Automatic Wrapper Induction from Hidden-Web Sources with Domain Knowledge

P. Senellart\textsuperscript{1,2} A. Mittal\textsuperscript{3} D. Muschick\textsuperscript{6} R. Gilleron\textsuperscript{4,5} M. Tommasi\textsuperscript{4,5}

University of Oxford, 5 May 2009
The Hidden Web

Definition (Hidden Web, Deep 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!
Hundreds of thousands of hidden Web databases.
Sources of the Hidden Web

Example

- Yellow Pages and other directories;
- Library catalogs;
- Weather services;
- US Census Bureau data;
- etc.
Discovering Knowledge from the Deep Web

- 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?
- How to do it automatically, in an unsupervised way?

Focus here: understanding form-based query interfaces, and corresponding result pages.
Extensional Approach

WWW discovery

siphoning

bootstrap

Index

indexing

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction with Domain Knowledge
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

Not always feasible (huge load on Web servers)
Intensional Approach

WWW discovery

Form wrapped as a Web service

analyzing

query

probing

Motivation

Probing

Two-Step Wrapper Induction

Experiments

Conclusion

Senellart, Mittal, Muschick, Gilleron, Tommassi

Wrapper Induction with Domain Knowledge
More ambitious

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
General architecture
Motivation

Probing

Two-Step Wrapper Induction

Experiments

Conclusion

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction with Domain Knowledge
Analyzing the structure of HTML forms.

Goal

Associating to each form field the appropriate domain concept.
First Step: Structural Analysis

1. Build a **context** for each field:
   - label tag;
   - id and name attributes;
   - text immediately before the field.

2. Remove **stop words**, **stem**.

3. **Match** this context with the concept names, extended with **WordNet**.

4. Obtain in this way **candidate annotations**.
First Step: Structural Analysis

1. Build a context for each field:
   - label tag;
   - id and name attributes;
   - text immediately before the field.

2. Remove stop words, stem.

3. Match this context with the concept names, extended with WordNet.

4. Obtain in this way candidate annotations.
First Step: Structural Analysis

1. Build a **context** for each field:
   - label tag;
   - id and name attributes;
   - text immediately before the field.

2. Remove **stop words**, **stem**.

3. **Match** this context with the concept names, extended with WordNet.

4. Obtain in this way **candidate annotations**.
First Step: Structural Analysis

1. Build a context for each field:
   - label tag;
   - id and name attributes;
   - text immediately before the field.

2. Remove stop words, stem.

3. Match this context with the concept names, extended with WordNet.

4. Obtain in this way candidate annotations.
Second Step: Confirm Annotations with Probing

For each field annotated with a concept $c$:

1. **Prove the field with nonsense word to get an error page.**
2. **Probe the field with instances of $c$ (chosen representatively of the frequency distribution of $c$).**
3. **Compare pages obtained by probing with the error page (by using clustering along the DOM tree structure of the pages), to distinguish error pages and result pages.**
4. **Confirm the annotation if enough result pages are obtained.**
Second Step: Confirm Annotations with Probing

For each field annotated with a concept \( c \):

1. Probe the field with nonsense word to get an error page.
2. Probes the field with instances of \( c \) (chosen representatively of the frequency distribution of \( c \)).
3. Compare pages obtained by probing with the error page (by using clustering along the DOM tree structure of the pages), to distinguish error pages and result pages.
4. Confirm the annotation if enough result pages are obtained.
Second Step: Confirm Annotations with Probing

For each field annotated with a concept $c$:

1. Probe the field with nonsense word to get an error page.
2. **Probe** the field with instances of $c$ (chosen representatively of the frequency distribution of $c$).
3. Compare pages obtained by probing with the error page (by using clustering along the DOM tree structure of the pages), to distinguish error pages and result pages.
4. Confirm the annotation if enough result pages are obtained.
Second Step: Confirm Annotations with Probing

For each field annotated with a concept $c$:

1. Probe the field with nonsense word to get an error page.
2. **Probe** the field with instances of $c$ (chosen representatively of the frequency distribution of $c$).
3. Compare pages obtained by probing with the error page (by using clustering along the DOM tree structure of the pages), to distinguish error pages and result pages.
4. **Confirm** the annotation if enough result pages are obtained.
1 Motivation

2 Probing

3 Two-Step Wrapper Induction

4 Experiments

5 Conclusion
Result Pages

Pages resulting from a given form submission:

- share the **same structure**;
- set of **records** with fields;
- **unknown** presentation!

**Goal**

Building wrappers for a given kind of result pages, in a fully automatic, **unsupervised**, way.

**Simplification**: restriction to a domain of interest, with some domain knowledge.
Different Approaches to Information Extraction
(Chang et al., TKDE 2006)
Annotation by domain knowledge

Automatic pre-annotation with domain knowledge (gazetteer):

- Entity recognizers for dates, person names, etc.
- Titles of articles, conference names, etc.: those that are in the knowledge base.

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction with Domain Knowledge
Annotation by domain knowledge

Automatic **pre-annotation** with domain knowledge (gazetteer):

- Entity recognizers for dates, person names, etc.
- Titles of articles, conference names, etc.: those that are in the knowledge base.

Showing results 1 through 25 (of 94 total) for all:xml

1. cs.LO/0601085 [abs, ps, pdf, other]
   Title: A Formal Foundation for ODRL
   Authors: Riccardo Pucella, Vicky Weissman
   ACM-class: H.2.7; K.4.4

2. astro-ph/0512493 [abs, pdf]
   Title: VOFILTER, Bridging Virtual Observatory and Industrial Office Applications
   Authors: Chengzhou Cui (1), Markus Dolemsky (2), Peter Quinn (2), Yong-heng Zhao (1), Francoise Genova (3) ((1)NAO China, (2) ESO, (3) CDS)
   Comments: Accepted for publication in China J. (9 pages, 2 figures, 18K).

3. cs.DS/0512061 [abs, ps, pdf, other]
   Title: Matching Subsequences in Trees
   Authors: Philip Bille, Inge Li Goertz
   Subj-class: Data Structures and Algorithms

4. cs.IR/0510025 [abs, ps, pdf, other]
   Title: Practical Semantic Analysis of Web Sites and Documents
   Authors: Thierry Despeyroux (INRIA Rocquencourt / INRIA Sophia Antipolis)
   Subj-class: Information Retrieval

5. cs.CR/0510013 [abs, pdf]
   Title: Safe Data Sharing and Data Dissemination on Smart Devices
   Authors: Luc Bouzeghroum (INRIA Rocquencourt), Cosmin Cremaere (INRIA Rocquencourt), Francois Dang Ngoc (INRIA Rocquencourt, PRISM - UVSQ), Nicolas Dieu (INRIA Rocquencourt), Philippe Pucheral (INRIA Rocquencourt, PRISM - UVSQ)
   Subj-class: Cryptography and Security; Databases
Automatic pre-annotation with domain knowledge (gazetteer):

- **Entity recognizers for dates, person names, etc.**
- **Titles of articles, conference names, etc.: those that are in the knowledge base.**
Automatic **pre-annotation** with domain knowledge (gazetteer):

- Entity recognizers for dates, person names, etc.
- Titles of articles, conference names, etc.: those that are in the knowledge base.

Both **incomplete and imprecise!**
Unsupervised Wrapper Induction

- Use the pre-annotation as the input of a structural supervised machine learning process.
- Purpose: remove outliers, generalize incomplete annotations.
Conditional Random Fields

- Generalization of hidden Markov Models
- Probabilistic **discriminative** model: models the probability of an annotation **given an observable** (different from **generative** models)
- **Graphical model**: every annotation can depend on the neighboring annotations (as well as the observable); dependencies measured through (boolean or integer) **feature functions**.
- Features are automatically assigned a weight and combined to find the **most probable annotation** given the observable.
Conditional Random Fields for XML (XCRF)

**Observables**: various structural and content-based features of nodes (tag names, tag names of ancestors, type of textual content...).

**Annotations**: domain concepts assigned to nodes of the tree.

**Tree probabilistic model**:
- models dependencies between annotations;
- conditional independence: annotations of nodes only depend on their neighbors (and on observables).

Most **discriminative** features selected.
Architecture

- **CORPUS**
  - Pretreatment
    - Tokenization
    - Preprocessing
  - Extract Features
  - Gazetteer
  - XCRF
  - Train XCRF

- **POSTTREATMENT**
  - LCA Analysis
  - Outlier Removal

- **ANNOTATED**

- **EXTRACT DATA**

**Senellart, Mittal, Muschick, Gilleron, Tommasi**

**Wrapper Induction with Domain Knowledge**
Motivation

Probing

Two-Step Wrapper Induction

Experiments

Conclusion

1 Motivation

2 Probing

3 Two-Step Wrapper Induction

4 Experiments

5 Conclusion

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction with Domain Knowledge
Experimental Setup

- 10 services of research publication databases.
- Domain knowledge extracted from DBLP.
- Forms analyzed and probed (5 probes per field and candidate annotation).
- Induction of wrappers from training (unannotated) set of result pages, and evaluation of wrappers on test set of result pages.
Results for form analysis

<table>
<thead>
<tr>
<th></th>
<th>Initial annot.</th>
<th>Confirmed annot.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( p(%) )</td>
<td>( r(%) )</td>
</tr>
<tr>
<td>Average</td>
<td>49</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td></td>
<td>82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>73</td>
</tr>
</tbody>
</table>

- Good precision and recall.
- Probing raises precision without hurting recall.

Remark

Much better results for distinguishing error and result pages by clustering according to the paths in the DOM tree than previous approaches.

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction with Domain Knowledge
## Results for wrapper induction

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_g$</td>
<td>$F_x$</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>63</td>
<td>85</td>
</tr>
<tr>
<td>$F_g$</td>
<td>$F_x$</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>70</td>
<td>76</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $F_g$: $F$-measure (%) of the annotation by the gazetteer.
- $F_x$: $F$-measure (%) of the annotation by the induced wrapper.
Motivation

Probing

Two-Step Wrapper Induction

Experiments

Conclusion

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction with Domain Knowledge
Summary

Important point

It is indeed possible to use content and structure together for automatic, unsupervised, information extraction!

- better than content only (gazetteer);
- better than structure only (RoadRunner).

- Content is used to bootstrap a structure-based learning: isn’t it what humans do to understand the structure of such pages?
- Not perfect (yet), should be possible to get much better!
Important point

It is indeed possible to use content and structure together for automatic, unsupervised, information extraction!

- better than content only (gazetteer);
- better than structure only (RoadRunner).

- Content is used to bootstrap a structure-based learning: isn’t it what humans do to understand the structure of such pages?
- Not perfect (yet), should be possible to get much better!
Perspectives

- More **intelligent** gazetteer: use NL features to extract noun phrases that look like titles?
- A machine learning framework adapted to a **noisy** and **incomplete** annotation, without **overfitting**: minimal-length description?
- Exploit **probabilities** that come with learned features to produce **ranked** information extractor.
Merci.