Error Link Detection and Correction in Wikipedia

Chengyu Wang, Rong Zhang, Xiaofeng He, Aoying Zhou

School of Computer Science and Software Engineering
East China Normal University
Shanghai, China
Outline

• Introduction
• Related Work
• Proposed Approach
• Experiments
• Conclusion
Hyperlinks in Wikipedia

- The hyperlink network in Wikipedia is valuable for knowledge harvesting, entity linking, etc.
- Errors in the network structure are almost unavoidable and difficult to detect.
- Goal of this paper: detect and correct error links in Wikipedia automatically.

<table>
<thead>
<tr>
<th>Wikipedia</th>
<th>#Entities</th>
<th>#Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>3.6M</td>
<td>92M</td>
</tr>
<tr>
<td>Chinese</td>
<td>0.9M</td>
<td>11M</td>
</tr>
</tbody>
</table>
The backend is written in **Java**... *Correct!*
Introduction (2)

• **Challenges**
  – Error sparsity
    • A small number of error links v.s.10M+ Wikipedia links
  – Non-existent ground truth assumption
    • Wikipedia is treated as “ground truth” in traditional EL research.
    • No human-annotated error links are available.

• **Two-stage Approach**
  – Stage 1: generate candidate error links from Wikipedia with higher error density
  – Stage 2: predict error links and provide corrections at the same time
Outline

• Introduction
• Related Work
• Proposed Approach
• Experiments
• Conclusion
Related Work (1)

- **Entity linking (EL)**
  - Link an entity mention in text to a named entity in knowledge base
  - Methods: textual similarity, classification, learning to rank, graph-based ranking, etc.
  - Limitations
    - Wikipedia can not serve as the knowledge base for EL.
    - It is computationally costly to link all the anchor texts to Wikipedia pages.
Related Work (2)

• **Wikification**
  – Add links in documents to Wikipedia
  – A generalized task of EL

• **Error link detection in Wikipedia**
  – Pateman and Johnson’s method
    • Highlight Wikipedia linking errors by analyzing the “semantic contribution” of Wikipedia links
Outline

• Introduction
• Related Work
• Proposed Approach
• Experiments
• Conclusion
General Framework

Two-stage Approach

• **Candidate Error Link Generation**
  – Construct a dictionary $M = \{(m, E_m)\}$ containing pairs of an anchor text $m$ and its referent entity collection $E_m$
    • “Java”: Java, Java (programming language)
  – Generate candidate error link set $CL_m = \{< l_{i,j}, l_{i,j}' >\}$ containing pairs of a candidate error link $l_{i,j}$ and its most possible correction $l_{i,j}'$
    • “Java”: Facebook → Java, Facebook → Java (programming language)

• **Link Classification and Correction**
  – Train a classifier $f$ to predict whether $l_{i,j}$ is an error link and $l_{i,j}'$ is a corrected link simultaneously
    • Error link: Facebook → Java
    • Corrected link: Facebook → Java (programming language)
Candidate Error Link Generation

Dictionary and ATSN

**Dictionary Construction**
- Utilize Wikipedia to construct *ambiguous anchor text-referent entity* dictionary
  - Sources: redirect pages, disambiguation pages, hyperlinks, etc.
  - Example

<table>
<thead>
<tr>
<th>Anchor Text $m$</th>
<th>Possible Referent Entity Collection $E_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>Java</td>
</tr>
<tr>
<td>New York</td>
<td>New York City</td>
</tr>
</tbody>
</table>

**ATSN (Anchor Text Semantic Network)**
- For each *anchor text*
  - Nodes: referent entities and their neighbors
  - Links: hyperlinks between nodes

---

[Diagram of ATSN network with nodes like Java, Java applet, PHP, etc.]
Candidate Error Link Generation

**LinkRank Algorithm**

- **LinkRank**
  - A PageRank-like algorithm to assign weights to links in an ATSN
  - Weight transition:
    - Links with non-zero outdegrees: pass weights to outlinks
      \[ u_{i,j}^{(n)} = \frac{1}{|OutLink_j|} \cdot w_{i,j}^{(n-1)} \]
      - Links with zero outdegree: distribute weights to all links uniformly
  - Weight update rule
    - Transitional weights + weights from zero out-degree links
      \[ w_{i,j}^{(n)} = \sum_{l_{k,i} \in InLink_i} u_{k,i}^{(n)} + \frac{1}{|L_m|} \sum_{l_{p,q} \in L_m} w_{p,q}^{(n-1)} \]
Candidate Error Link Generation

Set Generation

- **Semantic Closeness (SC) between Two Entities in a Link**
  - An asymmetric measurement based on LinkRank
  - SC from $e_i$ to $e_j$: sum of weights of links between $e_i$ and all $e_j$'s neighbors
    \[
    SC(e_i \rightarrow e_j) = \sum_{e_j' \in \text{Neighbor}(e_j) \land l_{i,j'} \in L_m} w_{i,j'}
    \]

- **Criterion for candidate error link generation (three necessary conditions)**
  - $e_j$ and $e_{j'}$ share the same entity mention
  - $e_i$ links to $e_j$ in Wikipedia
  - Given a pre-defined threshold $\tau$, we have
    \[
    \frac{SC(e_i \rightarrow e_{j'}) - SC(e_i \rightarrow e_j)}{SC(e_i \rightarrow e_{j'})} > \tau
    \]
Link Classification and Correction
Feature Sets of a Link

- **Graph-based Features**
  - Inlink similarity
  - \( ILS(i,j) = \frac{|\text{InLinkNode}_i \cap \text{InLinkNode}_j| + 1}{|\text{InLinkNode}_i|} \)
  - Outlink similarity \( OLS(i,j) \)
  - Inlink relatedness
  - \( ILR(i,j) = \frac{\{e_k \in \text{InLinkNode}_i \mid l_{k,j} \in L_m\}}{|\text{InLinkNode}_i|} \)
  - Outlink relatedness \( OLR(i,j) \)

- **Context-based Features**
  - Context similarity \( CS(i,j) = \frac{S^T_i \cdot S_j}{\|S_i\|_2 \cdot \|S_j\|_2} \)
  - Frequent context similarity \( FCS(i,j) = \frac{FS^T_i \cdot FS_j}{\|FS_i\|_2 \cdot \|FS_j\|_2} \)
Link Classification and Correction

Pairwise Learning

• **Feature Vector Construction**
  - Feature vector of a link $l_{i,j}$
    \[ v(l_{i,j}) = < ILS(i,j), OLS(i,j), ILR(i,j), OLR(i,j), CS(i,j), FCS(i,j) > \]
  - Vector difference between two links: \[ v_S(l_{i,j}, l_{i,j'}) = v(l_{i,j}) - v(l_{i,j'}) \]
  - Feature vector of a data instance: \[ v_{PL}(l_{i,j}, l_{i,j'}) = < v(l_{i,j}), v(l_{i,j'}), v_S(l_{i,j}, l_{i,j'}) > \]
  - Example
    • Facebook → Java: 6 features
    • Facebook → Java (programming language): 6 features
    • The data instance: 6+6+6=18 features

• **Pairwise Learning**
  - Train a SVM classifier $f$ to predict whether $l_{i,j}$ is an error link and $l_{i,j'}$ is a corrected link based on $v_{PL}(l_{i,j}, l_{i,j'})$
Outline

- Introduction
- Related Work
- Proposed Approach
- Experiments
- Conclusion
Experiments (1)

- **Datasets:** English and Chinese Wikipedia dumps
- **Candidate Error Link Generation**
  - Sample candidate error links and compare the density of error links
  - Methods for comparison
    - **Simple:** extract links that connects ambiguous entities based on disambiguation pages
    - **AnchorText:** extract links with ambiguous anchor texts based on the dictionary
    - **Unweighted:** the proposed approach with uniform link weights
    - **LinkRank:** the proposed approach with varied parameter settings

<table>
<thead>
<tr>
<th>Method</th>
<th># Error links in sample set</th>
<th>Density of error links</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dataset: English Wikipedia</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simple</td>
<td>0</td>
<td>0% (approx.)</td>
</tr>
<tr>
<td>AnchorText</td>
<td>0</td>
<td>0% (approx.)</td>
</tr>
<tr>
<td>Unweighted</td>
<td>21</td>
<td>4.2%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.2))</td>
<td>28</td>
<td>5.6%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.4))</td>
<td>34</td>
<td>6.8%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.6))</td>
<td>43</td>
<td>8.6%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.8))</td>
<td>58</td>
<td>11.6%</td>
</tr>
<tr>
<td><strong>Dataset: Chinese Wikipedia</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simple</td>
<td>0</td>
<td>0% (approx.)</td>
</tr>
<tr>
<td>AnchorText</td>
<td>1</td>
<td>0.2%</td>
</tr>
<tr>
<td>Unweighted</td>
<td>17</td>
<td>3.4%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.2))</td>
<td>20</td>
<td>4.0%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.4))</td>
<td>26</td>
<td>5.2%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.6))</td>
<td>38</td>
<td>7.6%</td>
</tr>
<tr>
<td>LinkRank ((\tau = 0.8))</td>
<td>42</td>
<td>8.4%</td>
</tr>
</tbody>
</table>
**Experiments (2)**

- **Link Classification and Correction**
  - Use SVM as the classifier to train models on candidate error link sets
  - Methods for comparison (considering feature subsets)
    - PL-C: use context-based features only
    - PL-G: use graph-based features only
    - PL-Full: use both context-based and graph-based features
Experiments (3)

- Comparison between PL-Full and other methods

1. VSM: Compare content similarity based on Vector Space Model
2. EL: Link ambiguous anchor texts to referent entities in Wikipedia
3. LS: Detect incorrect links based on Wikipedia link structure
4. ELD: Use a classifier to predict error links directly (w/o pairwise learning)

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dataset: English Wikipedia</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSM based</td>
<td>VSim</td>
<td>53.2%</td>
<td>40.8%</td>
<td>46.2%</td>
</tr>
<tr>
<td></td>
<td>IntroVSim</td>
<td>57.9%</td>
<td>53.2%</td>
<td>55.5%</td>
</tr>
<tr>
<td>EL based</td>
<td>Wikify! [14]</td>
<td>45.4%</td>
<td>48.9%</td>
<td>47.1%</td>
</tr>
<tr>
<td></td>
<td>LINDEN [24]</td>
<td>46.5%</td>
<td>61.4%</td>
<td>52.9%</td>
</tr>
<tr>
<td>Error link detection based</td>
<td>LS [17]</td>
<td>71.4%</td>
<td>58.6%</td>
<td>64.4%</td>
</tr>
<tr>
<td></td>
<td>ELD</td>
<td>76.9%</td>
<td>47.3%</td>
<td>58.6%</td>
</tr>
<tr>
<td></td>
<td>PL-Full</td>
<td>83.7%</td>
<td>77.1%</td>
<td>80.3%</td>
</tr>
<tr>
<td><strong>Dataset: Chinese Wikipedia</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VSM based</td>
<td>VSim</td>
<td>50.1%</td>
<td>42.1%</td>
<td>45.8%</td>
</tr>
<tr>
<td></td>
<td>IntroVSim</td>
<td>56.3%</td>
<td>51.2%</td>
<td>53.6%</td>
</tr>
<tr>
<td>EL based</td>
<td>Wikify! [14]</td>
<td>48.2%</td>
<td>41.5%</td>
<td>44.6%</td>
</tr>
<tr>
<td></td>
<td>LINDEN [24]</td>
<td>43.8%</td>
<td>38.6%</td>
<td>41.0%</td>
</tr>
<tr>
<td>Error link detection based</td>
<td>LS [17]</td>
<td>68.5%</td>
<td>62.3%</td>
<td>65.3%</td>
</tr>
<tr>
<td></td>
<td>ELD</td>
<td>54.7%</td>
<td>39.7%</td>
<td>46.0%</td>
</tr>
<tr>
<td></td>
<td>PL-Full</td>
<td><strong>76.9%</strong></td>
<td><strong>75.6%</strong></td>
<td><strong>76.2%</strong></td>
</tr>
</tbody>
</table>
Analysis of Error Links

**Different types of ambiguity**

- **MSNE**: Multiple Senses of Named Entities
  - Error link: Josh White → Bob Gibson
  - Correction: Bob Gibson (musician)
- **MSC**: Multiple Senses of Concepts
  - Error link: Cheltenham Town F.C. → Administration (law)
  - Correction: Administration (British football)
- **ACNE**: Ambiguity Between Concepts and Named Entities
  - Error link: Tactical role-playing game → Steam
  - Correction: Steam (software)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Category of error links</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSNE</td>
</tr>
<tr>
<td>Wikipedia Error Link Set (English)</td>
<td>75.8%</td>
</tr>
<tr>
<td>Wikipedia Error Link Set (Chinese)</td>
<td>83.6%</td>
</tr>
</tbody>
</table>
## Case Studies

### English Wikipedia

<table>
<thead>
<tr>
<th>Category</th>
<th>Source Wikipage</th>
<th>Target Wikipage</th>
<th>Correct Wikipage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSNE</td>
<td>Augustus of Prima Porta&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Mars</td>
<td>Mars (mythology)</td>
</tr>
<tr>
<td></td>
<td>Josh White</td>
<td>Bob Gibson</td>
<td>Bob Gibson (musician)</td>
</tr>
<tr>
<td>MSC</td>
<td>Cheltenham Town F.C.</td>
<td>Administration (law)</td>
<td>Administration (British football)</td>
</tr>
<tr>
<td>ACNE</td>
<td>Tactical role-playing game</td>
<td>Steam</td>
<td>Steam (software)</td>
</tr>
<tr>
<td></td>
<td>Ireland in the Eurovision Song Contest 2011&lt;sup&gt;2&lt;/sup&gt;</td>
<td>Lipstick</td>
<td>Lipstick (Jedward song)</td>
</tr>
</tbody>
</table>

### Chinese Wikipedia

<table>
<thead>
<tr>
<th>Category</th>
<th>Source Wikipage</th>
<th>Target Wikipage</th>
<th>Correct Wikipage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSNE</td>
<td>Theodore Beza&lt;sup&gt;1&lt;/sup&gt; (泰奥多尔·贝扎)</td>
<td>Baden (巴登)</td>
<td>Baden (Switzerland) (巴登 (瑞士))</td>
</tr>
<tr>
<td></td>
<td>Light Rail 705 &amp; 706&lt;sup&gt;2&lt;/sup&gt; (香港轻铁705、706线)</td>
<td>Ginza Station (银座站)</td>
<td>Ginza Stop (Hong Kong) (银座站 (香港))</td>
</tr>
<tr>
<td>MSC</td>
<td>Unit sphere&lt;sup&gt;3&lt;/sup&gt; (单位球面)</td>
<td>Boundary (边界)</td>
<td>Boundary (topology) (边界 (拓扑学))</td>
</tr>
<tr>
<td>ACNE</td>
<td>Donnie Yen&lt;sup&gt;4&lt;/sup&gt; (甄子丹)</td>
<td>Hero (英雄)</td>
<td>Hero (film) (英雄 (电影))</td>
</tr>
<tr>
<td></td>
<td>Zhou Yang (actress)&lt;sup&gt;5&lt;/sup&gt; (周扬 (演员))</td>
<td>Tea house (茶馆)</td>
<td>Tea House (TV series) (茶馆 (电视剧))</td>
</tr>
</tbody>
</table>
Outline

• Introduction
• Related Work
• Proposed Approach
• Experiments
• Conclusion
Conclusion

• **Methods**
  – The two-stage approach is effective to detect and correct error links in Wikipedia.
    • Stage 1: generate candidate error links with higher density
    • Stage 2: predict error links and provide corrections at the same time

• **Analysis**
  – Most linking errors in Wikipedia are caused by multiple senses of named entities.

• **Future work**
  – Detecting error links where the correct entities is outside Wikipedia.
  – Detecting and correcting errors in other Web-scale networks.
Thanks!

Questions & Answers

* The first author would like to thank CIKM 2016 for the SIGIR student travel grant.