Sub-Word Similarity based Search for Embeddings: Inducing Rare-Word Embeddings for Word Similarity Tasks and Language Modelling

Mittul Singh  Clayton Greenberg  Youssef Oualil  Dietrich Klakow
firstname.lastname@lev.uni-saarland.de

Spoken Language Systems, Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

### Rare Words

Rare words are words which occur with low frequency. Examples are:
- out-of-vocabulary (OOV) words
- words with a frequency one (RW)

Problem learning good word embeddings for such words

- For low occurrence languages, there isn’t enough rare-word data.
- Hence, handling such words in a predictive system is difficult.

### Inducing Rare-Word Embeddings

**Steps**

1. **Map** non-rare words to sub-word units
2. **Index** non-rare words using sub-word units
3. **Search** for matches of a rare word
4. **Combine** matches to form a rare-word embedding

**Map**

- Map all non-rare words to their sub-word units
- Example: \( D_i(\text{language}) = \{ \text{lan}, \text{ang}, \text{ngu}, \text{gua}, \text{ug}, \text{age} \} \)

**Index**

- Create an inverted index of list words per sub-word unit
- Example: \( \text{ded} = \{ \text{padded}, \text{dedifferentiation}, \text{decided} \ldots \} \)
- Use a search engine library to do indexing, e.g. Lucene

**Search**

- Map rare-word (\( w' \)) to its sub-word units
- Example: \( D_i(\text{globalise}) = \{ \text{glo}, \text{lab}, \text{oba}, \text{bal}, \text{ali}, \text{lis}, \text{ise} \} \)
- Query the index for most relevant matches of these sub-word units
- Returns a ranked list of words (\( R^k(w') \))

**Combine**

Combine matches’ embeddings (\( v_w \)) to form rare-word embedding (\( v_{w'} \))

\[
 v_{w'} = \sum_{w \in R^k(w')} S(w, w') \times v_w
\]

where, \( S \) is used to calculate similarity score between the rare word and its matched word. Following types of function were used:

- Sequence Specific: Jaro Similarity (jaro), Jaro-Winkler Similarity (jw), Subsequence Kernels (ssk)
- Bag-of-Char: Jaccard Coefficient (jc), Most frequent K Characters (mkf), Tversky Coefficient (tc)
- Default weighting: \( S(w, w') = 1 \) (labelled with subscript 1)

**Table**

<table>
<thead>
<tr>
<th>Language</th>
<th>Train V</th>
<th>RW</th>
<th>#ENF</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>1000K</td>
<td>37K</td>
<td>16K</td>
<td>13K</td>
</tr>
<tr>
<td>Tagalog</td>
<td>585K</td>
<td>22K</td>
<td>11K</td>
<td>8K</td>
</tr>
<tr>
<td>Turkish</td>
<td>239K</td>
<td>25K</td>
<td>14K</td>
<td>10K</td>
</tr>
<tr>
<td>Vietnamese</td>
<td>985K</td>
<td>6K</td>
<td>1K</td>
<td>305</td>
</tr>
</tbody>
</table>

**Table:** This table reports various statistics for different corpora

**Table:** This table reports various statistics for different corpora used for language modelling.

- Column labelled **Language** shows four corpora
- **Train** shows the training set size in thousands of tokens
- **V** reports the vocabulary size for the various corpora
- **RW** shows the number of rare words (\( RW = \text{OOVs and RW1} \))
- **#ENF** shows the number of rare words for which embeddings were not found using externally available embeddings
- Last column shows the **coverage** of our method in percentage

**Table:** This table shows matches for rare words from English (en) and German (de)

**Table:** This table shows matches for rare words from English (en) and German (de)

**Figure:** NLP Task Pipeline

**Figure:** NLP Task Pipeline

**Table:** This table shows matches for rare words from English (en) and German (de)

**Table:** This table shows matches for rare words from English (en) and German (de)

**Table:** Correlation (%) experiments for various string similarity functions used to generate word vectors for German word similarity task (Gur65).

**Table:** Correlation (%) experiments evaluating SO2015 (Soricut and Och, NAACL 2015) versus SWordSS used to generate representations for rare-word similarity task.

**Table:** Test set and RW perplexities (PPL & RW1PPI) for Kneser-Ney 5-gram (KN5), Log-bilinear LM (LBL), LSTM LM, Character-aware neural network LM (Char-LSTM) and LBL initialised with SWordSS embeddings (LBL2SWordSS) in millions, presented on two language datasets.

**Word Relatedness & Similarity Tasks**

**Table:** Word Relatedness & Similarity Tasks

**We applied the following set of pretrained embeddings**

- word2vec-based embeddings trained on Google’s English News Corpus (100 billion tokens)
- Polyglot embeddings trained on a language’s wikipedia dumps (1 million to 1 billion tokens)

**Conclusion**

SWordSS forms a simple and fast method to devise rare-word embeddings, which performs comparably to state-of-the-art on the rare-word similarity task and outperforms more complex Char-LSTM on rare-word perplexity.

26th International Conference on Computational Linguistics, 11-16 December 2016, Osaka, Japan.