Inducing Crosslingual Distributed Representations of Words

Alexandre Klementiev  Ivan Titov  Binod Bhattacharjya
{aklement, titov, bhattara}@mmci.uni-saarland.de

**Motivation**

With large vocabularies and limited annotated data treating words as atomic symbols often means poor model estimates. Instead, we can induce alternative representations from cheap unsupervised data and use them instead of words:

<table>
<thead>
<tr>
<th>Clustering</th>
<th>Vector space</th>
<th>Distributed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign words to (hierarch.) clusters</td>
<td>Words defined by context</td>
<td>○ Vector space + probabilistic models</td>
</tr>
<tr>
<td>○ How to choose granularity?</td>
<td>Algo. induced</td>
<td>○ Dense embedding</td>
</tr>
<tr>
<td>Many clusterings possible</td>
<td>Not learned for a given task</td>
<td>Learned (for a given task)</td>
</tr>
</tbody>
</table>

Inducing the same representation for a pair of languages has additional benefits for low resource languages. We can learn in one language where annotation is available and apply to the other language directly.

**Our Contribution**

A general multitask learning (MTL) inspired framework to induce crosslingual distributed representations.

- Treat words as individual tasks
- Task relatedness is derived from co-occurrence statistics in bilingual parallel data

**Motivation**

The goal of Multitask Learning (MTL) is to improve generalization performance across a set of tasks by learning them jointly:

The MTL setup of Cavalanti et al. (2010):
- Consider $k$ tasks
- Learn a linear classifier parameterized by $c_k$ for each task ($c$ is the concatenated parameter vector)
- Minimize the following objective:

$$ L(c) = \sum_{k=1}^{K} L_k(c_k) + \frac{1}{2} c^T (A \otimes I_m)c $$

- Interaction matrix $A$ encodes task "relatedness"

**Crosslingual Representation Induction**

Idea: frame crosslingual distributed representation induction as multi-task learning

- Treat words in both languages as individual tasks
- Define $A$ by how often words align in parallel data
- Use the regularizer from the MTL objective
- Applicable to any distributed representation induction set-up
- In this work, we apply it to the neural probabilistic language model [Bengio et al. (2003)]

**Evaluation**

- Induced 40-dimensional representation of words in German and English
- Used RCV1/2 monolingual corpora (~8 million tokens in each language)
- Used Europarl parallel data to define the interaction matrix
- Crosslingual document classification (4 RCV document topics):
  - Train on annotated data in one language
  - **DistribReps**: test on second language directly with no additional training
  - **Glossed**: translate test docs into the original language (most frequently aligned words)
  - **MT**: same but with phrase-based MT
  - Results are likely to improve with increased embedding dimensionality

**Encoding prior knowledge into the interaction matrix:**

- Represent tasks with an undirected weighted graph

$$ H = (R, E, S) $$

- The graph Laplacian is defined as:

$$ L_{i,j}(H) = \begin{cases} 
\sum_{c(k) \in E} s(i, k) & \text{if } i = j \\
- \sum_{c(k) \in E} s(i, k) & \text{if } (i, j) \in E \\
0 & \text{otherwise}
\end{cases} $$

- $A^{-1}$ defines the degree of "relatedness" between tasks
- $A$ is invertible ($J$ is positive semi-definite)

**The work was supported by the Cluster of Excellence and a Google research award.**