Extracting Commonsense Properties from Embeddings with Limited Human Guidance

Property Comparison from Embeddings (PCE model)

Yiben Yang, Larry Birnbaum, Ji-Ping Wang and Doug Downey
July 18, 2018

Northwestern University
Table of contents

1. Motivation
2. Method
3. Experiment
4. Demo
Motivation
Commonsense Property Comparison Task

Is an elephant **bigger** or **smaller** than a **mouse**?
Is Ferrari more **expensive** or **cheaper** than **beer**?
Problem Definition

Three-way task:

\[ P(L|O_1, O_2, \text{Property}), L \in \{<, >, \approx\}. \]

Four-way task:

\[ P(L|O_1, O_2, \text{Property}), L \in \{<, >, \approx, \text{N/A}\}. \]
Challenges:

- **Reporting bias** [Gordon and Van Durme 2013]: Commonsense knowledge is rarely *explicitly* stated.

- Large knowledge dimensions: Property specified by adjectives: large, heavy, fast, rigid, etc. Creating training examples and building separate models on each type of property requires *expensive* labeling efforts. Handling unseen properties during the test phase (zero-shot prediction)?

- Language variation: An ideal model should be able to take flexible natural language inputs.
Can we build an efficient commonsense comparison model with **word embedding** inputs only?

I carry a **dog** around.  
I carry an **elephant** around.
Method
Figure 1: Creating a softmax regression model for each property.
Our PCE model

- Object 1 emb
- Object 2 emb
- Project
- Compare (dot product)
- Pole 1 emb
- Pole 2 emb

Additional labels:
- Big
- Heavy
- Strong
- Fast
- Small
- Light
- Breakable
- Slow
Experiment
• VERB PHYSICS (5 physical properties) [Forbes and Choi 2017]
• PROPERTY COMMON SENSE (32 commonsense properties)
Results: Supervised Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>size</td>
</tr>
<tr>
<td>Majority</td>
<td>0.51</td>
</tr>
<tr>
<td>F&amp;C</td>
<td>0.75</td>
</tr>
<tr>
<td>PCE(LSTM)</td>
<td>0.80</td>
</tr>
<tr>
<td>PCE(GloVe)</td>
<td>0.76</td>
</tr>
<tr>
<td>PCE(Word2vec)</td>
<td>0.76</td>
</tr>
</tbody>
</table>

Table 1: Supervised accuracy on the VERB PHYSICS data set. PCE outperforms the F&C model from previous work.
### Table 2: Accuracy of zero-shot learning on the VERB PHYSICS data set (using LSTM embeddings).

<table>
<thead>
<tr>
<th>Model</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>size</td>
</tr>
<tr>
<td>Random</td>
<td>0.33</td>
</tr>
<tr>
<td>Emb-Similarity</td>
<td>0.37</td>
</tr>
<tr>
<td>PCE</td>
<td>0.74</td>
</tr>
</tbody>
</table>
Results

Table 3: Accuracy on the four-way task on the PROPERTY COMMON SENSE data.

<table>
<thead>
<tr>
<th>Model</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.25</td>
</tr>
<tr>
<td>Majority Class</td>
<td>0.51</td>
</tr>
<tr>
<td>PCE(GloVe)</td>
<td>0.63</td>
</tr>
<tr>
<td>PCE(Word2vec)</td>
<td>0.67</td>
</tr>
<tr>
<td>PCE(LSTM)</td>
<td>0.67</td>
</tr>
</tbody>
</table>
Figure 3: Test accuracy as a function of the number of queried training examples. The synthesis approach performs best.
Active Learning

Figure 4: The uncertainty measure of each queried training example. As training proceeds, the synthesis approach continues to select more uncertain examples.
Demo
http://thor.cs.northwestern.edu:1959/