LaSEWeb: Automating Search Strategies over Semi-Structured Web Data

Oleksandr Polozov  
University of Washington  
polozov@cs.washington.edu

Sumit Gulwani  
Microsoft Research  
sumitg@microsoft.com
Motivation: search engine micro-segments
Motivation: search engine micro-segments
Motivation: search engine micro-segments
Motivation: search engine micro-segments
Repetitive search tasks

Structured databases

- Precise, but limited in content
- No time-sensitive information
- Provide no context (sources)
Repetitive search tasks

Structured databases
- Precise, but limited in content
- No time-sensitive information
- Provide no context (sources)

Web mining scripts
- Two extremes:
  - Powerful ML, which has to be re-learned for each micro-segment
  - Fragile HTML layout parser
- Inaccessible for end-users
LaSEWeb Query Language

- A semantic scripting language for semi-structural information extraction from the Web
- Models natural patterns from the humans’ search strategies

LaSEWeb interpreter

- Explores multiple webpages, clusters different answer candidates, and provides context for each answer
- Makes use of state-of-the-art NLP/ML/PL algorithms
Example: phone number

$$\tilde{\nu} = \text{"Sumit Gulwani"}$$

let \(\eta_t = \text{Emphasized}(\nu_1)\) in
let \(\eta_b = \text{AttributeLookup}(\text{Syn}(\text{"phone"}), \ell_a)\) in
\(\text{Union}(\eta_t, \eta_b)\)

where \(\text{Regex}(\ell_a, \text{"/(d+)/W * \d + W * \d+"})\)
where \(\text{Layout}(\eta_t, \eta_b, \text{Down})\) and \(\text{Nearby}(\eta_t, \eta_b)\)
Example: phone number

\[ \tilde{\nu} = ("Sumit Gulwani") \]

\[
\begin{align*}
\text{let } & \eta_t = \text{Emphasized}(\nu_1) \text{ in } \\
\text{let } & \eta_b = \text{AttributeLookup}(\text{Syn}("phone"), \ell_a) \text{ in } \\
& \text{Union}(\eta_t, \eta_b) \\
\text{where } & \text{Regex}(\ell_a, "((d+)W*d + W*d+)" ) \\
\text{where } & \text{Layout}(\eta_t, \eta_b, \text{Down}) \text{ and } \text{Nearby}(\eta_t, \eta_b)
\end{align*}
\]

- Visual attributes
Example: phone number

\[ \tilde{v} = ("\text{Sumit Gulwani}") \]

let \( \eta_t \) = \text{Emphasized}(v_1) \text{ in }

let \( \eta_b \) = \text{AttributeLookup}(\text{Syn("phone")}, \ell_a) \text{ in }

\text{Union}(\eta_t, \eta_b)

where \text{Regex}(\ell_a, "\((d+)\)W*d + W*d+")

where \text{Layout}(\eta_t, \eta_b, \text{Down}) \text{ and } \text{Nearby}(\eta_t, \eta_b)

- Visual attributes
- Implicit table detection
Example: phone number

\[ \tilde{v} = \text{"Sumit Gulwani"} \]

\[
\begin{align*}
\text{let } & \eta_t = \text{Emphasized}(v_1) \text{ in } \\
\text{let } & \eta_b = \text{AttributeLookup}(\text{Syn("phone")}, \ell_a) \text{ in } \\
\end{align*}
\]

\[
\begin{align*}
\text{Union}(\eta_t, \eta_b) \\
\text{where } & \text{Regex}(\ell_a, "\(\(d+\)\)\W*\d+\W*\d+") \\
\text{where } & \text{Layout}(\eta_t, \eta_b, \text{Down}) \text{ and } \text{Nearby}(\eta_t, \eta_b)
\end{align*}
\]

- Visual attributes
- Implicit table detection
- Linguistic patterns
Example: phone number

\[ \tilde{v} = \text{("Sumit Gulwani")} \]

\[
\text{let } \eta_t = \text{Emphasized}(v_1) \text{ in } \\
\text{let } \eta_b = \text{AttributeLookup}(\text{Syn("phone")}, [\ell_a]) \text{ in } \\
\text{Union}(\eta_t, \eta_b) \\
\text{where } \text{Regex}(\ell_a, "/(d+)/W*d + W*d+)"
\]

where  \( \text{Layout}(\eta_t, \eta_b, \text{Down}) \) and  \( \text{Nearby}(\eta_t, \eta_b) \)

- Visual attributes
- Implicit table detection
- Linguistic patterns
- Clustering across webpages
Language Structure

Visual patterns
- Match: webpage layout, style, end-user appearance
- Use: in-memory rendering, DOM analysis
- Nearby, Emphasized, Layout, CSS ...

Structural patterns
- Match: relational patterns on implicit tables
- Use: table detection, plain text analysis using programming-by-example technologies
- VLOOKUP, AttributeLookup ...

Linguistic patterns
- Match: semantic text properties
- Use: POS tagging, sentence parsing, entity recognition, synonymy detection...
- Syn, POS, Entity, NP, SameSentence ...

Program interpreter: “user emulation” algorithm
Program interpreter: “user emulation” algorithm

\[ \tilde{v} = \text{"computer"} \]
Program interpreter: “user emulation” algorithm

\[ \tilde{v} = ("\text{computer"}) \]

LaSEWeb Engine

Seed query

LaSEWeb “inventors” MS script
Program interpreter: “user emulation” algorithm

\[ \tilde{v} = ("computer") \]
Program interpreter: “user emulation” algorithm

\[ \hat{v} = \text{"computer"} \]

LaSEWeb Engine

Seed query

LaSEWeb “inventors” MS script

\[ \text{score}(C_i) = \frac{1}{|U|} \sum_{j=1}^{|U|} \sum_{s \in C_i} \frac{c(s, u_j)}{c(u_j)} \]

“John Atanasoff”

“John Vincent Atanasoff”

“Charles Babbage”

“Babbage, C.”

“konrad zuse”
Program interpreter: “user emulation” algorithm

\[ \vec{v} = \text{"computer"} \]

LaSEWeb Engine

LaSEWeb “inventors” MS script

Seed query

John Atanasoff (14.5%)
http://www.computerhope.com
http://www.ehow.com
http://inventors.about.com

Charles Babbage (10.5%)
http://www.buzzle.com
http://www.ask.com
...

\[
\text{score}(C_i) = \frac{1}{|U|} \sum_{j=1}^{|U|} \sum_{s \in C_i} \frac{c(s, u_j)}{c(u_j)}
\]

“John Atanasoff”
http://www.computerhope.com
http://www.ehow.com
http://inventors.about.com

“John Vincent Atanasoff”

“Charles Babbage”
http://www.buzzle.com
http://www.ask.com
...

“Babbage, C.”
http://www.buzzle.com
http://www.ask.com
...

“konrad zuse”
http://www.buzzle.com
http://www.ask.com
...
Experiments

- ~95% precision and 71% recall on factoid micro-segments
  - For micro-segments: Precision measured by random sampling, based on top-3 results
  - For end-user repetitive search tasks: Precision/recall measured manually
- Average execution time: ~5 sec/webpage
  - Depends on the rendering settings
- Current setting: offline deployment / database population

<table>
<thead>
<tr>
<th>Micro-segment</th>
<th># queries</th>
<th>Recall</th>
<th>Bing recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII code of a symbol</td>
<td>1,551</td>
<td>32.88%</td>
<td>0%</td>
</tr>
<tr>
<td>Calories in a food</td>
<td>9,207</td>
<td>71.80%</td>
<td>0%</td>
</tr>
<tr>
<td>Inventor of a product</td>
<td>8,994</td>
<td>75.91%</td>
<td>16.01%</td>
</tr>
<tr>
<td>Lyrics of a song</td>
<td>48,995</td>
<td>24.36%</td>
<td>0%</td>
</tr>
<tr>
<td>Phone number of a company</td>
<td>6,881</td>
<td>95.49%</td>
<td>0%</td>
</tr>
<tr>
<td>Population of a place</td>
<td>18,151</td>
<td>92.53%</td>
<td>57.58%</td>
</tr>
<tr>
<td>Release date of a product</td>
<td>12,339</td>
<td>97.24%</td>
<td>12.60%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Search task</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone #</td>
<td>29/37</td>
<td>21/29</td>
</tr>
<tr>
<td>Affiliation</td>
<td>34/37</td>
<td>22/34</td>
</tr>
<tr>
<td>PhD institution</td>
<td>21/37</td>
<td>13/21</td>
</tr>
<tr>
<td>General chair</td>
<td>21/28</td>
<td>17/21</td>
</tr>
<tr>
<td>Invited talks</td>
<td>13/28</td>
<td>11/13</td>
</tr>
<tr>
<td>Average</td>
<td>71%</td>
<td>73%</td>
</tr>
</tbody>
</table>
Summary & Future work

• Typical patterns of human search strategies in a scripting language for IE
  • Match semi-structured Web content
  • Existing cross-disciplinary technologies used as building blocks
  • Exploit information redundancy across multiple webpages

• Applications:
  1. Micro-segments of factoid questions in search engines
  2. Repeatable batch data extraction tasks for end-users
  3. Structured database population from free Web text
  4. English language comprehension problem generation

• Future work:
  • Automatic query execution plans in the language
  • Integration with “natural language → logic” engines
Summary & Future work

1. The principal characterized his pupils as __________ because they were pampered and spoiled by their indulgent parents.
2. The commentator characterized the electorate as __________ because it was unpredictable and given to constantly shifting moods.
   (a) cosseted   (b) disingenuous   (c) corrosive   (d) laconic   (e) mercurial

2. Repeatable batch data extraction tasks for end-users.
3. Structured database population from free Web text.
4. English language comprehension problem generation.

• Future work:
  • Automatic query execution plans in the language
  • Integration with “natural language → logic” engines
Summary & Future work

• Typical patterns of human search strategies in a scripting language for IE
  • Match semi-structured Web content
  • Existing cross-disciplinary technologies used as building blocks
  • Exploit information redundancy across multiple webpages

• Applications:
  1. Micro-segments of factoid questions in search engines
  2. Repeatable batch data extraction tasks for end-users
  3. Structured database population from free Web text
  4. English language comprehension problem generation

• Future work:
  • Automatic query execution plans in the language
  • Integration with “natural language → logic” engines
Thanks for listening!

Questions?