Learning Semantic Similarity by Selecting Random Word Subsets

Artem Sokolov
LIMSI-CNRS Orsay, France
gpsasha@gmail.com

Highlights

**Semantic Textual Similarity** learning between sentences
- degrees of semantic equivalence: 5 (exactly the same meaning) → 0 (nothing in common)
- task:
  - given 2 sentences
  - predict the degree of semantic equivalence ∈ [0, 5]
- evaluation: Pearson’s correlation
- contributions:
  - improved Random Indexing (RI):
    - selection of informative projection directions
    - learning-to-rank + boosting approach
  - no special preprocessing, no linguistic resources
  - in top-25% among 89 participants (SemEval’12/Task6)

Symmetric Vector Transformations

Combine 2 RI sentence vectors \( v(s_1), v(s_2) \) in a symmetric manner:
- ‘sumdiff’: \( \bar{x} = (v(s_1) + v(s_2), sgn(v_1(s_1) - v_1(s_2))(v_1(s_1) - v_1(s_2))) \)
- ‘concat’: \( \bar{x} = (v(s_1), v(s_2)) \)
- ‘product’: \( x_{ij} = v_i(s_1) \cdot v_j(s_2) \)
- ‘crossprod’: \( x_{ij} = v_i(s_1) - v_j(s_2) \)
- ‘absdiff’: \( x_{ij} = |v_i(s_1) - v_j(s_2)| \)

Performance on Training Data

<table>
<thead>
<tr>
<th>learner</th>
<th>transform</th>
<th>correlation</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>pure RI</td>
<td>cos</td>
<td>0.264</td>
<td>0.005</td>
</tr>
<tr>
<td>logistic reg.</td>
<td></td>
<td>0.508</td>
<td>0.041</td>
</tr>
<tr>
<td>logistic reg.</td>
<td>concat</td>
<td>0.537</td>
<td>0.052</td>
</tr>
<tr>
<td>RankBoost</td>
<td>sumdiff</td>
<td>0.685</td>
<td>0.027</td>
</tr>
<tr>
<td></td>
<td>product</td>
<td>0.663</td>
<td>0.018</td>
</tr>
<tr>
<td></td>
<td>crossprod</td>
<td>0.648</td>
<td>0.028</td>
</tr>
<tr>
<td></td>
<td>crossdiff</td>
<td>0.643</td>
<td>0.023</td>
</tr>
<tr>
<td></td>
<td>concat</td>
<td>0.625</td>
<td>0.025</td>
</tr>
<tr>
<td></td>
<td>absdiff</td>
<td>0.602</td>
<td>0.021</td>
</tr>
<tr>
<td>RtRank</td>
<td>sumdiff</td>
<td>0.730</td>
<td>0.020</td>
</tr>
<tr>
<td></td>
<td>product</td>
<td>0.721</td>
<td>0.023</td>
</tr>
</tbody>
</table>

- selected transformations for the submission in **bold**
- optimal parameters for RankBoost & RtRank found with cross-validation on 5 folds

Classic Random Indexing

**Distributional hypothesis**: words with similar meanings tend to appear in similar contexts
- map words \( w \) to context vectors \( v(w) \):
  1. initialize with ternary sparse vectors
  2. sum over contexts
- 3-4: normalize for frequency and sentence length
- \( \approx \): similar contexts ⇒ similar vectors
- \( \leq \): dimension reduction by random projections
- \( \leq \): ternary turns on/off random word sets \( w \)
- drawbacks:
  - agnostic of the modeled semantic relation
  - trial-and-error tuning to specific task
  - all (even useless) random sets have effect
  - adaption to different tasks needed!

CONCLUSION & EXPERIMENTAL RESULTS

**RI extension**: selects word sets relevant for semantics learning
- learns semantic order, not absolute values
- promising results, despite ignoring preprocessing & linguistics
- underperforms on SMT sets: possible incorrect lexical choice in MT output should be trained separately

**Contrib.: Selecting Word Subsets with Boosting to Rank by Semantic Similarity**

**Idea**:
- exploit ordering of sentence pairs under \( y \)
- use boosting to select important elements of \( \bar{x} \) \( \Leftarrow \) select random word sets relevant to the similarity modeled

**Given**:
- training sentence pair: \( s_1, s_2 \) & similarity: \( y \)

**Want to Learn**:
- similarity score \( H(\bar{x}) = \sum_{i=1}^T \alpha_i h_i(\bar{x}), \) where
  - \( \bar{x} \) = some symmetric transform of sentence vectors \( v(s_1), v(s_2) \) (see frame to the left)
  - \( h_i(\bar{x}) \) = simple (weak) functions of \( \bar{x} \), \( \alpha_i \) = weights
  - the more similar \( v(s_1) \) is to \( v(s_2) \), the higher should be \( H(\bar{x}) \)

**Ranking losses & Weak functions** \( h_i \):
- **RankBoost** [Freund et al.2003]
  - loss: \( \sum_{(x_i,y_i) \leq (x,y)} P(i, j)[H(x_i) \geq H(x_j)] \)
  - weak functions: \( h(x; \theta, k) = [x_k > \theta] \) (decision stump)
  - weights \( P(i, j) = y'_i - y'_j \)
- **RtRank** [Zheng et al.2007, Mohan et al.2011]
  - loss: \( \sum_{(x_i,y_i) \leq (x,y)} (\max(0, H(x_i) - H(x_j)))^2 \)
  - weak functions: decision trees of depth 4
- Well-known algorithms, implementations available online