Semi-automatic WordNet Linking using Word Embeddings

Kevin Patel, Diptesh Kanojia and Pushpak Bhattacharyya
Presented by: Ritesh Panjwani

January 11, 2018
Introduction

- **Wordnet**
  - Lexical resource
  - Groups words into sets of synonyms called Synsets
  - Records relations among these synsets

- **Linked Wordnet**
  - Synsets with same meaning, but belonging to wordnets of different languages are linked

- **Challenge in linking Wordnets**
  - Linking done manually
  - Tools such as Joshi et al. (2012b) to assist humans
Background

- Princeton WordNet (Miller et al., 1990) or the English WordNet was the first wordnet.
- EuroWordNet (Vossen and Letteren, 1997) is a linked wordnet comprising of wordnets for European languages.
  - Each wordnet separately captures a language-specific information.
  - Wordnets use Princeton WordNet as an Inter-Lingual-Index.
  - Enables one to go from concepts in one language to similar concepts in any other language.
- IndoWordNet Bhattacharyya (2010) is a linked wordnet comprising of wordnets for 18 Indian languages.
  - Created using the expansion approach using Hindi WordNet as a pivot.
  - Partially linked to English WordNet.
Joshi et al. (2012a) developed a heuristic based measure where they use bilingual dictionaries to link two wordnets. Combine scores using various heuristics and generate a list of potential candidates for linked synsets.

Singh et al. (2016) discuss a method to improve the current status of Hindi-English linkage and present a generic methodology. Their method is beneficial for culture-specific synsets, or for non-existing concepts. Cost and time inefficient; requires a lot of manual effort on the part of a lexicographer.

Our intention: reduce effort on the part of lexicographers
Problem Statement

Given wordnets of two different languages $E$ and $F$ with sets of synsets $\{s^1_E, s^2_E, \ldots, s^m_E\}$ and $\{s^1_F, s^2_F, \ldots, s^n_F\}$ respectively, find mappings of the form $<s^i_E, s^j_F>$ which are semantically correct.

<table>
<thead>
<tr>
<th>Hindi Synsets</th>
<th>English Synsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: {हाथ, हस्त, कर}</td>
<td>1: {hand, paw}</td>
</tr>
<tr>
<td>2: {tax, revenue enhancement}</td>
<td></td>
</tr>
</tbody>
</table>
Motivation

Adapted from Mikolov et al. (2013a)

\[
y = W \hat{x}
\]
Algorithm: Notations

- Let $E$ and $F$ be two languages
- Let $|E|$ and $|F|$ be the number of synsets in wordnets of $E$ and $F$ respectively
- Let $s^i_E$ and $s^j_F$ be the $i^{th}$ and $j^{th}$ synsets of $E$ and $F$ respectively,
  - $s^i_E = \{e^1_\alpha, e^2_\alpha, \ldots, e^{m_i}_\alpha\}$
  - $s^j_F = \{f^1_\beta, f^2_\beta, \ldots, f^{n_j}_\beta\}$
- $e^p_\alpha$ and $f^q_\beta$ are words in vocabulary of $E$ and $F$ respectively for $1 \leq p \leq m_i$ and $1 \leq q \leq n_j$, and $1 \leq i \leq |E|$ and $1 \leq j \leq |F|
- Let $v_\alpha e^p_\alpha$ be the word vector corresponding to $e^p_\alpha$
Algorithm: Training

- Estimate $v_{s_i}^E$ as

$$v_{s_i}^E = \frac{1}{m_i} \sum_{p=0}^{m_i} v_{e_p}^\alpha$$  \hspace{1cm} (1)

- Similarly,

$$v_{s_j}^F = \frac{1}{n_j} \sum_{q=0}^{n_j} v_{f_q}^\beta$$  \hspace{1cm} (2)

- Given links of the form $\langle s_i^E, s_j^F \rangle$, we learn $W$ such that the error $Err$

$$Err = \| W \cdot v_{s_i}^E - v_{s_j}^F \|^2$$  \hspace{1cm} (3)

is minimized.
Algorithm: Prediction

- To find a mapping for a new synset $s^k_E$, one needs to:
  - Calculate $v' = W.v_{s^k_E}$
  - Find $v_{s^l_F}$ such that $v_{s^l_F}.v'$ is maximized
  - Create link $\langle s^k_E, s^l_F \rangle$
Datasets

- Linking Hindi WordNet to English WordNet
- **English Vectors**: Pretrained vectors from Google’s word2vec tool Mikolov et al. (2013b), trained on News dataset (around 100 billion tokens)
- **Hindi Vectors**: Trained using word2vec on Bojar corpus Bojar et al. (2014) (around 365 million tokens)
- Linked data: Created at CFILT, IITB
  - Of the form $\langle$ hindi_synset_id, english_synset_id, link_type $\rangle$, where $link_type \in \{DIRECT, HYPERNYMY, etc.\}$
  - Focus on only DIRECT links
  - 6863 such links available
## Distribution of links

<table>
<thead>
<tr>
<th>Class</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>4757</td>
</tr>
<tr>
<td>Adjective</td>
<td>1283</td>
</tr>
<tr>
<td>Verb</td>
<td>680</td>
</tr>
<tr>
<td>Adverb</td>
<td>143</td>
</tr>
</tbody>
</table>

Distribution of available links among various classes
Accuracy@n: One of the top $n$ predictions can be correct

<table>
<thead>
<tr>
<th>True label</th>
<th>Prediction1</th>
<th>Accuracy @1</th>
<th>Accuracy @3</th>
<th>Accuracy @5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prediction2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prediction3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prediction4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Prediction5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results: Overall

<table>
<thead>
<tr>
<th></th>
<th>Acc@1</th>
<th>Acc@3</th>
<th>Acc@5</th>
<th>Acc@8</th>
<th>Acc@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>0.29</td>
<td>0.45</td>
<td>0.52</td>
<td>0.58</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Results for the overall setting: Dimension of English embeddings=300, Dimensions of Hindi embeddings=300
### Results: Per word class 1

<table>
<thead>
<tr>
<th>Word Class</th>
<th>Acc@1</th>
<th>Acc@3</th>
<th>Acc@5</th>
<th>Acc@8</th>
<th>Acc@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>0.35</td>
<td>0.53</td>
<td>0.60</td>
<td>0.65</td>
<td>0.67</td>
</tr>
<tr>
<td>Adjective</td>
<td>0.26</td>
<td>0.44</td>
<td>0.50</td>
<td>0.57</td>
<td>0.60</td>
</tr>
<tr>
<td>Verb</td>
<td>0.15</td>
<td>0.25</td>
<td>0.29</td>
<td>0.33</td>
<td>0.37</td>
</tr>
<tr>
<td>Adverb</td>
<td>0.28</td>
<td>0.51</td>
<td>0.59</td>
<td>0.70</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Results for the setting: Dimension of English Vectors=300, Dimensions of Hindi Vectors=300
## Results: Per word class II

<table>
<thead>
<tr>
<th>Word Class</th>
<th>Acc@1</th>
<th>Acc@3</th>
<th>Acc@5</th>
<th>Acc@8</th>
<th>Acc@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>0.35</td>
<td>0.51</td>
<td>0.58</td>
<td>0.64</td>
<td>0.66</td>
</tr>
<tr>
<td>Adjective</td>
<td>0.12</td>
<td>0.20</td>
<td>0.24</td>
<td>0.30</td>
<td>0.32</td>
</tr>
<tr>
<td>Verb</td>
<td>0.17</td>
<td>0.27</td>
<td>0.32</td>
<td>0.36</td>
<td>0.39</td>
</tr>
<tr>
<td>Adverb</td>
<td>0.38</td>
<td>0.52</td>
<td>0.65</td>
<td>0.76</td>
<td>0.80</td>
</tr>
</tbody>
</table>

Results for the setting: Dimension of English Vectors=300, Dimensions of Hindi Vectors=1200
Possible reasons for poor performance

- Something is fundamentally missing in word vectors. Probably presence of only co-occurrence information, and lack of other information such as word ordering, argument frames (for verbs), etc.
- The approach to create synset vectors is not optimal.
- The linear transformation approach is not optimal.
- Synset members are often phrases instead of words. How to create phrase vectors?
- Currently, a word has only one vector. That is one of the reasons for ambiguity. Perhaps for each word, multiple vectors (one vector per sense) is the way to go.
Possible reasons for poor performance

- Something is fundamentally missing in word vectors. Probably presence of only co-occurrence information, and lack of other information such as word ordering, argument frames (for verbs), etc.
Possible reasons for poor performance

- Something is fundamentally missing in word vectors. Probably presence of only co-occurrence information, and lack of other information such as word ordering, argument frames (for verbs), etc.
- The approach to create synset vectors is not optimal.
**Discussion**

Possible reasons for poor performance

- Something is fundamentally missing in word vectors. Probably presence of only co-occurrence information, and lack of other information such as word ordering, argument frames (for verbs), etc.
- The approach to create synset vectors is not optimal.
- The linear transformation approach is not optimal.
Possible reasons for poor performance

- Something is fundamentally missing in word vectors. Probably presence of only co-occurrence information, and lack of other information such as word ordering, argument frames (for verbs), etc.
- The approach to create synset vectors is not optimal.
- The linear transformation approach is not optimal.
- Synset members are often phrases instead of words. How to create phrase vectors?
Possible reasons for poor performance

- Something is fundamentally missing in word vectors. Probably presence of only co-occurrence information, and lack of other information such as word ordering, argument frames (for verbs), etc.
- The approach to create synset vectors is not optimal.
- The linear transformation approach is not optimal.
- Synset members are often phrases instead of words. How to create phrase vectors?
- Currently, a word has only one vector. That is a one of the reason for ambiguity. Perhaps for each word, multiple vectors (one vector per sense) is the way to go.
Conclusion and Future Work

- Described an approach to link wordnets
- Creates synset embeddings using word embeddings, followed by learning transformation from source to target language synsets
- Our approach achieves accuracy@10 of approximately 60% and 70% of all synsets and noun synsets, respectively
- Discussed reasons for poor performance on classes such as verbs
- Plan to integrate it in tools such as Joshi et al. (2012a)
References


References


Questions?
For more details, write to: kevin.patel@cse.iitb.ac.in