Learning Word Vectors Efficiently Using Shared Representations and Document Representations

Qun Luo and Weiran Xu

Beijing University of Posts and Telecommunications, 10 Xitucheng road, Beijing, China.
86-15210860278 and luoqun9191@163.com, 86-13041219475 and xuweiran@bupt.edu.cn

Abstract

We propose some better word embedding models based on vLBL model and ivLBL model by sharing representations between context and target words and using document representations. Our proposed models are much simpler which have almost half less parameters than the state-of-the-art methods. We achieve better results on word analogy task than the best ones reported before using significantly less training data and computing time.

Introduction

Word representations learned by neural network language models can be used for many NLP tasks such as POS tagging, chunking, named entity recognition, semantic and syntactic similarity (Collobert et al. 2011; Mikolov et al. 2013a). The first neural net language model is proposed in 2003(Bengio et al. 2003), which consists of input, projection, hidden and output layers. Mikolov removed the nonlinear hidden layers and proposed two state-of-the-art models (CBOW and Skip-gram) to capture better semantic and syntactic word similarity. Mnih adds weight vectors to the input context and captures comparable results on semantic and syntactic word similarity with less training data and computing time. In the models(vLBL and ivLBL) proposed in (Mnih and Kavukcuoglu 2013), each word has a context representation for input and a target representation for output.

There are two main problems existing in vLBL and ivLBL. The first main problem is that context representation and target representation are reduplicative. The second problem is that local context windows based models are shallow. Even the GloVe model(Pennington et al. 2014) is shallow because the co-occurrence matrix is based on context window. To solve the first problem, we propose CBOW-LBL model and Skip-LBL model by sharing context representation and target representation, namely, context representation is same with target representation in our models. To solve the shallow problem of window-based methods, we propose D-CBOW model by leveraging document representations. To assess our models, we use semantic and syntactic questions released by google.

Methods

CBOW-LBL

The CBOW-LBL uses context words to predict the target words. We denote context representation and target representation for word w as r_w. Given a sequence of context words $h = w_0, w_1, \ldots, w_{n-1}, w_{n+1}, w_{n+2}, w_{2n}$, the model predicts the target word w_n by taking a linear combination of context words h:

$$\hat{r}(h) = \sum_{i=0, i\neq n}^{2n} c_i \odot r_{w_i} \quad (1)$$

where c_i is the weight vector for the context word w_i, n is the length of left window and right window, $\odot$ denotes element-wise multiplication. We use score function to compute the similarity between predicted representation $\hat{r}(h)$ and real representation r_{w_n}:

$$s_\theta(w_n, h) = \hat{r}(\mathrm{h})^T r_{w_n} \quad (2)$$

CBOW-LBL is similar to the vLBL where representation between context and target is shared and bias in score function is removed. By sharing the representation between context representation and target representation, the number of parameters is half less than the number of vLBL model.

Skip-LBL

Different from the CBOW-LBL model, Skip-LBL model uses the target words to predict the context words surrounding them. Given the target word w_n, the model predicts the context words h(h is same with the CBOW-LBL) surrounding w_n. Take w_i as an example:

$$\hat{r}(w_i) = c_i \odot r_{w_n} \quad (3)$$

The same score function then computes the similarity between the predicted representation $\hat{r}(w_i)$ and real representation r_{w_i}:

$$s_{i,\theta}(w_i, w_n) = \hat{r}(w_i)^T r_{w_i} \quad (4)$$

D-CBOW

Different from window-based methods, D-CBOW use context words and document information to predict target words.

$$\hat{r}(h,D) = \sum_{i=0,i\neq n}^{2n} c_i \odot r_{w_i} + c_d \odot r_D \quad (5)$$

where D is the document that contains h, c_d is the weight vector for all documents, r_D is the representation of the document D.

We use the same score function with CBOW-LBL and Skip-LBL to compute the similarity between predicted representation $\hat{r}(h,D)$ and real representation r_{w_n}:

$$s_\theta(w_n, h\ D) = \hat{r}(h, D)^T r_{w_n} \quad (6)$$

Experimental results

We evaluated our word representations on word pairs similarity. The dataset released by google is applied in our similarity task, and there are 8869 semantic questions of five types and 10675 syntactic questions of nine types (https://code.google.com/p/word2vec/). We use a subset from Wikipedia2010 as our training data which consists of 990 million training words (http://nlp.stanford.edu/data/WestburyLab.wikicorp.201004.txt.bz2). We also choose the first 413 million words as training dataset for CBOW-LBL and Skip-LBL.

All models are trained for only one epoch with 10 noise samples. The window length for all models is 5. We chose starting learning rate 0.025 and decreased it linearly.

We compare our results on word pair similarity with other best ones reported in (Mikolov et al. 2013a; Mnih and Kavukcuoglu 2013; Pennington et al. 2014). The results are shown in Table 1. The results in Table 1 show that our models substantially outperform other best models reported before using smaller amount of training data and computing time. For example, D-CBOW with 300 dimensions trained for just 170 minutes on a small dataset, achieves better accuracy scores than other models trained on larger dataset.

Table 1: Overall accuracy in percent on word pairs similarity task. The Skip-gram and CBOW results are from (Mikolov et al. 2013a), The vLBL and ivLBL results are from (Mnih and Kavukcuoglu 2013), GloVe result is from (Pennington et al. 2014).

Model	EMBED. Dim.	Training words	Overall accuracy	Time days
Skip-gram	300	783M	49.2	1
Skip-gram	300	1.6B	53.8	2
Skip-gram	1000	6B	65.6	
CBOW	300	1.6B	36.1	0.6
CBOW	1000	6B	63.7	
vLBL	300	1.5B	60.0	2
vLBL	600	1.5B	62.1	2
ivLBL	300	1.5B	62.6	4.1
ivLBL	600	1.5B	64.0	4.1
GloVe	300	1.6B	70.3	
GloVe	300	42B	75.0	
CBOW-LBL	200	413M	65.6	0.03
CBOW-LBL	300	413M	67.8	0.04
CBOW-LBL	300	990M	73.8	0.11
Skip-LBL	200	413M	60.2	0.13
Skip-LBL	300	413M	63.2	0.16
D-CBOW	300	990M	**75.5**	0.12

Discussion

The results show that as our models are much simpler with less parameters, our models can achieve better results on word pairs similarity task than other reported best ones using less training data and computing time. D-CBOW outperforms other models because D-CBOW leverages document information. How to use global training data information is also worth investigating as it might result in much better word representations.

Acknowledgement

This work was supported by 111 Project of China under Gr ant No. B08004, key project of ministry of science and tec hnology of China under Grant No. 2011ZX03002-005-01, National Natural Science Foundation of China (612732 17) and the Ph.D. Programs Foundation of Ministry of Edu cation of China (20130005110004).

References

Bengio Y, Ducharme R, Vincent P, et al. A Neural Probabilistic Language Model[J]. Journal of Machine Learning Research, 2003, 3: 1137-1155.

Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch[J]. The Journal of Machine Learning Research, 2011, 12: 2493-2537.

Jeffrey Pennington, R S, Manning C D. GloVe: Global Vectors for Word Representation[J]. EMNLP 2014.

Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013a.

Mnih A, Kavukcuoglu K. Learning word embeddings efficiently with noise-contrastive estimation[C]//Advances in Neural Information Processing Systems. 2013: 2265-2273.