VecShare: A Framework for Sharing Word Representation Vectors

Jared Fernandez, Zhaocheng Yu, Doug Downey
{jared.fern | zhaochengyu2017}@u.northwestern.edu, ddowney@eecs.northwestern.edu
Department of Electrical Engineering and Computer Science, Northwestern University

Objectives

The VecShare framework aims to:

- Centralize web repositories of pre-trained embedding
- Enable programmatic sharing of embeddings
- Reduce time & computational costs of selecting relevant pre-trained embeddings

Visit us at: **www.vecshare.org**

Introduction

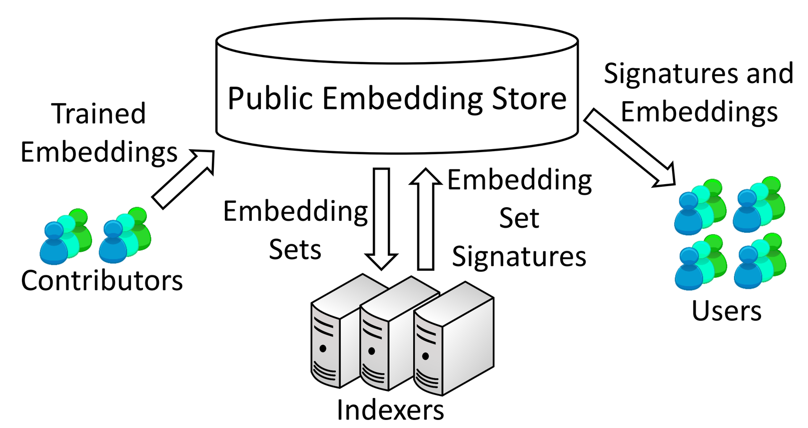

Figure 1: The VecShare Framework

Indexers poll the data share for shared embeddings uploaded by **contributors**. Indexers compute and store compact representations, or **signatures**, of each uploaded embedding. Each **signature** has an associated **similarity measure** which estimates the relevance of the signature's associated embedding to a user's target corpus. **Libraries** integrated with the data share enable programatic access and selection of embeddings.

Signature & Similarity Measures

Signatures rapidly compare embeddings to a target corpora and estimate the relevance of the embedding to a user's task.

- **AvgRank Signature**: T_v most frequent words in the embedding corpus, excluding a set of stop words
- **AvgRank Similarity Method:**
  1. Compute the negative average rank of signature words within the user's frequency-ordered vocabulary.
  2. Most similar embedding is the one with lowest average rank.
- **SimCorr Signature:** Embeddings for T_E most frequent stopwords in the embedding's corpus.
- **SimCorr Similarity Method:**
  1. Compute a set of embeddings from the target corpus
  2. Compute all pairwise similarities between words in both target and embedding corpora
  3. Most similar embedding is one with highest Pearson correlation between shared and corpus embedding pairwise similarities.

Experiments

Experiments were performed to determine the effectiveness of VecShare signatures and similarity measures in selecting accurate embeddings for NLP tasks. Selected embeddings were used as features in a convolutional neural net for text classification.

We evaluate the ability of the previously described signatures to select embeddings from:

- Large-corpus settings: Set of state-of-the-art embeddings trained on billions of tokens
  - Google News, Wikipedia-Gigaword, Twitter, Common Crawl embeddings
- Small-corpus settings: Specific, targeted embeddings trained on corpora of a single topic
  - word2vec embeddings on categorized subsets of the New York Times corpus.

Results

Two measures of the quality of a signature method are reported :

- **ρ**: The Pearson correlation between similarity scores assigned by a method and the set's accuracy on the classification task
- **Acc**: The accuracy of the selected embedding embedding.

Embedding selection was evaluated against: *Random*, and *MaxTkn* baselines. The ensemble method *All* combines the average ranks of the *AvgRank*, *SimCorr*, *MaxTkn* methods.

	Reuters			Subjectivity			IMDB			20news			Average	
	ρ	Sel.	Acc.	ρ	Sel.	Acc.	ρ	Sel.	Acc.	ρ	Sel.	Acc.	ρ	Acc
Random	-	-	0.844	-	-	0.667	-	-	0.829	-	-	0.610	-	0.738
MaxTkn	0.62	govt	0.856	-0.64	govt	0.568	-0.02	govt	0.763	**0.82**	govt	**0.647**	0.20	0.709
VocabRk	0.74	econ	**0.880**	0.51	mov	0.686	0.89	mov	0.835	0.64	book	0.629	**0.70**	0.758
SimCorr	0.51	econ	**0.880**	**0.62**	book	**0.706**	**0.93**	book	0.842	-0.25	agri	0.551	0.45	0.745
All	**0.82**	econ	**0.880**	0.16	book	**0.706**	0.87	book	**0.842**	0.67	book	0.629	0.63	**0.764**
Oracle	-	econ	0.880	-	book	0.706	-	book	0.842	-	govt	0.647	-	0.769

Table 1: Experimental results using small-corpus embeddings. The *VocabRk* and *SimCorr* methods outperform the baselines, and the *All* method performs best in terms of both correlation ρ and text classification accuracy.

	Reuters			Subjectivity			IMDB			20news			Average	
	ρ	Sel.	Acc.	ρ	Sel.	Acc.	ρ	Sel.	Acc.	ρ	Sel.	Acc.	ρ	Acc
Random	-	-	0.862	-	-	0.688	-	-	0.868	-	-	0.763	-	0.795
MaxTkn	**0.63**	web	**0.888**	0.19	web	0.728	0.38	web	0.881	**0.97**	web	**0.863**	**0.54**	0.840
VocabRk	0.46	gnws	0.882	0.02	gnws	**0.759**	0.40	gnws	**0.886**	0.20	gnws	0.719	0.27	0.812
SimCorr	-0.65	wik+	0.84	**0.81**	gnws	**0.759**	0.45	gnws	**0.886**	0.60	twtr	0.748	0.30	0.808
All	0.26	gnws	0.882	0.43	gnws	**0.759**	**0.49**	gnws	**0.886**	0.87	web	**0.863**	0.51	**0.848**
Oracle	-	web	0.888	-	gnws	0.759	-	gnws	0.886	-	web	0.863	-	0.85

Table 2: Experimental results using large-corpus embeddings. All of the signature methods outperform the random baseline, and the *All* method performs best in terms of both correlation ρ and text classification accuracy.

Efficiency Experiements

Average Time for Embedding Selection:

- Conventional Approach: 177 minutes
- *VocabRk* Signature: 38 seconds

The VecShare *AvgRank* signature method provided an average 280x speedup over current practice of manually training and evaluating each embedding. VecShare reduces memory overhead: the total size of signatures is 4-5 orders of magnitude smaller than full embedding sets.

Library

The VecShare library for Python 2.7/3.5 is available by `pip install vecshare` on PyPi, with support for:

- Word Vector Query and Extraction
- Embedding Selection: *AvgRank*, *MaxTkn*, & *WordSim*
- Embedding Upload and Download

```
>>> from vecshare import vecshare as vs
>>> vs.check()
        embedding_name   case_sensitive   dimension   emb_typ  vocab_size
0            reutersr8            False         100  word2vec        7821
1        reuters21578            False         100  word2vec       20203
2                brown            False         100  word2vec       15062
3    glove_gigaword100d           False         100  word2vec      399922
4         oanc_written            False         100  word2vec      732127

>>> vs.query(['The', 'farm'], 'agriculture_40')
   text       d99       d98       d97       d96  ...        d1        d0
0   the -1.414755  0.414973  1.115698  0.03408   ...  0.037287 -1.004704
1  farm  0.349535 -0.379208 -0.189476  2.776809  ...  0.067443 -1.391604
[2 rows x 101 columns]

>>> vs.extract('agriculture_40', 'Test_Input/reutersR8_all')
Embedding extraction begins.
100% (23584 of 23584) |################################| Elapsed Time: 0:01:04
Embedding successfully extracted.
```

Figure 2: VecShare API Example

The library is extensible and allows for the addition of both new indexers and signatures at any time.

Acknowledgements

Research supported in part by NSF grant IIS-1351029 & the Allen Institute for Artificial Intelligence. Travel supported in part by Northwestern University EECS Department and Undergraduate Research Grant Program.