Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling

Wei Shen†, Jianyong Wang†, Ping Luo‡, Min Wang♯
†Tsinghua University, Beijing, China
‡HP Labs China, Beijing, China
♯Google Research, Mountain View, USA

KDD 2013
Chicago, Illinois USA
Outline

- Motivation
- Problem Definition
- KAURI Framework
- Experiments
- Conclusion
Outline

- Motivation
- Problem Definition
- KAURI Framework
- Experiments
- Conclusion
Motivation

- Twitter: important information source
- Tweets: social status updates about topics ranging from daily life to news events

Tweets

- $t_1$: Bulls should still aim for a title, even through the horrible news.
- $t_2$: McNealy finished, he was pretty much squarely in Sun's camp. @jniccolai
- $t_3$: Scott explains what open means...
- $t_4$: Tyson Chandler says Tony Allen is the best on-ball defender in the #NBA http://t.co/YGmByJMC

Sun: the star at the center of the Solar System
Sun Microsystems: a multinational computer company
Sun-Hwa Kwong: a fictional character named “Sun-Hwa Kwong”
Motivation

- Many large scale knowledge bases have emerged
  - Dbpedia, YAGO, Freebase, Probase, and etc.

- Bridging these knowledge bases with the collection of tweets

Figure 1: An example of YAGO knowledge base

Source: From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. PODS’10.
Outline

- Motivation
- **Problem Definition**
- KAURI Framework
- Experiments
- Conclusion
Problem definition

- Tweet entity linking
  - link the textual named entity mentions detected from tweets with their mapping entities existing in a knowledge base.

---

Figure 2: An illustration of the tweet entity linking task. Named entity mentions detected in tweets are in bold; candidate mapping entities for each entity mention are generated by a dictionary-based method and ranked by their prior probabilities in decreasing order; true mapping entities are underlined.
Applications

- Twitter user interest discovery
- Twitter users Recommendation
- Tweets recommendation and re-ranking
- Entity information collection from Twitter
  - e.g., products and celebrities
Tweet entity linking

- Challenge
  - noisy, short, and informal nature of tweets
- Previous entity linking methods (EACL’06, EMNLP’07, KDD’09, SIGIR’11, EMNLP’11, and WWW’12)
  - focus on linking entities in Web documents
  - Context Similarity
  - Topical Coherence

Not work well
Tweet entity linking

- **Challenge**
  - noisy, short, and informal nature of tweets

- **Previous entity linking methods** (EACL’06, EMNLP’07, KDD’09, SIGIR’11, EMNLP’11, and WWW’12)
  - focus on linking entities in Web documents
  - Context Similarity
  - Topical Coherence

- **We can increase the linking accuracy, if we**
  - combine *intra-tweet local information*
  - with *inter-tweet user interest information*
Tweet entity linking

- **Challenge**
  - noisy, short, and informal nature of tweets
- **Previous entity linking methods** (EACL’06, EMNLP’07, KDD’09, SIGIR’11, EMNLP’11, and WWW’12)
  - focus on linking entities in Web documents
  - Context Similarity
  - Topical Coherence

Not work well
Outline

- Motivation
- Problem Definition
- KAURI Framework
- Experiments
- Conclusion
KAURI Framework

Assumption 1.
- Each Twitter user has an underlying topic interest distribution over various topics of named entities.

Assumption 2.
- If some named entity is mentioned by a user in his tweet, that user is likely to be interested in this named entity.

Assumption 3.
- If one named entity is highly topically related to the entities a user is interested in, that user is likely to be interested in this named entity as well.
Graph construction

- For each Twitter user,
- we construct a graph

Running example: Example 1

Example 1
Graph construction

Candidate entity

Weight:
- indicating the strength of interdependence
- calculated using the Wikipedia Link-based Measure [1].

Graph construction

- Assumption 1.
  - Each Twitter user has an underlying topic interest distribution over various topics of named entities.

\[ r_{j,q}^i \quad \text{candidate mapping entity} \]

\[ s_{j,q}^i \quad \text{interest score indicating the strength of the user's interest in it} \]
Graph construction

- **Assumption 2.**
  - If some named entity is mentioned by a user in his tweet, that user is likely to be interested in this named entity.

$r_{j,q}^i$ : candidate mapping entity

$s_{j,q}^i$ : interest score indicating the strength of the user's interest in it

$p_{j,q}^i$ : initial interest score estimated from intra-tweet local information
Initial interest score estimation

\[ p_{j,q}^i = \alpha \ast Pp(r_{j,q}^i) + \beta \ast Sim(r_{j,q}^i) + \gamma \ast Coh(r_{j,q}^i) \]

The initial interest score

Prior probability

Context similarity

Topical coherence in tweet

entity frequency in the Wikipedia article corpus
The initial interest score

\[ p_{j,q}^{i} = \alpha \times Pp(r_{j,q}^{i}) + \beta \times Sim(r_{j,q}^{i}) + \gamma \times Coh(r_{j,q}^{i}) \]

Prior probability

Context similarity

bag of words cosine similarity

entity frequency in the Wikipedia article corpus

Topical coherence in tweet

estimated using the iterative substitution algorithm proposed in [2]

\[ \overrightarrow{w} = \langle \alpha, \beta, \gamma \rangle \]

Weight vector

\[ \alpha + \beta + \gamma = 1 \]

• We utilize the max-margin technique to automatically learn the weight vector which gives proper weights for those three intra-tweet local features.

Algorithm 1 Iterative Substitution Algorithm

Input: Web list \( L \), candidate mapping entity sets \( R \).
Output: mapping entity list \( M \).

1: for each \( l_i \in L \) do
   2: \[ m_i^{(0)} = \arg \max_{r_{i,j}} P_{pr}(r_{i,j}), \ r_{i,j} \in R_i \]
   3: end for

4: \( M^{(0)} = \{m_i^{(0)} | l_i \in L\} \)
5: \( \text{iter} = 1 \)

6: while true do
   7: for each \( l_i \in L \) do
      8: for each \( r_{i,j} \neq m_i^{(\text{iter}-1)} \in R_i \) do
         9: \[ M_{r_{i,j}}^{(\text{iter})} = (M^{(\text{iter}-1)} - \{m_i^{(\text{iter}-1)}\}) \cup \{r_{i,j}\} \]
         10: \( \text{IncreLQ}_{r_{i,j}} = LQ(M_{r_{i,j}}^{(\text{iter})}) - LQ(M^{(\text{iter}-1)}) \)
      11: end for
   12: end for
   13: \[ r_{i,j}^{\text{max}} = \arg \max_{r_{i,j}} \text{IncreLQ}_{r_{i,j}}, \ r_{i,j} \in R_i, R_i \in R \]
   14: if \( \text{IncreLQ}_{r_{i,j}^{\text{max}}} > 0 \) then
      15: \[ M^{(\text{iter})} = (M^{(\text{iter}-1)} - \{m_i^{(\text{iter}-1)}\}) \cup \{r_{i,j}^{\text{max}}\} \]
      16: \( \text{iter}++ \)
      17: else
         18: break
   19: end if
20: end while
21: \( M = M^{(\text{iter}-1)} \)
Table 2: The initial interest scores for candidate mapping entities in Example 1
User interest propagation algorithm

\[ \overrightarrow{s} = \lambda \overrightarrow{p} + (1 - \lambda) \overrightarrow{B} \overrightarrow{s} \]

- Initialization: \( \overrightarrow{s} = \overrightarrow{p} \)
- Then apply this formula iteratively until \( \overrightarrow{s} \) stabilizes within some threshold

The final interest score vector
The initial interest score vector
The interest propagation strength matrix

Normalized \( p_{j,q} \)
column-normalized
User interest propagation algorithm

Table 2: The initial interest scores for candidate mapping entities in Example 1

<table>
<thead>
<tr>
<th>$r_{i,q}^t$</th>
<th>Bulls (rugby)</th>
<th>Chicago Bulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{i,q}^t$</td>
<td>0.13</td>
<td>0.0492</td>
</tr>
<tr>
<td>$r_{i,q}^t$</td>
<td>Bulls, New Zealand</td>
<td>Tyson Chandler</td>
</tr>
<tr>
<td>$p_{i,q}^t$</td>
<td>0.0208</td>
<td>0.318</td>
</tr>
<tr>
<td>$r_{i,q}^t$</td>
<td>Tony Allen (musician)</td>
<td>Tony Allen (basketball)</td>
</tr>
<tr>
<td>$p_{i,q}^t$</td>
<td>0.145</td>
<td>0.155</td>
</tr>
<tr>
<td>$r_{i,q}^t$</td>
<td>National Basketball Association</td>
<td></td>
</tr>
<tr>
<td>$p_{i,q}^t$</td>
<td>0.402</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: The final interest scores for candidate mapping entities in Example 1

<table>
<thead>
<tr>
<th>$r_{i,q}^t$</th>
<th>Bulls (rugby)</th>
<th>Chicago Bulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_{i,q}^t$</td>
<td>0.0624</td>
<td>0.189</td>
</tr>
<tr>
<td>$r_{i,q}^t$</td>
<td>Bulls, New Zealand</td>
<td>Tyson Chandler</td>
</tr>
<tr>
<td>$s_{i,q}^t$</td>
<td>0.00682</td>
<td>0.194</td>
</tr>
<tr>
<td>$r_{i,q}^t$</td>
<td>Tony Allen (musician)</td>
<td>Tony Allen (basketball)</td>
</tr>
<tr>
<td>$s_{i,q}^t$</td>
<td>0.0476</td>
<td>0.122</td>
</tr>
<tr>
<td>$r_{i,q}^t$</td>
<td>National Basketball Association</td>
<td></td>
</tr>
<tr>
<td>$s_{i,q}^t$</td>
<td>0.297</td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Motivation
- Problem Definition
- KAURI Framework
- Experiments
- Conclusion
Experimental setting

- **Data sets**

  | # Twitter user | 20 |
  | # tweets | 3818 |
  | # tweets having at least one named entity mention | 1721 |
  | # named entity mentions | 2918 |
  | # uncertain named entity mentions | 241 |
  | # test named entity mentions | 2677 |
  | # linkable named entity mentions | 2240 |
  | # unlinkable named entity mentions | 437 |

Table 4: A summary of the gold standard data set

- **Weight learning:**
  - Two-fold cross validation
### Experimental results

<table>
<thead>
<tr>
<th>Method</th>
<th>Linkable</th>
<th></th>
<th>Unlinkable</th>
<th></th>
<th>All</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>Accu.</td>
<td>#</td>
<td>Accu.</td>
<td>#</td>
<td>Accu.</td>
</tr>
<tr>
<td>LINDEN</td>
<td>1852</td>
<td>0.827</td>
<td>353</td>
<td>0.808</td>
<td>2205</td>
<td>0.824</td>
</tr>
<tr>
<td>LOCAL$_{\beta=0,\gamma=0}$</td>
<td>1784</td>
<td>0.796</td>
<td>355</td>
<td>0.812</td>
<td>2139</td>
<td>0.799</td>
</tr>
<tr>
<td>LOCAL$_{\gamma=0}$</td>
<td>1795</td>
<td>0.801</td>
<td>355</td>
<td>0.812</td>
<td>2150</td>
<td>0.803</td>
</tr>
<tr>
<td>LOCAL$_{\beta=0}$</td>
<td>1862</td>
<td>0.831</td>
<td>355</td>
<td>0.812</td>
<td>2217</td>
<td>0.828</td>
</tr>
<tr>
<td>LOCAL$_{full}$</td>
<td>1863</td>
<td>0.832</td>
<td>355</td>
<td>0.812</td>
<td>2218</td>
<td>0.829</td>
</tr>
<tr>
<td>KAURI$_{\beta=0,\gamma=0}$</td>
<td>1882</td>
<td>0.840</td>
<td>356</td>
<td>0.815</td>
<td>2238</td>
<td>0.836</td>
</tr>
<tr>
<td>KAURI$_{\gamma=0}$</td>
<td>1894</td>
<td>0.846</td>
<td>357</td>
<td>0.817</td>
<td>2251</td>
<td>0.841</td>
</tr>
<tr>
<td>KAURI$_{\beta=0}$</td>
<td>1913</td>
<td>0.854</td>
<td>371</td>
<td>0.849</td>
<td>2284</td>
<td>0.853</td>
</tr>
<tr>
<td>KAURI$_{full}$</td>
<td>1923</td>
<td>0.858</td>
<td>373</td>
<td>0.854</td>
<td>2296</td>
<td>0.858</td>
</tr>
</tbody>
</table>

Table 5: Experimental results over the data set

\[ p_{j,q}^i = \alpha \times Pp(r_{j,q}^i) + \beta \times Sim(r_{j,q}^i) + \gamma \times Coh(r_{j,q}^i) \]

LINDEN is the model proposed in [3] to address the task of linking entities in Web documents.

### Incremental update

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tweet index</td>
<td>11-160</td>
<td>21-170</td>
<td>31-180</td>
<td>41-190</td>
<td>51-200</td>
</tr>
<tr>
<td># added nodes</td>
<td>827</td>
<td>598</td>
<td>726</td>
<td>775</td>
<td>780</td>
</tr>
<tr>
<td># added edges</td>
<td>172296</td>
<td>139398</td>
<td>122824</td>
<td>217440</td>
<td>201368</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.853</td>
<td>0.859</td>
<td>0.859</td>
<td>0.855</td>
<td>0.858</td>
</tr>
<tr>
<td>Incremental annot. time (s)</td>
<td>35.58</td>
<td>30.22</td>
<td>28.13</td>
<td>45.36</td>
<td>42.50</td>
</tr>
<tr>
<td>Incremental annot. time per mention (ms)</td>
<td>17.98</td>
<td>15.20</td>
<td>14.03</td>
<td>22.53</td>
<td>21.08</td>
</tr>
</tbody>
</table>

Table 6: Incremental update performance of KAURI
Outline

- Motivation
- Problem Definition
- KAURI Framework
- Experiments
- Conclusion
Conclusion

- A novel problem
  - tweet entity linking

- KAURI
  - a graph-based framework that unifies
    - intra-tweet local information
    - with inter-tweet user interest information

- Good performance
  - significantly outperforms the baseline methods in terms of accuracy
  - efficient and scales well to tweet stream
Thanks!

Question?