Integrating Distributional Lexical Contrast into Word Embeddings for Antonym–Synonym Distinction

Kim Anh Nguyen, Sabine Schulte im Walde, Ngoc Thang Vu

Institute for Natural Language Processing (IMS)
University of Stuttgart

ACL-2016, Berlin
Overview

1. Introduction
   1.1. Word Vector Representations
   1.2. Antonym-Synonym Distinction Task

2. Contributions
   2.1. Improving Weights of Feature Vectors
   2.2. Distributional Lexical Contrast Embeddings Model

3. Experiments

4. Conclusion
1.1. Word Vector Representations

1.1.1. Distributional Semantic Model (DSM)

- A means to represent meaning vectors of words.
- DSM rely on the *distributional hypothesis*. (Harris, 1954)
- Words with similar distributions have related meanings.
- Each weighted feature can be:
  - Co-occurrence frequency.
  - Association measure: local mutual information (LMI) (Evert, 2005)

1.1.2. Word Embeddings

- Representing words as low-dimensional dense vectors.
- Words with similar distributions have similar vectors.
1.2. Antonym-Synonym Distinction Task

- **Goal**
  - Distinguishing antonyms from synonyms.

- **Problems**
  - DSM tend to capture both antonyms (formal–informal) and synonyms (formal–conventional).
  - Word embeddings represent vectors of both antonyms and synonyms as similar vectors.

- **Causes**
  - Antonymy and synonymy are paradigmatic relations.
  - Antonyms and synonyms often occur in similar contexts.
Outline

1. Introduction
   1.1. Word Vector Representations
   1.2. Antonym-Synonym Distinction Task

2. Contributions
   2.1. Improving Weights of Feature Vectors
   2.2. Distributional Lexical Contrast Embeddings Model

3. Experiments

4. Conclusion
2.1. Improving Weights of Feature Vectors

- **Goal:**
  - Improving the quality of weighted feature vectors.

- **Solution:**
  - Strengthening most salient features in the vectors.
  - Using the lexical contrast information of the target words and their contexts.
  - Proposing the new weight for feature vectors.

- Representing words based on DSM with positive LMI.

- For each target word $w$:
  - Determining the sets of antonyms $A(w)$ and synonyms $S(w)$.
  - Determining the set of shared words $W(f)$ for each feature $f$.

- Computing the new weight (called $\text{weight}^{SA}$) as follows:
2.1. Improving Weights of Feature Vectors

\[
\text{weight}^{SA}(w,f) = \frac{1}{\#(w,u)} \sum_{u \in W(f) \cap S(w)} \text{sim}(w, u)
\]

\[
- \frac{1}{\#(w',v)} \sum_{w' \in A(w)} \sum_{v \in W(f) \cap S(w')} \text{sim}(w', v)
\]
2.1. Improving Weights of Feature Vectors

- $w = \text{"formal"}$
  - $S(w) = \{\text{methodical, precise, conventional, ...}\}$

- $A(w) = w' = \text{"informal"}$
  - $S(w') = \{\text{irregular, unofficial, unconventional, ...}\}$

- $f = \text{"conception"}$

- $f = \text{"issue"}$

- $f = \text{"rumor"}$
2.1. Improving Weights of Feature Vectors

\[ w = \text{“formal”} \]
\[ S(w) = \{ \text{methodical, precise, conventional, …} \} \]

\[ A(w) = w' = \text{“informal”} \]
\[ S(w') = \{ \text{irregular, unofficial, unconventional, …} \} \]

\[ f = \text{“conception”} \]
\[ f = \text{“issue”} \]
\[ f = \text{“rumor”} \]

\[ \text{weight}^{SA}(w, f) = \frac{1}{\#(w, u)} \sum_{u \in W(f) \cap S(w)} \text{sim}(w, u) - \frac{1}{\#(w', v)} \sum_{w' \in A(w)} \sum_{v \in W(f) \cap S(w')} \text{sim}(w', v) \]
2.2. Distributional Lexical Contrast Embeddings Model (dLCE)

- **Aims:**
  - Learning word embeddings.
  - Moving synonyms closer to each other in space.
  - Moving antonyms further away from each other in space.

- **Solution:**
  - Integrating distributional lexical contrast into the transformation of Skip-gram model (Mikolov et al. 2013, Levy et al., 2014).
  - Applying lexical contrast to every single context of the target word.

- The proposed objective function as follows:
2.2. Distributional Lexical Contrast Embeddings Model (dLCE)

\[
\sum_{w \in V} \sum_{c \in V} \left\{ \left( \#(w,c) \log \sigma(sim(w,c)) \right) + k \#(w)P_0(c) \log \sigma(-sim(w,c)) \right\} + \left( \frac{1}{\#(w,u)} \sum_{u \in W(c) \cap S(w)} sim(w,u) - \frac{1}{\#(w,v)} \sum_{v \in W(c) \cap A(w)} sim(w,v) \right) \}
\]
Outline

1. Introduction
   1.1. Word Vector Representations
   1.2. Antonym-Synonym Distinction Task

2. Contributions
   2.1. Improving Weights of Feature Vectors
   2.2. Distributional Lexical Contrast Embeddings Model

3. Experiments

4. Conclusion
3. Experiments

- Evaluating $\text{weight}^{SA}$ on Antonym-Synonym distinction task.
- Evaluating effects of dLCE model:
  - Antonym-Synonym distinction task.
  - Similarity task.
3.1. Antonym–Synonym Distinction

- Corpus: ENCOW14A (Schäfer and Bildhauer, 2012) contains 14.5 billion tokens.
- Dataset: a gold standard resource of paradigmatic relation pairs (Roth and Schulte im Walde, 2014)

<table>
<thead>
<tr>
<th>Word Class</th>
<th>Ant-pairs</th>
<th>Syn-pairs</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjective</td>
<td>300</td>
<td>300</td>
<td>600</td>
</tr>
<tr>
<td>Noun</td>
<td>350</td>
<td>350</td>
<td>700</td>
</tr>
<tr>
<td>Verb</td>
<td>400</td>
<td>400</td>
<td>800</td>
</tr>
</tbody>
</table>

- Using average precision (AP) to evaluate.
- Using box-plots to compare the cosine medians of antonymous vs. synonymous pairs.
3.1. Antonym–Synonym Distinction

- AP evaluation results\(^1\):

<table>
<thead>
<tr>
<th></th>
<th>Adjectives</th>
<th></th>
<th>Nouns</th>
<th></th>
<th>Verbs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANT</td>
<td>SYN</td>
<td>ANT</td>
<td>SYN</td>
<td>ANT</td>
<td>SYN</td>
</tr>
<tr>
<td>LMI</td>
<td>0.46</td>
<td>0.56</td>
<td>0.42</td>
<td>0.60</td>
<td>0.42</td>
<td>0.62</td>
</tr>
<tr>
<td>weight(^{SA})</td>
<td>0.36**</td>
<td>0.75**</td>
<td>0.40</td>
<td>0.66</td>
<td>0.38*</td>
<td>0.71*</td>
</tr>
<tr>
<td>LMI + SVD</td>
<td>0.46</td>
<td>0.55</td>
<td>0.46</td>
<td>0.55</td>
<td>0.44</td>
<td>0.58</td>
</tr>
<tr>
<td>weight(^{SA}) + SVD</td>
<td>0.36***</td>
<td>0.76***</td>
<td>0.40*</td>
<td>0.66*</td>
<td>0.38***</td>
<td>0.70***</td>
</tr>
</tbody>
</table>

\(^1\) \(\chi^2\), *** \(p < .001\), ** \(p < .005\), * \(p < .05\)
3.1. Antonym–Synonym Distinction

- Results in box-plots:
3.2. Effects of dLCE model

3.2.1. Antonym-Synonym Distinction:

- Dataset: the gold standard resource of paradigmatic relation pairs (Roth and Schulte im Walde, 2014).
- Using area under curve (AUC) to identify antonyms.
- Comparison Models: Skip-gram (SGNS), mLCM (Pham et al., 2015)

- Results:

<table>
<thead>
<tr>
<th></th>
<th>Adjectives</th>
<th>Nouns</th>
<th>Verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGNS</td>
<td>0.64</td>
<td>0.66</td>
<td>0.65</td>
</tr>
<tr>
<td>mLCM</td>
<td>0.85</td>
<td>0.69</td>
<td>0.71</td>
</tr>
<tr>
<td>dLCE</td>
<td><strong>0.90</strong></td>
<td><strong>0.72</strong></td>
<td><strong>0.81</strong></td>
</tr>
</tbody>
</table>
3.2. Effects of dLCE model

3.2.2. Similarity Task:

- Dataset: SimLex-999 (Hill et al., 2015)
- Using Spearman correlation coefficient $\rho$ to evaluate.
- Comparison Models: SGNS, mLCT.
- Results:

<table>
<thead>
<tr>
<th></th>
<th>SGNS</th>
<th>mLCT</th>
<th>dLCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results</td>
<td>0.38</td>
<td>0.51</td>
<td>0.59</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
   1.1. Word Vector Representations
   1.2. Antonym-Synonym Distinction Task

2. Contributions
   2.1. Improving Weights of Feature Vectors
   2.2. Distributional Lexical Contrast Embeddings Model

3. Experiments

4. Conclusion
4. Conclusion

- We have presented two methods to address the task of antonym-synonym distinction:
  - Improving the quality of weighted feature vectors.
  - Integrating distributional lexical contrast into word embeddings.
- The results from the experiments show that our approaches can model semantic similarity and distinguish between antonyms and synonyms.
Thank you!