Automatic Disambiguation of English Puns

Paper by Tristan Miller and Iryna Gurevich (ACL 2015)
Presentation by Laura Wendlandt

http://xkcd.com/1378/
http://www.feedyouneedtoread.com/feature/5-snoopy-comic-puns/
Authors

Tristan Miller

Iryna Gurevych

https://www.ukp.tu-darmstadt.de
Word Sense Disambiguation

- **Regular WSD**: single, unambiguous meaning for each word
- **Puns WSD**: multiple valid meanings for each word
Types of Puns

• Homographic = same *written* word, different meanings

• Homophonic = same *spoken* word, different meanings
Types of Puns

- **Homographic** = same *written* word, different meanings
  - A lumberjack’s world revolves on its axes.

- **Homophonic** = same *spoken* word, different meanings
Types of Puns

- **Homographic** = same *written* word, different meanings
  - A lumberjack’s world revolves on its axes.

- **Homophonic** = same *spoken* word, different meanings
  - She fell through the window but felt no pane.
Types of Puns

- **Homographic** = same *written* word, different meanings
  - A lumberjack’s world revolves on its axes.

- **Homophonic** = same *spoken* word, different meanings
  - She fell through the window but felt no pane.

- **Both**: A political prisoner is one who stands behind her convictions.
Types of Puns

• **Homographic** = same *written* word, different meanings
  • A lumberjack’s world revolves on its axes.

• **Homophonic** = same *spoken* word, different meanings
  • She fell through the window but felt no pane.

• **Both**: A political prisoner is one who stands behind her convictions.

• **Neither**: The sign at the nudist camp read, “Clothed until April.” *(imperfect pun)*
Outline

• Previous Work
• Data Set
• Algorithms
• Results
Outline

• Previous Work

• Data Set

• Algorithms

• Results
Previous Work

• **Yokogawa (2002)** - detecting puns in Japanese text using syntactic cues

• **Taylor and Mazlack (2004)** - recognizing humorous puns in knock-knock jokes

• **Mihalcea and Strapparava (2005, 2006)** - classifying humor in text using stylistic features (alliteration, antonymy, etc.)

• **Mihalcea et al. (2010)** - choosing the most humorous punchline to a joke
Previous Data Sets

• **Bell et. al (2011)** - 373 puns from church marquees and literature, 1,515 general puns

• **Zwicky and Zwicky (1986)** - several thousand puns from advertisements, catalogues, and previously published collections

• Other smaller data sets of puns and wordplay
Outline

• Previous Work
• Data Set
• Algorithms
• Results
Data Set

- 7,750 original puns
- Manually filtered to 1,652 puns
- Manually annotated using WordNet

- Previously published corpora
- Pun of the Day website (www.punoftheday.com)
- Private collections from professional humorists

- One pun per instance
- One content word per pun
- Two meanings per pun
- Weak homography (same spelling, can have different particles and inflections)

- Two sets of sense keys per instance
- Krippendorff’s $\alpha$ for sense annotations: 0.777
Data Set

- 7,750 original puns
- Manually filtered to 1,652 puns
- Manually annotated using WordNet

- Previously published corpora
- Pun of the Day website (www.punoftheday.com)
- Private collections from professional humorists
- One pun per instance
- One content word per pun
- Two meanings per pun
- Weak homography (same spelling, can have different particles and inflections)
- Two sets of sense keys per instance
- Krippendorff’s $\alpha$ for sense annotations: 0.777
Data Set

- 7,750 original puns
  - Manually filtered to 1,652 puns
    - Manually annotated using WordNet

- One pun per instance
- One content word per pun
- Two meanings per pun
- Weak homography (same spelling, can have different particles and inflections)
- Two sets of sense keys per instance
- Krippendorff’s $\alpha$ for sense annotations: 0.777
Data Set

Resolved disagreements where possible

Removed unassignable annotations

Final 1,298 puns

- Taking intersection of contradictory sense sets
- Human adjudicator

- 3-44 words (average: 11.9)
- 2,596 total meanings
Data Set

- Resolved disagreements where possible
- Taking intersection of contradictory sense sets
- Human adjudicator

- Removed unassignable annotations

- Final 1,298 puns
  - 3-44 words (average: 11.9)
  - 2,596 total meanings
Resolved disagreements where possible

- Taking intersection of contradictory sense sets
- Human adjudicator

Removed unassignable annotations

Final 1,298 puns

- 3-44 words (average: 11.9)
- 2,596 total meanings

Pie chart showing:
- Noun: 50%
- Verb: 34%
- Adjective: 13%
- Adverb: 2%
- Multiple: 1%
Outline

• Previous Work
• Data Set
• Algorithms
• Results
Algorithms

• Simplified Lesk (Kilgarriff and Rosenzweig, 2000)

“My friend's bread shop burned down last night. Now his business is toast.”

WordNet: Toast

S: (n) toast (slices of bread that have been toasted)
S: (n) goner, toast (a person in desperate straits; someone doomed) "I'm a goner if this plan doesn't work"; "one mistake and you're toast"
Algorithms

• Simplified Lesk (Kilgarriff and Rosenzweig, 2000)

“My friend's bread shop burned down last night. Now his business is toast.”

WordNet: Toast

S: (n) toast (slices of bread that have been toasted)
S: (n) goner, toast (a person in desperate straits; someone doomed) "I'm a goner if this plan doesn't work"; "one mistake and you're toast"
Algorithms

• Simplified Extended Lesk (Ponzetto and Navigli, 2010)

• Sense definitions are concatenated with the definitions of neighboring senses
Algorithms

• Simplified Lexically Extended Lesk (Miller et al., 2012)

• Every word is expanded with up to 100 entries from a large distributional thesaurus
Tie Breakers

- **Part-of-speech tie breaker:** Select the most grammatical part of speech (assigned by Stanford POS tagger)

- **Cluster tie breaker:** Use OmegaWiki LSR to make more coarse-grained clusters of WordNet senses, choose two senses not in the same coarse-grained cluster
Outline

• Previous Work
• Data Set
• Algorithms
• Results
## Results

<table>
<thead>
<tr>
<th>system</th>
<th>C</th>
<th>P</th>
<th>R</th>
<th>F&lt;sub&gt;1&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>35.52</td>
<td>19.74</td>
<td>7.01</td>
<td>10.35</td>
</tr>
<tr>
<td>SEL</td>
<td>42.45</td>
<td>19.96</td>
<td>8.47</td>
<td>11.90</td>
</tr>
<tr>
<td>SLEL</td>
<td>98.69</td>
<td>13.43</td>
<td>13.25</td>
<td>13.34</td>
</tr>
<tr>
<td>SEL+POS</td>
<td>59.94</td>
<td>21.21</td>
<td>12.71</td>
<td>15.90</td>
</tr>
<tr>
<td>SEL+cluster</td>
<td>68.10</td>
<td>20.70</td>
<td>14.10</td>
<td>16.77</td>
</tr>
<tr>
<td>random</td>
<td>100.00</td>
<td>9.31</td>
<td>9.31</td>
<td>9.31</td>
</tr>
<tr>
<td>MFS</td>
<td>100.00</td>
<td>13.25</td>
<td>13.25</td>
<td>13.25</td>
</tr>
</tbody>
</table>

Table 1: Coverage, precision, recall, and F<sub>1</sub> for various pun disambiguation algorithms.
## Results

<table>
<thead>
<tr>
<th>POS</th>
<th>C</th>
<th>P</th>
<th>R</th>
<th>R_{rand}</th>
</tr>
</thead>
<tbody>
<tr>
<td>noun</td>
<td>66.60</td>
<td>20.89</td>
<td>13.91</td>
<td>10.44</td>
</tr>
<tr>
<td>verb</td>
<td>65.61</td>
<td>14.54</td>
<td>9.54</td>
<td>5.12</td>
</tr>
<tr>
<td>adj.</td>
<td>68.87</td>
<td>39.73</td>
<td>27.36</td>
<td>16.84</td>
</tr>
<tr>
<td>adv.</td>
<td>100.00</td>
<td>75.00</td>
<td>75.00</td>
<td>46.67</td>
</tr>
<tr>
<td>pure</td>
<td>66.77</td>
<td>21.44</td>
<td>14.31</td>
<td>9.56</td>
</tr>
<tr>
<td>mult.</td>
<td>72.58</td>
<td>18.43</td>
<td>13.38</td>
<td>12.18</td>
</tr>
</tbody>
</table>

Table 2: Coverage, precision, and recall for SEL+cluster, and random baseline recall, according to part of speech.
Conclusions

• For puns WSD, recall is comparable to MFS, but precision is much greater.

• Future work
  • Additional WSD algorithms
  • Alternative tie-breaking strategies
  • Pun detection
References


References


