Motivation

Probing

Two-Step Wrapper Induction

Experiments

Conclusion

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}

\textit{WIDM}, 28 October 2008
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 (2001) : 500 times more content than on the surface Web!
Sources of the Hidden Web

Example

- *Yellow Pages* and other directories;
- Library catalogs;
- Weather services;
- US Census Bureau data;
- etc.
Analyzing the **structure** of HTML forms.

<table>
<thead>
<tr>
<th>Authors</th>
<th></th>
<th></th>
<th></th>
<th>Year</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conference</td>
<td></td>
<td></td>
<td></td>
<td>ID</td>
<td></td>
</tr>
<tr>
<td>Journal</td>
<td></td>
<td></td>
<td></td>
<td>Volume</td>
<td>Number</td>
</tr>
</tbody>
</table>

**Goal**

Associating to each form field the appropriate **domain concept**.
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**.

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction from Domain Knowledge
General architecture
1 Motivation

2 Probing

3 Two-Step Wrapper Induction

4 Experiments

5 Conclusion

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction from Domain Knowledge
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. **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.
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.
Motivation

Probing

Two-Step Wrapper Induction

Experiments

Conclusion

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction from 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.
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.
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.
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.

Both incomplete and imprecise!
Unsupervised Wrapper Induction

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

```
<table>
<thead>
<tr>
<th>table / articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>tr / article</td>
</tr>
<tr>
<td>td / title</td>
</tr>
<tr>
<td>token / title</td>
</tr>
<tr>
<td>#text</td>
</tr>
<tr>
<td>td / authors</td>
</tr>
<tr>
<td>token / author</td>
</tr>
<tr>
<td>#text</td>
</tr>
</tbody>
</table>
```

Senellart, Mittal, Muschick, Gilleron, Tommasi

Wrapper Induction from Domain Knowledge
Conditional Random Fields for XML (XCRF)

**Observations:** 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 observations).

Most **discriminative** features selected.
Architecture
<table>
<thead>
<tr>
<th></th>
<th>Motivation</th>
<th>Probing</th>
<th>Two-Step Wrapper Induction</th>
<th>Experiments</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Motivation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Probing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Two-Step Wrapper Induction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Experiments</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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>82</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.
# 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>$F_g$</td>
</tr>
<tr>
<td>Average</td>
<td>44</td>
<td>63</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.

Senellart, Mittal, Muschick, Gilleron, Tommasi  
Wrapper Induction from Domain Knowledge
1 Motivation

2 Probing

3 Two-Step Wrapper Induction

4 Experiments

5 Conclusion

Senellart, Mittal, Muschick, Gilleron, Tommasi
Wrapper Induction from Domain Knowledge
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!
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!
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.