Probabilistic Soft Logic for Semantic Textual Similarity

Islam Beltagy, Katrin Erk, Raymond Mooney
The University of Texas at Austin

Abstract

- Using probabilistic logic for semantic representation, combining Formal Semantics and Distributional Semantics
- PSL is the probabilistic logic framework we use
- Evaluate on the STS task: judge sentence similarity

Task

- Compute the Semantic Similarity score of the given sentences

Formal Semantics

- Represent semantics using formal languages like FOL
- Handle many complex semantic phenomena like embedded propositions, logical operators, and quantifiers
- Unable to handle uncertain knowledge

Distributional Semantics

- Represent words as vectors in high dimensional space
- Capture the "graded" nature of linguistic meaning, but do not adequately capture logical structure
- Similarity \( \text{sim}(\text{water}, \text{tea}) = \cos(\text{water}, \text{tea}) \)

System Architecture

- Boxer: Semantic analysis tool maps sentences to logical form
- PSL inference: Computes: \( P(\text{S1} | \text{S2}, \text{RB}) \), \( P(\text{S2} | \text{S1}, \text{RB}) \)
- Rule Base (RB): A set of weighted rules representing background knowledge
- Boxer: Computes: \( f(t) = \frac{1}{2} \sum \lambda_n |d(t)|^p \)
- MPE inference is a linear program when \( p = 1 \)
- Distributional Semantics: Robust but shallow
- PSL is the probabilistic logic framework we use
- Atoms have continuous truth values in interval \([0,1]\) (in contrast with boolean atoms in MLNs)
- Efficient inference: 100 times faster than MLN in our experiments
- Logical operators are replaced with Łukasiewicz logic:
  \( I(\neg \ell_1) = 1 - I(\ell_1) \)
  \( I(\ell_1 \land \ell_2) = \max \{0, I(\ell_1) + I(\ell_2) - 1\} \)
  \( I(\ell_1 \lor \ell_2) = \min \{1, I(\ell_1) + I(\ell_2)\} \)
- Implication via distance to satisfaction:
  \( d(\ell_1 \rightarrow \ell_2) = \max \{0, I(\ell_1) \land I(\ell_2)\} \)
- PDF: \( \text{VMP} \) inference is a linear program when \( p = 1 \)
- Implication via distance to satisfaction:
  \( d(\ell_1 \rightarrow \ell_2) = \max \{0, I(\ell_1) \land I(\ell_2)\} \)
- Implication via distance to satisfaction:
  \( d(\ell_1 \rightarrow \ell_2) = \max \{0, I(\ell_1) \land I(\ell_2)\} \)

Heuristic Grounding

- Evidence: \( I(\text{man(M)}) = 1 \), \( I(\text{drive(D)}) = 0.8 \)
- Constants: \( M, D \)
- Rule: \( \text{man}(x) \land \text{drive}(y) \land \text{agent}(y, x) \rightarrow \text{result}(\) 

Evaluation

<table>
<thead>
<tr>
<th>System</th>
<th>msr-vid</th>
<th>msr-par</th>
<th>SICK</th>
</tr>
</thead>
<tbody>
<tr>
<td>vec-add</td>
<td>0.78</td>
<td>0.24</td>
<td>0.65</td>
</tr>
<tr>
<td>MLN</td>
<td>0.63</td>
<td>0.16</td>
<td>0.47</td>
</tr>
<tr>
<td>PSL</td>
<td>0.79</td>
<td>0.53</td>
<td>0.70</td>
</tr>
<tr>
<td>PSL+vec-add</td>
<td>0.83</td>
<td>0.49</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Conclusion

1) Probabilistic logic in general is a promising semantic representation
2) PSL fits the STS task better than MLNs. It is faster and more accurate

References