Rankbox: An Adaptive Ranking System for Mining Complex Semantic Relationships Using User Feedback

Na Chen, Viktor K. Prasanna
IEEE IRI2012, Las Vegas
Presented by Om Patri
Semantic Relationships

- Relationship mining from the semantic data
  - critical for better integration, search and decision making

- Semantic relationship search
  - Query: a pair of resources
  - Results: all relationships between them

Example results of query <Harry Potter, James Potter>:

<table>
<thead>
<tr>
<th></th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Harry Potter — has_parent — James Potter</td>
</tr>
<tr>
<td>2</td>
<td>Harry Potter — education — Hogwarts School — student_graduate — James Potter</td>
</tr>
<tr>
<td>3</td>
<td>Harry Potter — married_to — Ginny Weasley — education — Hogwarts School — student_graduate — James Potter</td>
</tr>
<tr>
<td>4</td>
<td>Harry Potter — has_posseased — Ginny Weasley — owner — Elder Wand — founder_of — Order of the Phoenix — has_member — James Potter</td>
</tr>
<tr>
<td>5</td>
<td>Harry Potter — power_or_ability — Magic — character_with_this_ability — James Potter</td>
</tr>
</tbody>
</table>
Ranking Semantic Relationships

- Ranking is important
  - Challenge: understanding user preferences

- User feedback is a valuable source of user preferences.

- We present Rankbox: an adaptive semantic relationship mining system using user feedback
  - Personalized
  - Improvable
  - End-user friendly
System Overview
Demo: Rankbox
Behind the Demo

- **Semantic relationship search**
  - Input: \(<e1, e2>\)
  - Output: paths that connect e1 and e2
  - Depth limited search

- **Ranking search results**

  \[f_u(a) = x(a) \cdot w_u\]

  - **ranking function for user** \(u\)
  - **feature vector of result** \(a\)
  - **weight vector that represents user** \(u\)'s preferences

- Results are sorted by their \(f_u\) values
Feature Vector

\[ x(a) = (l(a), t_1(a), t_2(a), \ldots, t_n(a)) \]

- Association length \( l(a) \)
- Topic features

\[ t_i(a) = \frac{|E_i|}{l(a)} \]

\[ x(a) = (0.3, 0.667, 0.333) \]
Learn the Weight Vector \( w_u \)

- **Training data**
  - Results *liked/disliked* by the user

- **Learning algorithm**
  - Linear Discriminant Analysis (LDA)
Our Interactive Learning-to-rank Algorithm

**Algorithm 1: InteractiveLtR for user u**

**Initialization:** \( w_u \leftarrow w_{default}, D_u^+ \leftarrow \emptyset, D_u^- \leftarrow \emptyset \)

**Data:** an RDF knowledge base \( M_{rdf} \)

**foreach query \( q \) from user u do**

- find \( A(q) \) by \texttt{DepthLimitedSearch}(\( M_{rdf}, q \))
- for \( a \in A(q) \) do
  - calculate feature vector \( x(a) \)
  - \( f_u(a) \leftarrow x(a) \cdot w_u \)
- sort \( A(q) \) by \( f_u(a) \)
- get u’s feedback \( F_u^+ \) (results liked by \( u \)) and \( F_u^- \) (results disliked by \( u \))
- if \( F_u^+ \neq \emptyset \) or \( F_u^- \neq \emptyset \) then
  - \( D_u^+ \leftarrow D_u^+ \cup F_u^+ \)
  - \( D_u^- \leftarrow D_u^- \cup F_u^- \)
  - apply \texttt{LDA} on \((D_u^+, D_u^-)\) to learn a new \( w_u \)
Evaluation

- **Dataset**
  - Fictional_universe domain of Freebase linked-open data
  - 36 topics, 340K entities, 590K properties

- **Platform**
  - Tomcat 7 server
  - Intel i-7 CPU 1.60GHz and 6GB memory

- **Time complexity**
  - Search: 10~15s to retrieve 500 results per query
  - Feature calculation, LDA, sorting < 5s
Qualitative Results

The user selects some search results and gives his feedback (user feedback is in blue).

The user’s preference on personal relationships has been reflected in the ranking results.
Evaluation

- 20 human participants. Each interacts with the system for 20 iterations.

- Evaluation metrics
  - Precision@10
    - the ratio of records ranked in the top 10 results that are labeled as relevant
  - # of active users@ query k
    - the number of users who give feedback in the k-th iteration
Evaluation

- Quantitative results

- Precision@10
  - Rankbox can continuously improve the ranking precision through iterations.

- # of active users@ query k
  - As the ranking quality increases, more and more users stop giving feedback
Thank You!

- **Contact:**
  - Na Chen  nchen@usc.edu
  - Viktor K. Prasanna  prasanna@usc.edu