Web Page Clustering Using Heuristic Search in the Web Graph

Ron Bekkerman
Shlomo Zilberstein
James Allan
Example application

- Given a person name
- Find out *everything* about this person
  - What is available in the Web
- Possible solution:
  - Query a search engine with the person name
  - Retrieve documents
  - Cluster retrieved documents
    - Build *largest* clusters possible
Query: “Michel Décary”
Query: “Michel Décary”

Lawyer

Computer scientist

Chanson singer
Query: “Michel Décary”

Education
McGill University, Master’s Program in Law and Economics, course requirements completed (1969-1970), Université de Montréal (LL.B., 1967), and Université de Montréal (B.A., 1964).

Décary holds a B.A. in linguistics and an M.Sc. in computer science from Université de Montréal. He also pursued doctoral studies in computational linguistics at Université de Montréal and in applied linguistics at McGill University.
Query: “Michel Décary”

Results 1 - 6 of about 7 linking to www.cogilex.com. (0.16 seconds)

Founder: Cogilex R&D inc. - Montreal, QC

Computer scientist

Chanson singer

N'hésitez pas à m'écrire à decary@cogilex.com
Query: “Michel Décary”

- Lawyer
- Computer scientist
- Chanson singer
Summary of observations

- *Topical* clustering is not enough
  - Although not to be ignored

- Web graph topology should be exploited
  - Close pages tend to be semantically related
    - There’s a *reason* for hyperlinking page $A \rightarrow$ page $B$
    - Be careful: arbitrary connections exist as well
  - Apply *beam search* to find close pages
    - Use heuristics to prune undesired branches
Example: breadth first search
Clustering by multi-agent search

- Each page is represented by a Web agent
  - Whose task is to traverse the Web graph
  - And meet as many other agents as possible
Real-world case

- The Web is tightly interconnected
  - About 70% agents meet after 3 search iterations
  - Which is clearly an undesired outcome
Heuristic 1

- Elimination of high-degree nodes
  - Both high in-degree and high out-degree
  - They often connect unrelated pages
Heuristic 2

- Person name sharing
  - Expanded nodes share a hyperlink
  - AND a person name (ignore too popular names)
Heuristic 3

- Anchor text sharing
  - Anchor texts often summarize the content of hyperlinked pages
- Same idea as in person name sharing
  - But much simpler to implement
    - No sophisticated information extraction needed
    - Shallow parsing of HTML is enough
  - Again, ignore too frequent anchor texts
    - Like “Contact Us” or “Copyright”
Algorithmic enhancement

- Unpleasant artifact: too long connections
  - Too weak semantic relationships

- Proposed solution: keep track of cluster’s diameter
  - Start with a tightly connected component
  - Add pages found within one hop
Experimentation domains

- Web appearance disambiguation
  - Given pages retrieved on $N$ people names
    - From one social network
  - Filter out pages that refer to their unrelated namesakes

- Clustering of Web search results
  - Represent ranked lists of retrieved documents as clusters of semantically related documents

*Bekkerman & McCallum, WWW-05*

*Hearst & Pedersen, SIGIR-96*
# Disambiguation dataset

- 12 names out of Melinda Gervasio’s social network

<table>
<thead>
<tr>
<th>Personal name</th>
<th>Position</th>
<th>Pages</th>
<th>Namesakes</th>
<th>Relevant pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam Cheyer</td>
<td>SRI Manag</td>
<td>97</td>
<td>2</td>
<td>96</td>
</tr>
<tr>
<td>William Cohen</td>
<td>CMU Prof</td>
<td>88</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Steve Hardt</td>
<td>SRI Eng</td>
<td>81</td>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>David Israel</td>
<td>SRI Manag</td>
<td>92</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>Leslie Pack Kaelbling</td>
<td>MIT Prof</td>
<td>89</td>
<td>2</td>
<td>88</td>
</tr>
<tr>
<td>Bill Mark</td>
<td>SRI Manag</td>
<td>94</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>Andrew McCallum</td>
<td>UMass Prof</td>
<td>94</td>
<td>16</td>
<td>54</td>
</tr>
<tr>
<td>Tom Mitchell</td>
<td>CMU Prof</td>
<td>92</td>
<td>37</td>
<td>15</td>
</tr>
<tr>
<td>David Mulford</td>
<td>Stanf Undergrad</td>
<td>94</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>Andrew Ng</td>
<td>Stanf Prof</td>
<td>87</td>
<td>29</td>
<td>32</td>
</tr>
<tr>
<td>Fernando Pereira</td>
<td>UPenn Prof</td>
<td>88</td>
<td>19</td>
<td>32</td>
</tr>
<tr>
<td>Lynn Voss</td>
<td>SRI Eng</td>
<td>89</td>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td><strong>OVERALL:</strong></td>
<td></td>
<td>1085</td>
<td>187</td>
<td>420</td>
</tr>
</tbody>
</table>
Disambiguation results

- **SHS** – sequential heuristic search (basic algorithm)
- **IHS** – incremental heuristic search

- **h/d** – high degree node elimination
- **names** – person name heuristic

Each point: one iteration of search
- Only 2 iterations are enough

- **SHS** – sequential heuristic search (basic algorithm)
- **IHS** – incremental heuristic search
- With the enhancement of diameter tracking
Jaguar dataset

- 100 pages retrieved on query “Jaguar”
  - 23 different categories!
- The task is to build 3 clusters
  - Of cars, Mac OS and wild cats

<table>
<thead>
<tr>
<th>Category</th>
<th>Pages</th>
<th>Category</th>
<th>Pages</th>
<th>Category</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car</td>
<td>36</td>
<td>Reef lodge</td>
<td>2</td>
<td>Atari game</td>
<td>5</td>
</tr>
<tr>
<td>MacOS</td>
<td>11</td>
<td>Book</td>
<td>1</td>
<td>Guitar</td>
<td>1</td>
</tr>
<tr>
<td>Wild cat</td>
<td>23</td>
<td>Singer</td>
<td>2</td>
<td>TV channel</td>
<td>1</td>
</tr>
<tr>
<td>Biotech firm</td>
<td>2</td>
<td>Emulator</td>
<td>2</td>
<td>Web designer</td>
<td>2</td>
</tr>
<tr>
<td>Youth org</td>
<td>1</td>
<td>Cornell project</td>
<td>2</td>
<td>E-commerce firm</td>
<td>1</td>
</tr>
<tr>
<td>Maya culture</td>
<td>1</td>
<td>Metal band</td>
<td>1</td>
<td>Game archive</td>
<td>1</td>
</tr>
<tr>
<td>Resin models</td>
<td>1</td>
<td>Movie</td>
<td>1</td>
<td>Aircraft</td>
<td>1</td>
</tr>
<tr>
<td>Web hosting</td>
<td>1</td>
<td>Photo gallery</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OVERALL:** 100
Jaguar results

- SHS algorithm fails
  - 70 agents meet together
- anchors – anchor text heuristic
- Best performance:
  - High degree AND anchors heuristics
Topical & topological clustering

- Build topical clusters
  
- Enrich topical clusters with pages obtained by heuristic search based clustering

<table>
<thead>
<tr>
<th>Web appearance disambiguation</th>
<th>Clustering of Web search results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Method</strong></td>
<td><strong>Precision</strong></td>
</tr>
<tr>
<td>Topical</td>
<td>87.3%</td>
</tr>
<tr>
<td>IHS (iteration 1)</td>
<td>89.9%</td>
</tr>
<tr>
<td>Hybrid (iteration 1)</td>
<td>84.5%</td>
</tr>
<tr>
<td>IHS (iteration 2)</td>
<td>81.7%</td>
</tr>
<tr>
<td>Hybrid (iteration 2)</td>
<td>78.5%</td>
</tr>
</tbody>
</table>

- Best performance: after one iteration of heuristic search only!
Conclusion

- First application of heuristic search to the Web graph
  - Very simple algorithms / heurstics
  - Heuristic search theory yet to be applied
    - E.g., can an admissible heuristic be proposed?

- Search can be performed in real time!
  - Modern search engines store the link structure of most of the Web
  - Maximum 2 search iterations are enough
    - Fully distributable in a multi-agent fashion