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@lsv.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 frequency one (RW1)

Problem learning good word embeddings for such words

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

Language	Train	V	RW	#ENF	Coverage
German	1000K	37K	16K	13K	99.9
Tagalog	585K	22K	11K	8K	98.1
Turkish	239K	25K	14K	10K	99.0
Vietnamese	985K	6K	1K	305	69.1

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 = 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

Embeddings in NLP tasks

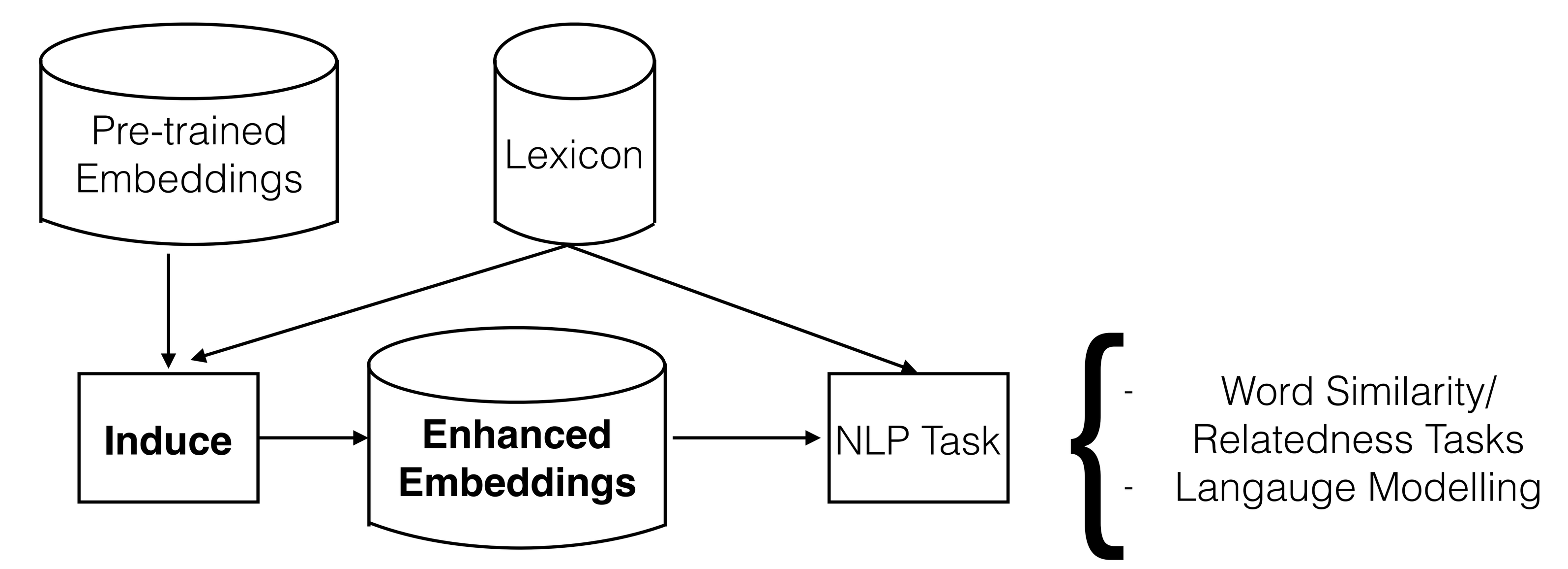


Figure : NLP Task Pipeline

We applied the following set of pre-trained embeddings

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

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: $D_3(language) = \{lan, ang, ngu, gua, uag, age\}$

Index

- Create an inverted index of list words per sub-word unit
- Example: ded = {pad*ded*, *ded*ifferentiation, deci*ded* ... }
- Use a search engine library to do indexing, e.g. Lucene

Search

- Map rare-word (w') to its sub-word units
- Example: $D_3(globalise) = \{glo, lob, oba, bal, ali, lis, ise\}$
- Query the index for most relevant matches of these sub-word units
- Returns a ranked list of words ($R^K(w')$)

globalise (en)	**embroidress** (en)	**Computergraphik** (de)
Globalised	Embroider	computer
globalises	embroider	Graphik
Globalize	embroiderer	Computergrafik

Table : This table shows matches for rare words from english (en) and german (de)

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 (mfk), Tversky Coefficient (tc)
- Default weighting: $S(w, w') = 1$ (labelled with subscript 1)

Sub-**Word** **S**imilarity based **S**earch (**SWordSS**)

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

Word Relatedness & Similarity Tasks

Word Vectors	Gur65
Polyglot	28.5
Polyglot+$SWordSS_{ji}$	37.5
Polyglot+$SWordSS_{jaro}$	37.1
Polyglot+$SWordSS_{jw}$	37
Polyglot+$SWordSS_{mfk}$	37.2
Polyglot+$SWordSS_{ssk}$	36.9
Polyglot+$SWordSS_{tc}$	**37.6**
Polyglot+$SWordSS_{1}$	35.8

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

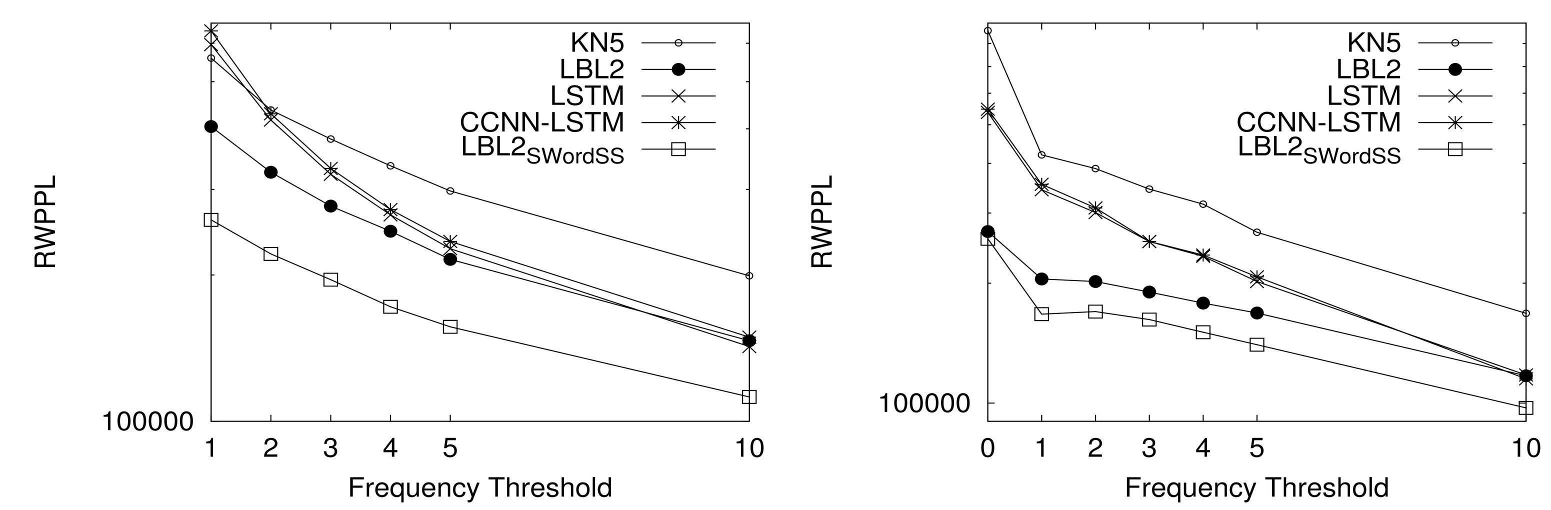
Task	RW Task
Word Vectors	*Google*
SO2015 w/o morph	44.7
SO2015 w/ morph	**52**
w/o SWordSS	45.3
w/ $SWordSS_1$	51.3
w/ $SWordSS_{sim}$	51.4

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

Perplexity Experiments

Language Model	German		Tagalog	
	PPL	RW1PPL	PPL	RW1PPL
KN5	364.2	559K	162.6	420K
LBL2	391.1	404K	171.4	204K
LSTM	323.1	596K	134.7	343K
Char-LSTM	**315.7**	636K	**117.4**	354K
$LBL2_{SWordSS}$	369.4	**260K**	167.2	**167K**

Table : Test set and RW1 perplexities (PPL & RW1PPL) 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 ($LBL2_{SWordSS}$) in millions, presented on two language datasets.



(a) German (b) Tagalog

Figure : Rare-word perplexity versus threshold on frequency of training-set words on German and Tagalog language model corpora.

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