Better Contextual Suggestions from ClueWeb12
Using Domain Knowledge Inferred from The Open Web

Thaer Samar
Alejandro Bellogin and Arjen P. de Vries
Our Submission

- **Contextual Suggestion model:**
  - Find attractions in ClueWeb12
  - Generating user profiles
  - Similarity between candidate attractions and users
  - Rank suggestion per (user, context) pair

- **RQ:**
  can we improve the performance of the contextual suggestions by applying domain knowledge?

- **Approach:**
  - Filter collection using domain knowledge to create sub-collections
  - Apply same retrieval model to different sub-collections
  - Compare differences in effectiveness
Creating Sub-collections

- GeoFiltered sub-collection
  - Applying geographical filter
    - Exact mention of the given contexts
      format: {City, ST} e.g., Miami, FL
    - Exclude documents that mention multiple contexts
      e.g., a Wikipedia page about cities in Florida state
TouristFiltered sub-collection

- Applying domain knowledge extracted from the structure of the Open Web:
  - Domain Oriented
    - Manual list of tourist websites
      {yelp, tripadvisor, wikitravel, zagat, xpedia, orbitz, and travel.yahoo}
    - From ClueWeb12
      - extract any document whose host in the list (TouristListFiltered)
        e.g., http://www.zagat.com/miami
  - Expand TouristListFiltered
    - Extract outlinks
    - Search for outlinks in ClueWeb12 (TouristOutlinksFiltered)
TouristFiltered sub-collection

- Attraction Oriented
  - Use Foursquare API to get attractions for given contexts

  ![Foursquare logo]

  Miami, FL ➔ Cortés Restaurant, http://cortesrestaurant.com

- If URL is missing for the attraction, then use Google API query: “Cortés Restaurant Miami, FL”

- For found attractions
  - Get host names of their URLs
  - From ClueWeb12 get any document whose host from the above (AttractionFiltered)
Sub-collections Summary

ClueWeb12
733,019,372 docs

GeoFiltered

“City, ST”
8,883,068 docs

TouristFiltered

TouristListFiltered (175,260)
TouristOutlinksFiltered (97,678)
Attractions Filtered (102,604)
Generating Users Profiles

- Aggregation of attractions descriptions
- Take into account ratings given by users
  - Build positive and negative profiles
Similarity

- Represent attractions and users in weighted VSM
  - Vector element \(<\text{term, frequency}>\)

- Cosine similarity

\[
sim(u^+, d) = \cos(u^+, d) = \frac{\sum_i u_i^+ \cdot d_i}{\sqrt{\|u^+\|_2} \sqrt{\|d\|_2}} \tag{1}
\]

\[
sim(u^-, d) = \cos(u^-, d) = \frac{\sum_i u_i^- \cdot d_i}{\sqrt{\|u^-\|_2} \sqrt{\|d\|_2}} \tag{2}
\]

\[
\text{score} = a \cdot \sim(u^+, d) + b \cdot \sim(u^-, d) \tag{3}
\]
Ranked suggestions

- For each (user, context) pair
  - Rank suggestions based on similarity score
  - Generate titles to represent attraction:
    - Extract from <title> or <header> tags
  - Generate descriptions tailored to the user
    - Extract content of <description> tag
    - Break documents into sentences
      - rank sentences based on their similarity with the user
    - Concatenate until 512 bytes reached
## Results (General Performance)

<table>
<thead>
<tr>
<th></th>
<th>P@5</th>
<th>MRR</th>
<th>TBG</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoFiltered</td>
<td>0.0468</td>
<td>0.0767</td>
<td>0.1256</td>
</tr>
<tr>
<td>TouristFiltered</td>
<td>0.1438</td>
<td>0.2307</td>
<td>0.6013</td>
</tr>
<tr>
<td>Median</td>
<td>0.0542</td>
<td>0.0886</td>
<td>0.1382</td>
</tr>
<tr>
<td>Best</td>
<td>0.2328</td>
<td>0.4232</td>
<td>0.9615</td>
</tr>
</tbody>
</table>
Analysis (General)

- Percentage of best and worst topics given by each run
- Exclude topics where best score=worst=0
- Compared with all runs based on ClueWeb12

<table>
<thead>
<tr>
<th></th>
<th>P@5</th>
<th>MRR</th>
<th>TBG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>best</td>
<td>worst</td>
<td>best</td>
</tr>
<tr>
<td>GeoFiltered</td>
<td>9.03</td>
<td>41.14</td>
<td>8.70</td>
</tr>
<tr>
<td>TouristFiltered</td>
<td><strong>28.43</strong></td>
<td><strong>20.07</strong></td>
<td><strong>25.42</strong></td>
</tr>
</tbody>
</table>
Analysis (TouristFiltered vs. GeoFiltered)

- Compare our runs against each other
- Percentage of topics where TouristFiltered is better than equal to and worse than GeoFiltered
- In case of equality, ignore topics when best score is zero

<table>
<thead>
<tr>
<th>Metric</th>
<th>Better</th>
<th>Equal</th>
<th>Worse</th>
</tr>
</thead>
<tbody>
<tr>
<td>TouristFiltered</td>
<td>33.11</td>
<td>15.72</td>
<td>8.36</td>
</tr>
<tr>
<td>GeoFiltered</td>
<td>32.44</td>
<td>15.72</td>
<td>9.03</td>
</tr>
<tr>
<td></td>
<td>41.47</td>
<td>15.72</td>
<td>11.04</td>
</tr>
</tbody>
</table>
Analysis (decompose metrics dimensions)

- P@5 and MRR consider three dimensions of relevance
  - Geographical (geo), description (desc) and document (doc) relevance

- Considering the desc and doc relevance
  - Two runs have similar effectiveness

<table>
<thead>
<tr>
<th>Metric</th>
<th>GeoFiltered</th>
<th>TouristFiltered</th>
</tr>
</thead>
<tbody>
<tr>
<td>P@5_all</td>
<td>0.0468</td>
<td>0.1438</td>
</tr>
<tr>
<td>P@5_desc-doc</td>
<td>0.2281</td>
<td>0.2348</td>
</tr>
<tr>
<td>P@5_desc</td>
<td>0.3064</td>
<td>0.2910</td>
</tr>
<tr>
<td>P@5_doc</td>
<td>0.2836</td>
<td>0.3124</td>
</tr>
</tbody>
</table>
Analysis (decompose metrics evaluation)

- Considering the geo aspect only
  - TouristFiltered is geographically appropriate

<table>
<thead>
<tr>
<th>Metric</th>
<th>GeoFiltered</th>
<th>TouristFiltered</th>
</tr>
</thead>
<tbody>
<tr>
<td>P@5_all</td>
<td>0.0468</td>
<td>0.1438</td>
</tr>
<tr>
<td>P@5_geo</td>
<td>0.1605</td>
<td>0.4843</td>
</tr>
</tbody>
</table>
TouristFiltered sub-collection consists of three parts
- TouristListFiltered (TLF)
- TouristOutlinksFiltered (TOF)
- AttractionFiltered (AF)

Measure how each part contributes to the performance

<table>
<thead>
<tr>
<th>Metric</th>
<th>TLF</th>
<th>TLF + TOF</th>
<th>TLF + TOF + AF</th>
<th>AF</th>
</tr>
</thead>
<tbody>
<tr>
<td>P@5_all</td>
<td>0.0314</td>
<td>0.0441</td>
<td>0.1438</td>
<td>0.1084</td>
</tr>
<tr>
<td>P@5_geo</td>
<td>0.1612</td>
<td>0.2181</td>
<td><strong>0.4843</strong></td>
<td><strong>0.4468</strong></td>
</tr>
</tbody>
</table>
Conclusions and Future work

- Applying Open Web domain knowledge leads to have better suggestions
- We can think of each part in `TouristFiltered` collection as a binary filter
- For future work:
  - We can combine different weighted filters
  - Each filter can represent a different source of knowledge