Supplementary to: Incorporating Syntactic and Semantic Information in Word Embeddings using Graph Convolutional Networks

Shikhar Vashishth[1] **Manik Bhandari**[2*] **Prateek Yadav**[3*]
Piyush Rai[4] **Chiranjib Bhattacharyya**[1] **Partha Talukdar**[1]
[1]Indian Institute of Science, [2]Carnegie Mellon University
[4]Microsoft Research, [4]IIT Kanpur
{shikhar,chiru,ppt}@iisc.ac.in, mbhandar@andrew.cmu.edu
t-pryad@microsoft.com, piyush@cse.iitk.ac.in

1 Hyperparameters

Our vocabulary consists of 150k most frequent words in Wikipedia corpus. Following (Mikolov et al., 2013a; Pennington et al., 2014), we have reported results for 300-dimensional embeddings on intrinsic tasks. However, for extrinsic tasks results are reported for 256-dimensional embeddings since pre-trained ELMo model is available for only $\{128, 256, 512, 1024\}$ sizes. For training baselines, we use the code provided by the authors with the default hyperparameters. For training SynGCN and SemGCN, we use Adam optimizer (Kingma and Ba, 2014) with learning rate of 0.001. Following (Mikolov et al., 2013b), subsampling is used with threshold parameter $t = 10^{-4}$. The target and neighborhood embeddings are initialized randomly using Xavier initialization (Glorot and Bengio, 2010). In GCN, number of layers (k) is taken as 1 and ReLU is used as the activation function.

2 Evaluating performance with same semantic information

In this section, we present the complete set of results for comparison of SemGCN against other methods when provided with the same semantic information (synonyms from PPDB). Similar to Section 9.3, in Table 1 and 2 (**Please look at table on the next page**), we present the comparison on intrinsic tasks and extrinsic tasks. Please refer Section 9.4 for more details.

Method	POS	SQuAD	NER	Coref
SynGCN	95.4±0.1	79.6±0.2	89.5±0.1	65.8±0.1
Retro-fit (X,1)	94.8±0.1	79.6±0.1	88.8±0.1	66.0±0.2
Counter-fit (X,1)	94.7±0.2	79.9±0.1	88.2±0.3	65.5±0.1
JointReps (X,1)	95.4±0.2	79.4±0.3	89.1±0.3	65.6±0.0
SemGCN (X,1)	**95.5±0.1**	**80.4±0.2**	**89.7±0.2**	**66.1±0.2**

Table 1: Comparison of different methods when provided with same semantic information (synonym) for fine tuning SynGCN embeddings. Please refer Section 9.4 of paper for details.

References

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[*] Contributed equally to the work.

Init Embeddings (=X)	Word2vec			GloVe			Deps			EXT			SynGCN		
Dataset	WS-S	AP	MSR	WS-S	AP	MSR	WS-S	AP	MSR	WS-S	AP	MSR	WS-S	AP	MSR
Performance of X	71.4	63.2	44.0	69.2	58.0	45.8	65.7	61.8	40.3	**69.6**	52.6	18.8	73.2	69.3	52.8
Retro-fit (X,1)	72.3	**67.1**	**46.8**	72.6	58.7	47.2	65.2	**62.3**	41.0	69.1	54.2	40.5	75.3	67.1	51.4
Counter-fit (X,1)	69.0	63.3	31.5	68.3	56.6	29.6	57.5	56.3	32.0	55.6	53.5	35.8	71.4	62.5	31.7
JointReps (X,1)	69.7	56.9	28.7	70.5	52.7	37.5	61.8	58.7	36.8	70.1	54.2	21.1	76.4	61.8	28.2
SemGCN (X,1)	**74.3**	64.0	34.2	**78.3**	**59.1**	**51.2**	**68.5**	61.9	**44.4**	69.5	**56.0**	**50.0**	$\boxed{\mathbf{79.0}}$	$\boxed{\mathbf{70.0}}$	$\boxed{\mathbf{55.0}}$

Table 2: Evaluation of different methods for incorporating same semantic information (synonym) initialized using various pre-trained embeddings (X). M(X, R) denotes the fine-tuned embeddings using method M taking X as initialization embeddings. R denotes the number of semantic relations used as defined in Section 9.3. SemGCN outperforms other methods in 11 our of 15 settings. SemGCN with SynGCN gives the best performance across all tasks (highlighted using $\boxed{\cdot}$). Please refer Section 9.4 for details.