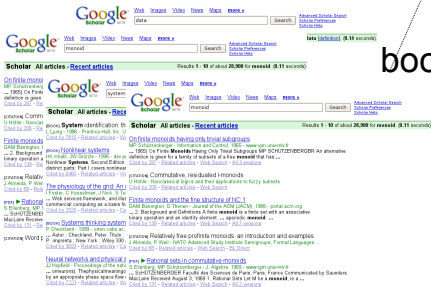
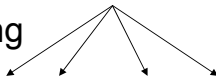
Extensional Approach



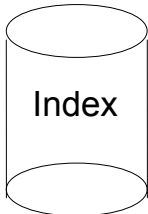
discovery



siphoning



bootstrap



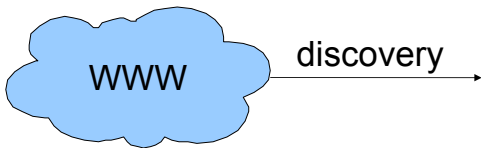
indexing



Notes on the Extensional Approach

- Main issues:
 - Discovering services
 - Choosing appropriate data to submit forms
 - Use of data found in result pages to bootstrap the siphoning process
 - Ensure good coverage of the database
- Approach **favored by Google**, used in production [Madhavan et al., 2006]
- Not always feasible (huge load on Web servers)

Intensional Approach



Google Scholar **Advanced Scholar Search** [Advanced Search Tips](#) | [About Google Scholar](#)

Find articles with all of the words 10 results | Search Scholar

with the **exact phrase**

with **at least one** of the words

without the words

where my words occur anywhere in the article

Author Return articles written by
e.g., "P.J. Hayes" or McCarthy

Publication Return articles published in
e.g., J Biol Chem or Nature

Date Return articles published between -
e.g., 1996

probing

Google Scholar data Search [Advanced Scholar Search](#) [Database Information](#) [Help](#)

Scholar All articles **Recent Articles** Results 1 - 19 of about 91,669,000 for data [Database](#) (0.18 seconds)

1. Fisher D. The use of multiple measurements in economic problems. [View Article](#)

2. Psychol. Adv. Ser. No. 1. [View Article](#)

3. Calhoun A, Fienberg D, Corcos E, Carter T, Higgins D. Between-group analysis of increasing data. [View Article](#)

4. The protein kinase encoded by the *AI1* proto-oncogene is a target of the PDGF-activated... [View Article](#)

5. RAG-2-deficient mice lack mature lymphocytes owing to inability to initiate V(D)J rearrangement... [View Article](#)

6. PD Pathway: SP Pathway: TGF-beta1... [View Article](#)

7. Both genetic and biochemical data point toward a physiological role for the complex... [View Article](#)

8. Random data analysis and measurement procedures... [View Article](#)

9. J Statist. 2001; 29(1): 1-10. [View Article](#)

10. BICM REVIEW: Random Data Analysis and Measurement Procedures... [View Article](#)

11. Data mining: practical machine learning tools and techniques with Java implementations... [View Article](#)

12. Data Mining: Practical Machine Learning Tools and... [View Article](#)

Form wrapped as
a Web service

analyzing

query





Notes on the Intensional Approach

- More **ambitious** [Chang et al., 2005, Senellart et al., 2008]
- Main issues:
 - Discovering services
 - Understanding the structure and semantics of a form
 - Understanding the structure and semantics of result pages
 - Semantic analysis of the service as a whole
- No significant load imposed on Web servers



Introduction

ProFoUnd

JavaScript and the Deep Web

Form Understanding through JavaScript Analysis

Conclusions



Introduction

ProFoUnd

JavaScript and the Deep Web

Form Understanding through JavaScript Analysis

Conclusions



Better Form Analysis

What

Where

eg. Restaurants
Hairdressers
Telstra
Apple Stores




Better Form Analysis


What

Find

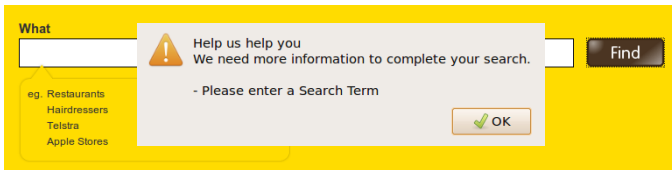
eg. Restaurants
Hairdressers
Telstra
Apple Stores

 Help us help you
We need more information to complete your search.

- Please enter a Search Term



Better Form Analysis



The screenshot shows a search interface on a yellow background. At the top left, the word "What" is displayed above a search input field. Below the input field, a list of suggestions is shown: "eg. Restaurants", "Hairdressers", "Telstra", and "Apple Stores". To the right of the input field is a "Find" button. A grey alert dialog box is overlaid on the form, containing a warning icon, the text "Help us help you We need more information to complete your search.", and a sub-message "- Please enter a Search Term". An "OK" button with a green checkmark is located at the bottom right of the alert box.

```
// Do not submit unless form is valid
$j("#searchForm").submit(function(event) {
    $j("#searchFormLocationClue").val($j("#searchFormLocationClue").val().trim());
    if ($j("#searchFormBusinessClue").val().isEmpty()) {
        alert('Help us help you\nWe need more information to
            complete your search.\n\n- Please enter a Search Term');
        return false;
    } else {
        return true;
    }
});
```



JavaScript: the Data Language of the Web

- Lots of JavaScript code on the Web (source is always available!)
- Lots of information can be gained by **static analysis** of this code:
 - **Required** fields
 - **Dependencies** between fields (if x is filled in, so should be y ; the value of x should be less than that of y ; etc.)
 - **Datatype** of each fields (regular expressions, numeric types, dates, etc.)
- Is this feasible in practice?



Outline

Introduction

ProFoUnd

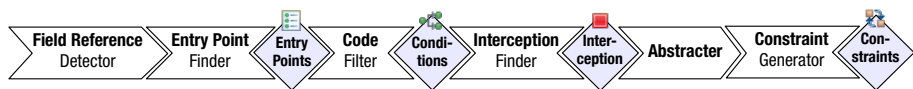
JavaScript and the Deep Web

Form Understanding through JavaScript Analysis

Conclusions



ProFoUnd architecture



- **Entry points** are HTML event attributes, setting of event handlers in code, etc. (event: *click* on a submit button, *submit* on a form)
- **Conditions** are (in)equality tests on form field values (possibly aliased)
- **Interceptions** are interruptions of the form submission process (error messages, simple return `false`; in event handler, etc.)



Abstracting the code

- Rice's theorem: no hope in a **sound** and **complete** constraint finder
- But that's ok! **Anything** that we can learn is more than what we have at the moment.
- **Coarse abstraction** of the JS code:
 - Only conditions on the code flow from entry points to interceptions are considered.
 - We consider only a simple subset of the JS language; anything beyond that is ignored.
 - Side-effects are mostly ignored
- As a consequence: no guarantee of either soundness or completeness \Rightarrow **only experimental guarantees**



Engineering issues to deal with

- Extracting a Web form model: **DIADEM**'s tools <http://www.diadem-project.info/>
- Parsing JavaScript: **Mozilla Rhino** (but see later)
- JavaScript frameworks: **ad-hoc support** for most popular ones (jQuery, Prototype, ASP.NET generated code, YUI, Dojo, MooTools)
- Evaluating JavaScript code (e.g., to determine what a jQuery selector (`$("#form#lookup .product")`) returns): **Mozilla JS engine**
- Abstraction, alias references, etc.: ProFoUnd core, **developed from scratch**

ProFoUnd interface [Benedikt et al., 2012]

The screenshot displays the ProFoUnd interface with three numbered callouts:

- 1**: A browser window showing a web page with a "Keyword:" field, "Price (USD)" section, and "Submit Query" button. The "Minimum:" field contains "0" (highlighted in green) and the "Maximum:" field contains "10000" (highlighted in blue).
- 2**: A "Constraints" panel on the left showing a list of constraints. The selected constraint is "CONSTRAINT: LE" with conditions "on <INPUT id=min>" and "on <INPUT id=max>".
- 3**: A "Javascript / Event handler" panel at the bottom showing a code snippet:

```
if (min > max)
{
  $(".label_price").css("color", "red");
  return false;
}
```

1. Web page view, with fields highlighted
2. Constraints found:
 $min < max, max \neq 0,$
 $product \neq "$
3. JS fragment for the highlighted constraint



Preliminary evaluation

- 70 real-estate websites containing search forms
- 30 out of 70 use client-side validation, with a total of 35 constraints
- **100% precision**: all identified constraints are correct
- **63% recall**: 22 out of 35 JS-enforced constraints were found
- Why did we miss some?
 - Use of complex JavaScript features, such as `eval`
 - Code obfuscation by introducing extra layers of computation
 - Limitations of the abstracter – work in progress!



Outline

Introduction

ProFoUnd

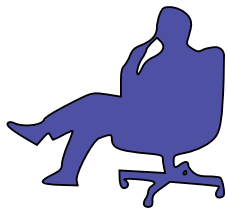
Conclusions



- Exploiting data from the deep Web in an automatic manner:
non-trivial, largely open problem
- Classical techniques exploit both **domain knowledge** and the structure of **forms and result pages**
- Possible to get very precise information about the behavior of Web forms by **static analysis of client-side code**



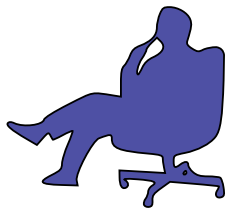
Perspectives



- Use a **real JS parser** (Rhino has lots of limitations); trying with SpiderMonkey, Mozilla's JS engine
- **Large-scale evaluation**, application to deep Web crawling
- **Type inference** for form fields: regular expressions, simple datatypes
- Combining with **dynamic analysis**
- **Type inference** for AJAX applications: static analysis of AJAX calls to determine input and output types (possibly JSON or XML types)

PhD Opportunity

PhD scholarship on this topic at U. Oxford, looking for excellent candidates!



- Use a **real JS parser** (Rhino has lots of limitations); trying with SpiderMonkey, Mozilla's JS engine
- **Large-scale evaluation**, application to deep Web crawling
- **Type inference** for form fields: regular expressions, simple datatypes
- Combining with **dynamic analysis**
- **Type inference** for AJAX applications: static analysis of AJAX calls to determine input and output types (possibly JSON or XML types)

PhD Opportunity

PhD scholarship on this topic at U. Oxford, looking for excellent candidates!



Merci.

Michael Benedikt, Tim Furche, Andreas Savvides, and Pierre Senellart. ProFoUnd: Program-analysis-based form understanding. In *Proc. WWW*, Lyon, France, April 2012. Demonstration.

BrightPlanet. The deep Web: Surfacing hidden value. White Paper, July 2001.

Kevin Chen-Chuan Chang, Bin He, Chengkai Li, Mitesh Patel, and Zhen Zhang. Structured databases on the Web: Observations and implications. *SIGMOD Record*, 33(3):61–70, September 2004.

Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Toward large scale integration: Building a metaquerier over databases on the Web. In *Proc. CIDR*, Asilomar, USA, January 2005.

Jayant Madhavan, Alon Y. Halevy, Shirley Cohen, Xin Dong, Shawn R. Jeffery, David Ko, and Cong Yu. Structured data meets the Web: A few observations. *IEEE Data Engineering Bulletin*, 29 (4):19–26, December 2006.

Pierre Senellart, Avin Mittal, Daniel Muschick, Rémi Gilleron, and Marc Tommasi. Automatic wrapper induction from hidden-Web sources with domain knowledge. In *Proc. WIDM*, pages 9–16, Napa, USA, October 2008.

Aparna Varde, Fabian M. Suchanek, Richi Nayak, and Pierre Senellart. Knowledge discovery over the deep Web, semantic Web and XML. In *Proc. DASFAA*, pages 784–788, Brisbane, Australia, April 2009. Tutorial.