A Framework for Political Portmanteau Decomposition

Nabil Hossain
nhossain@cs.rochester.edu

Minh Tran

Henry Kautz

Dept. Computer Science
University of Rochester, NY
Political Portmanteau

• Portmanteau
  • words formed by combining sounds and meanings of two words
    • brunch = breakfast + lunch    motel = motor + hotel
Political Portmanteau

• Portmanteau
  • words formed by combining sounds and meanings of two words
    • brunch = breakfast + lunch  motel = motor + hotel

• Political portmanteau (PP)
  • portmanteau in which at least one word refers to political entity
    • libtard = liberal + retard  repugnican = repugnant + republican
Political Portmanteau

• Portmanteau
  • words formed by combining sounds and meanings of two words
    • brunch = breakfast + lunch       motel = motor + hotel

• Political portmanteau (PP)
  • portmanteau in which at least one word refers to political entity
    • libtard = liberal + retard         repugnican = repugnant + republican
  • offensive; political framing
  • creative, humorous, slang, sticky
  • can be used in hate speech
Contributions

• Framework for identifying political portmanteau from the web
• Algorithm for PP detection and decomposition into root words
• First shared dataset of PP
Method

- Extract words from Reddit news comments
- Apply slang detection algorithm
- Classify the detected words into PP vs not-PP

Hossain, Nabil, Thanh Thuy Trang Tran, and Henry Kautz. "Discovering Political Slang in Readers' Comments." In ICWSM 2018.
Method

- Extract words from Reddit news comments
- Apply slang detection algorithm
- Classify the detected words into PP vs not-PP
- Decompose detected PP into root words:

Hossain, Nabil, Thanh Thuy Trang Tran, and Henry Kautz. "Discovering Political Slang in Readers' Comments." In ICWSM 2018.
Expert Annotators

---

**Method**

- Extract words from Reddit news comments
- Apply slang detection algorithm
- Classify the detected words into PP vs not-PP
- Decompose detected PP into root words:
  - $E + C \rightarrow PP$ or $C + E \rightarrow PP$

---

Model Details

- $\beta$ distribution Model — no contextual features
  - Edit distances, word length, usage frequency
  - capture sound blending and word popularity

- XGBoost — uses pre-trained GloVe word vector features from comments
  - also uses $\beta$ distribution model features

Questions: nhossain@cs.rochester.edu
Results

- $\beta$ distribution Model — no contextual features
  - Edit distances, word length, usage frequency
  - capture sound blending and word popularity
- XGBoost — uses pre-trained GloVe word vector features from comments
  - also uses $\beta$ distribution model features

<table>
<thead>
<tr>
<th>Model</th>
<th>Top1</th>
<th>Top3</th>
<th>Top5</th>
<th>Top10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>47.48</td>
<td>57.75</td>
<td>62.57</td>
<td>67.61</td>
</tr>
<tr>
<td>Beta Model (no context)</td>
<td>62.17</td>
<td>72.43</td>
<td>75.45</td>
<td>79.07</td>
</tr>
<tr>
<td>XGBoost (with context)</td>
<td>76.23</td>
<td>86.72</td>
<td>90.34</td>
<td>93.36</td>
</tr>
</tbody>
</table>

PP Decomposition Accuracy
Results

- $\beta$ distribution Model — no contextual features
  - Edit distances, word length, usage frequency
  - capture sound blending and word popularity
- XGBoost — uses pre-trained GloVe word vector features from comments
  - also uses $\beta$ distribution model features

<table>
<thead>
<tr>
<th>Model</th>
<th>Top1</th>
<th>Top3</th>
<th>Top5</th>
<th>Top10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>47.48</td>
<td>57.75</td>
<td>62.57</td>
<td>67.61</td>
</tr>
<tr>
<td>Beta Model (no context)</td>
<td>62.17</td>
<td>72.43</td>
<td>75.45</td>
<td>79.07</td>
</tr>
<tr>
<td>XGBoost (with context)</td>
<td>76.23</td>
<td>86.72</td>
<td>90.34</td>
<td>93.36</td>
</tr>
</tbody>
</table>

### PP Decomposition Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chance (always predict “not PP”)</td>
<td>66.2</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>69.5</td>
</tr>
<tr>
<td>Bi-LSTM + GloVe</td>
<td>75.1</td>
</tr>
<tr>
<td>BERT</td>
<td>78.8</td>
</tr>
<tr>
<td>XGBoost model</td>
<td>83.1</td>
</tr>
</tbody>
</table>

### PP Detection Accuracy
Results

• $\beta$ distribution Model — no contextual features
  • Edit distances, word length, usage frequency
  • capture sound blending and word popularity
• XGBoost — uses pre-trained GloVe word vector features from comments
  • also uses $\beta$ distribution model features

<table>
<thead>
<tr>
<th>Model</th>
<th>Top1</th>
<th>Top3</th>
<th>Top5</th>
<th>Top10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>47.48</td>
<td>57.75</td>
<td>62.57</td>
<td>67.61</td>
</tr>
<tr>
<td>Beta Model (no context)</td>
<td>62.17</td>
<td>72.43</td>
<td>75.45</td>
<td>79.07</td>
</tr>
<tr>
<td>XGBoost (with context)</td>
<td>76.23</td>
<td>86.72</td>
<td>90.34</td>
<td>93.36</td>
</tr>
</tbody>
</table>

PP Decomposition Accuracy

Questions: nhossain@cs.rochester.edu
Website: https://cs.rochester.edu/u/nhossain