Integrating Lexical Knowledge in Word Embeddings using Sprinkling and Retrofitting

Aakash Srinivasan*[1], Harshavardhan Kamarthi*[2], Devi Ganesan[2], and Sutanu Chakraborti[2]

[1]Dept. of Computer Science, University of California, Los Angeles
[2]Dept. of Computer Science and Engineering, Indian Institute of Technology Madras

Introduction

- Improving the quality of word embeddings has led to better performance in many downstream language tasks.

Word Embeddings
+
Lexical Knowledge Sources
(PPDB, WordNet, etc)
= Semantic Word Embeddings

- We introduce two novel approaches for incorporating semantic knowledge into word embeddings : **Sprinkling and Weighted Retrofitting**.

Sprinkling

- The objective function used in Word2Vec[6] implicitly factorizes a Shifted PPMI (SPPMI) matrix [4].
- We sprinkle lexical knowledge into the SPPMI matrix prior to factorization.
- Sprinkling was first used in the context of Latent Semantic Analysis [1], where class labels were sprinkled into the corresponding training documents (i.e. class labels were appended as extra features).

$$\textit{SPPMI} = max(PMI - \log(neg), 0) \quad (1)$$

$$\textit{SS-PPMI} = \textit{SPPMI} \circ L_k \quad (2)$$

$$\textit{SS-PPMI} \approx U_x \Sigma_x V_x^T \quad (3)$$

$$\textit{Embeddings} = U_x \Sigma_x^p \quad (4)$$

- Reachability matrix captures the presence/absence of syntactic relations such as synonymy or antonymy between words.
- Words that have similar neighbourhood in lexical graphs will have similar columns in the reachability matrix.
- Appending reachability matrix to SPPMI matrix would bring the embeddings of such words closer.

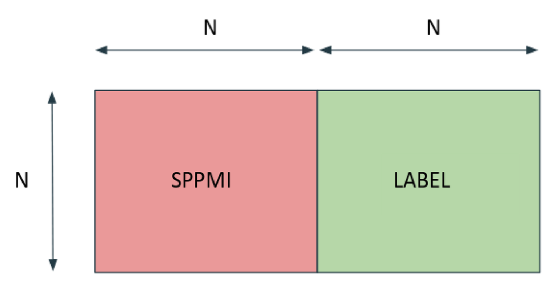

Fig. 1: SS-PPMI

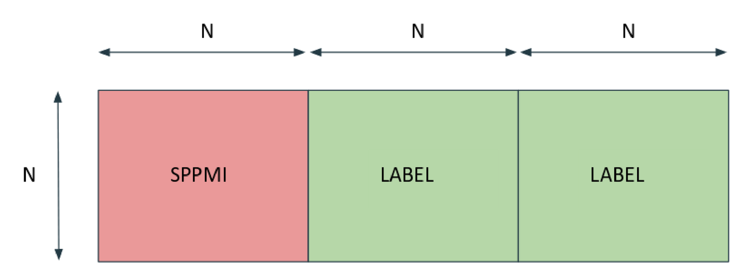

Fig. 2: DSS-PPMI

- When the reachability matrix is appended twice, we call the resulting matrix as *Doubly Sprinkled Shifted - Positive PMI* (**DSS-PPMI**).

Weighted Retrofitting

- Retrofitting [2] is a post-processing technique for injecting semantic knowledge into word embeddings.
- Given the pre-trained vectors $\hat{Q} = (\hat{q}_1, \hat{q}_2 \cdots \hat{q}_n)$, and a knowledge base represented by the adjacency matrix A, we need to learn new vectors $Q = (q_1, q_2 \cdots q_n)$ such that following objective $\psi(Q)$ is minimized:

$$\psi(Q) = \sum_{i=1}^{i=n} (\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{j=1}^{j=n} \beta_{ij} A_{ij} \|q_i - q_j\|^2) \quad (5)$$

- The β_{ij} term is usually assigned as $degree(i)^{-1}$.
- We use WordNet-based similarity scores $Sim(i,j)$ to obtain a better setting of β_{ij}.

$$\beta_{ij} = \frac{Sim(i,j)}{\sum_{j'} Sim(i,j')} \quad (6)$$

- The similarity score $Sim(i,j)$ is the maximum of the similarity scores of all possible pairs of synsets, taking one each from the two words.

Sources of Knowledge

- **WordNet**: 82313 nodes, 98678 edges; **PPDB**: 84467 nodes, 169703 edges.
- SPPMI matrix generated from 6 Billion Wikipedia articles.
- Lexical relations: **synonymy, hypernymy, meronymy** and **verb entailment**.

Intrinsic evaluation: Word Similarity and Analogy

Method	Graph	Hops	WS353	SimLex999	RW	SemEval
SPPMI	-	-	0.663	0.385	0.516	0.176
GloVe	-	-	0.601	0.37	0.41	0.164
SynGCN	-	-	0.601	**0.455**	0.337	**0.234**
SS-PPMI	PPDB	1	0.663	0.386	0.516	0.175
		2	0.669	0.398	0.521	**0.180**
DSS-PPMI		1	0.663	0.386	0.516	0.176
		2	0.668	0.420	**0.528**	0.188
SS-PPMI	WordNet	1	0.667	0.393	0.464	0.166
		2	0.671	0.394	0.435	0.165
DSS-PPMI		1	0.667	0.393	0.463	0.166
		2	**0.677**	0.394	0.414	0.161
Retrofit(path)	WordNet	1	0.631	0.496	0.431	0.171
W-Retrofit(path)		1	0.641	**0.509**	0.417	0.167
		2	0.562	0.422	0.372	0.151
Retrofit(jcn)	PPDB	2	0.607	0.434	0.387	**0.184**
W-Retrofit(jcn)		1	0.6	0.432	0.353	0.161
		2	0.616	0.399	0.389	0.155

1. Double Sprinkling (*DSS-PPMI*) works better than *SPPMI* on word similarity task.
2. Increasing the number of hops (k) in the reachability matrix improves the performance on word similarity task.
3. For W-retrofitting, the use of PPDB as knowledge source and path based similarity as weights gives the best performance and outperforms the baselines in most benchmarks.

Extrinsic Evaluation: NER and PoS Tagging

For Named Entity Recognition, we use the neural network architecture proposed in [3] on CoNLL-2003 dataset [8]. For Part of Speech Tagging, we use the LSTM based neural architecture discussed in [7] on the Penn treebank dataset [5].

Method	Graph	Hops	NER	PoST
SPPMI	-	-	82.3	92.9
GloVe	-	-	89.1	94.6
SynGCN	-	-	**89.5**	**95.4**
SS-PPMI	PPDB	1	83.4	93.3
		2	84.7	93.4
DSS-PPMI		1	82.3	93.5
		2	**87.3**	**93.4**
SS-PPMI	WordNet	1	83.5	92.8
		2	83.9	93.2
DSS-PPMI		1	83.2	93.2
		2	83.5	93.1
Retrofit(path)	WordNet	1	88.8	94.8
W-Retrofit(path)		1	88.7	95
		2	89.2	95.1
Retrofit(jcn)	PPDB	2	88.2	94.5
W-Retrofit(jcn)		1	88.9	95
		2	**89.4**	**95.3**

1. A clear increase in scores is observed in the both extrinsic tasks on using the proposed SS-PPMI matrix over only the SPPMI matrix.
2. W-Retrofitting model using jcn weights on wordnet graph are very similar to SynGCN model inspite of SynGCN being a more complex model with a lot of hyperparameters. Other methods of W-retrofitting have comparable performance to SynGCN.
3. In general, we see improved performance by considering upto 2 hop neighbours.

References

[1] Sutanu Chakraborti et al. “Sprinkling: supervised latent semantic indexing”. In: *European Conference on Information Retrieval.* Springer. 2006, pp. 510–514.

[2] Manaal Faruqui et al. “Retrofitting word vectors to semantic lexicons”. In: *arXiv preprint arXiv:1411.4166* (2014).

[3] Kenton Lee, Luheng He, and Luke S. Zettlemoyer. “Higher-order Coreference Resolution with Coarse-to-fine Inference”. In: *NAACL-HLT.* 2018.

[4] Omer Levy and Yoav Goldberg. “Neural word embedding as implicit matrix factorization”. In: *Advances in neural information processing systems.* 2014, pp. 2177–2185.

[5] Mitchell P. Marcus et al. “The Penn Treebank: Annotating Predicate Argument Structure”. In: *HLT.* 1994.

[6] Tomas Mikolov et al. “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781* (2013).

[7] Nils Reimers and Iryna Gurevych. “Reporting Score Distributions Makes a Difference: Performance Study of