ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-99

Cluster-Driven Model for Improved Word and Text Embedding

Zhe Zhao and **Tao Liu** and **Bofang Li** and **Xiaoyong Du**[1, 2]

Abstract. Most of the existing word embedding models only consider the relationships between words and their local contexts (e.g. ten words around the target word). However, information beyond local contexts (global contexts), which reflect the rich semantic meanings of words, are usually ignored. In this paper, we present a general framework for utilizing global information to learn word and text representations. Our models can be easily integrated into existing local word embedding models, and thus introduces global information of varying degrees according to different downstream tasks. Moreover, we view our models in the co-occurrence matrix perspective, based on which a novel weighted term-document matrix is factorized to generate text representations. We conduct a range of experiments to evaluate word and text representations learned by our models. Experimental results show that our models outperform or compete with state-of-the-art models. Source code of the paper is available at https://github.com/zhezhaoa/cluster-driven.

1 Introduction

Word embedding models (also known as neural language models) encode syntactic and semantic information of words into low-dimensional real vectors, where words share similar meanings tend to have similar representations. Generating word embedding is one of the most fundamental tasks in the NLP literature. Word embeddings are widely used in tasks such as tagging and text classification, and have been reported to bring significant improvements on those tasks. Most word embedding algorithms are trained by modeling the relationships between target words and their local contexts, which is based on the distributional hypothesis of Harris: ***words in similar contexts have similar meanings***. However, global contexts, which usually reflect semantic meanings of target words, are generally ignored by these models. For example, words that often co-occur in the same texts tend to reflect similar topics or sentiment tendencies, even they seldom appear in each others' local contexts.

As far as we know, there is still rare research which utilizes global context for word embedding training besides the following three works. Huang et al. [8] propose GCANLM on the basis of C&W [2], where authors use weighted average of word embeddings to represent texts (global contexts), and the embeddings of words and their corresponding texts are trained to obtain higher scores. However, C&W and GCANLM are slow in computation and are reported to perform relatively poorly on various linguist tasks compared to state-of-the-art methods, such as models in the word2vec toolkit [3] [19, 18].

Paragraph Vector(PV), proposed by Le and Mikolov [12], introduces global information into word2vec. PV embeds texts by predicting the words they include, and thus introduces global information into word embedding indirectly, though the aim of PV is training text embedding. Sun et al. [22] demonstrate the superiority of PV (or the variants of PV) on various word-level linguistic tasks. The problem of PV is that, it has to give every text a vector. For large-scale datasets such as Wikipedia, the number of texts is much larger than the vocabulary size, which requires expensive computational resources during the training process. Besides that, none of GCANLM and PV introduce global information of different degrees according to different applications. Intuitively, for tasks like word syntactic analogy, more local information is preferred, while tasks such as sentiment analysis tend to favor global information, where rich semantics are included.

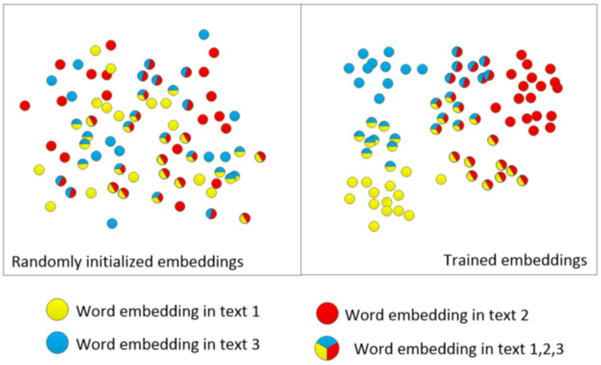

Figure 1. Visualization of forming clusters in two dimensional case.

In this paper, a more general and powerful framework of utilizing global context is proposed for learning improved word and text embedding, namely, the cluster-driven models. The main idea of our models follow the concept of clustering algorithms. The models are trained to make the embedding of words in the same text to form a cluster (as shown in figure 1). Different from other clustering algorithms, in our models, which word belongs to which cluster is preordained and a word may belong to multiple clusters (a word occurs in different texts). Nevertheless, the objectives of our models are the same with other clustering methods: intra-cluster distances are minimized while inter-cluster distances are maximized. As a result, our models extend words' contexts from local windows to the whole texts. Though our model is not based on neural networks, we still call it embedding model since it is trained in an on-line, stochastic fashion. The cluster-driven models can be used standalone, which are able to capture rich semantic information, and can also be inte-

[1] Key laboratory of Data Engineering and Knowledge Engineering, MOE, email: [helloworld][tliu][libofang][duyong]@ruc.edu.cn
[2] School of Information, Renmin University of China
[3] https://code.google.com/p/word2vec/

grated into existing word embedding models easily, and hence the degree of utilizing global information can be adapted to the requirements of different applications.

From word embedding to text embedding, considerable attention has been paid to designing various Neural Networks (NNs) to learn complex compositionality of texts, such as word order, sentence structure and even document structure [7]. Word order is taken into consideration in convolutional NNs (CNNs) [10] and recurrent NNs (RNNs) [17, 3]; Recursive NNs (RecNNs) make fully use of syntactic information by constructing neural networks on the basis of parse tree [21]; Recently several works have been proposed to use combination of NNs to model the documents hierarchically [11, 23]. For example, Li [15] uses recursive NN to learn sentence embeddings from word emebddings and use recurrent NN to learn document embeddings from sentence embeddings. Though complex compositionality are learned upon word embedding, these models still don't show significant superiority over bag-of-words models [9]. In this paper, we discover that with richer semantic word embeddings, superior text embeddings, at least for sentiment analysis, can be obtained even by simple strategy such as word embeddings averaging (VecAvg). This paper also provides us better understanding of Paragraph Vector (PV), a very popular method for learning text embeddings. We discover that the superiority of PV comes from the use of global information, rather than the way it trains text embeddings (it trains in prediction manner).

For a thorough comprehension of the cluster-driven models, we analyze it in the co-occurrence matrix perspective. Count-based and embedding methods are two families for generating low dimensional word and text representations. Count-based methods directly utilize co-occurrence statistics and usually obtain dense representations by factorizing co-occurrence matrix [4]. Count-based methods usually served as poor baselines in various linguistic tasks until the works done by Levy and Goldberg [13] and Pennington et al. [20], which demonstrate that count-based models can compete with state-of-the-art word embedding models [14]. In this paper, we extend their works from word representations to text representations. We present count-based counterparts of our cluster-driven models, based on which we factorize a novel weighted matrix of term-document type. Experimental results show this new count-based model can achieve comparable results with state-of-the-art text embedding models, and even outperforms previous approaches on small-scale datasets.

2 Models

2.1 Embedding Models Revisit: Train in Local Manner

Word embedding models can capture the syntactic and semantic information of words from large-scale unlabeled corpus. In contrast to traditional bag-of-words representations, relationship between word embeddings mirrors the syntactic and semantic similarities between two words. For example, words that share similar meanings are close to each other, e.g. 'strong' and 'powerful'. And embedding models can also preserve some interesting linear translation patterns, e.g. Vec('Madrid') - Vec('Spain') + Vec('France') = Vec('Paris').

Most word embedding algorithms are trained by maximizing the log-likelihood of the probability of the target word given its local context [1]:

$$L(\theta_1,\theta_2) = \sum_{i=1}^{|WN|} logP(w_i|w_i^{context}) \quad (1)$$

where $w_i^{context}$ denotes the local context of word w_i. $|WN|$ is the number of training words in the whole dataset. Word embeddings and parameters in neural network are respectively denoted by θ_1 and θ_1. Different word embedding models differ in how they define the conditional probability and how they represent the local contexts.

2.2 Cluster-Driven Models: Train in Global Manner

One obvious drawback of the existing models is that they don't use information beyond local contexts. For capturing global information, two versions of the cluster-driven models are designed: pairwise model and centric model, both of which are trained by making embeddings of words in the same text to form a cluster.

2.2.1 Pairwise Cluster-Driven (PCD)

In pairwise model, word pairs are sampled for distance adjustment according to whether they are in the same text or not. The objective function of the model consists of two components.

Minimizing intra-cluster distances The first component of the objective function is to decrease the distance between the embedding of words in the same texts. A certain number of word pairs are sampled and distances between them are minimized as the following objective:

$$G_1^P(\theta_1) = \sum_{i=1}^{|T|}\sum_{j=1}^{|t_i|}\sum_{k=1}^{|POS|} E_{w_k \sim PT_i(w)} intra\_dis(e_{w_{ij}}, e_{w_k}) \quad (2)$$

where *intra_dis* is used to measure the distance between two words in the same text. It penalizes the case where embeddings of two words in the same text are far away from each other. $t_i=\{w_{i1},w_{i2},......,w_{i|t_i|}\}$ denotes i_{th} text and $T=\{t_1,t_2,......,t_{|T|}\}$ denotes the whole dataset. e_w denotes the embedding of word *w*. For each word w_{ij}, $|POS|$ words in the same text are sampled from the distribution *$PT_i(w)$* and distances between them are minimized. Intuitively, it is better to shorten the distance between two related words, like 'amazing' and 'amazingly', instead of 'amazing' and 'the'. It is also favorable that the probabilities of words pairs being sampled decline as their distance increases, since very distant word pairs tend to share less relevant information. However, in this paper, *$PT_i(w)$* is just the uni-gram distribution of the i_{th} text. We find this simple strategy works pretty well if the number of word pairs sampled is large enough.

Maximizing inter-cluster distances The second component of the objective function is to increase the distance between embeddings of words in different texts. Word pairs are sampled from the whole dataset and the following objective function is maximized:

$$G_2^P(\theta_1) = \sum_{i=1}^{|T|}\sum_{j=1}^{|t_i|}\sum_{k=1}^{|NEG|} E_{w_k \sim P_n(w)} inter\_dis(e_{w_{ij}}, e_{w_k}) \quad (3)$$

where *inter_dis* is used to measure the distance between two words in different texts. It penalizes the case where the embedding of words in different texts are close to each other. For each training word, $|NEG|$ words are drawn from distribution *$P_n(w)$*, a uni-gram distribution raised to the *n-th* power [19].

The final objective function of the model is as follows:

$$G(\theta_1) = G_1^P(\theta_1) - G_2^P(\theta_2) \tag{4}$$

The size of the global contexts is much larger than local contexts. Intuitively, models require much more training time to exploit global contexts. However, in this model, global information can be utilized efficiently and effectively through sampling.

2.2.2 Centric Cluster-Driven (CCD)

In centric model, centroid vector which has the same dimension with word embedding is introduced to denote the center of each cluster. Instead of adjusting distances between the embedding of words directly, we adjust distances between centroid vectors and word embeddings. Like the pairwise case, the objective of the centric model also consists of two components. The first component is to minimize the distances between the centroid vector and the embedding of words in the corresponding text:

$$G_1^C(\theta_1, ct) = \sum_{i=1}^{|T|}\sum_{j=1}^{|t_i|} intra\_dis(e_{w_{ij}}, ct^i) \tag{5}$$

where ct^i denotes the centroid vector of the text t_i. By introducing centroid vectors, we indirectly decrease the distances between all word pairs in the text.

The second component of the objective is to maximize the distances between the centroid vector and the embedding of words in different texts:

$$G_2^C(\theta_1, ct) = \sum_{i=1}^{|T|}\sum_{k=1}^{|NEG|*|t_i|} E_{w_k \sim P_n(w)} inter\_dis(e_{w_k}, ct^i) \tag{6}$$

$|NEG|$ words are drawn from distribution $P_n(w)$ for each training word. Namely, $|NEG| * |t_i|$ words are drawn for text t_i and distances between the centroid vector and the embedding of these sampled words are maximized.

Empirically, the centric model requires less training time to achieve comparable results with the pairwise model. However, the centric model requires much more memory since each text has a unique centroid vector. Besides that, The centric model performs relatively poorly in sentence-level texts. We speculate the reason is that too much noise is introduced when utilizing a vector to represent only a few words.

2.2.3 Distance Measures

Distance/Similarity measures reflect the degree of closeness or separation of two embeddings. They are important for the performance of models. In this paper, different distance measures are used according to whether two words are in the same text or not. Table 1 lists two sets of distance measures. To make sure that global objectives are in the same numeric range with local objectives, we add sigmoid function on distance measures since most objectives of the local embedding models are trained by maximizing the conditional probabilities of target words. When we use the second set of distance measures, the centric model is similar to PV-DBOW, a variant of PV [12]. PV-DBOW can be viewed as a special case of cluster-driven models when only negative sampling is used as softmax.

Table 1. Different sets of distance measures.

	Intra-cluster	Inter-cluster
Measures1	$(\sigma(e_1{}^T e_2) - 1)^2$	$-(\sigma(e_1{}^T e_2) - 0)^2$
Measures2	$log(\frac{1}{\sigma(e_1{}^T e_2)})$	$log(\sigma(-e_1{}^T e_2))$

2.3 Integrated Model

The cluster-driven models can be integrated into existing word embedding models by linearly combining the local and global objective functions:

$$(1-\lambda)(-L(\theta_1, \theta_2)) + \lambda G(\theta_1) \tag{7}$$

By adjusting λ, we can easily balance the local and global information during the training process. When λ equals to zero, the model is the same with existing local embedding models. More global information is introduced into the model as λ increases. When λ equals to one, only global information is utilized to train word embeddings. In section 5.2, we will demonstrate that the embedding trained in local manner tends to capture syntactic information while the embedding trained in global manner tends to capture semantic information. As a result, we can train word embeddings of different properties according to where embeddings are used. Figure 2 shows the framework of the integrated model.

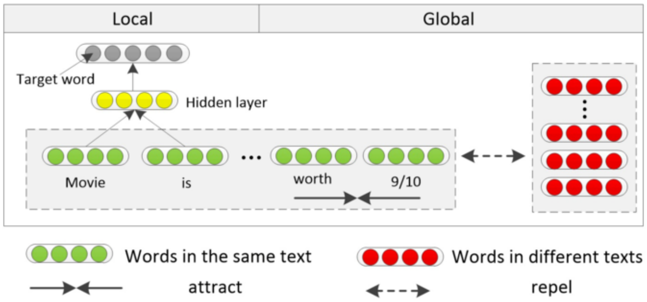

Figure 2. Illustration of the integrated model.

2.4 From Word Embedding to Text Embedding

PV and VecAvg are two approaches for learning text embedding from word embedding in an unsupervised framework [12]. Here, we use PV to refer to the process of learning text embedding by predicting the words it includes. Assumption behind PV is that a good text embedding should be able to predict the words it includes in larger probabilities, while assumption behind VecAvg is that a good text embedding should be similar with the words it includes. They are both essentially bag-of-words models and enjoy the advantages of being efficient and robust. In this paper, we discover that with rich semantic word embedding, neural bag-of-words models like PV and VecAvg can still rival the models that learn complex compositionality upon word embeddings.

3 Theoretical Analysis

To better understand the cluster-driven models, we further explore them in co-occurrence matrices perspective though they do not require to construct co-occurrence matrices at all. Following the theoretical analysis, we factorize the shifted positive pointwise mutual information matrix (SPPMI) of term-document type via singular value decomposition (SVD) to obtain text representations. Count-based models usually serve as poor baselines or are even seldom taken into consideration for generating text representation [16]. However, we discover that when suitable weighted co-occurrence matrix is factorized, count-based models can still achieve comparable results with state-of-the-art models.

3.1 Co-occurrence Matrix Perspective for Cluster-Driven Model

Take CCD for example, we begin by rewriting its objective:

$$\begin{aligned}&\sum_{i=1}^{|T|}\sum_{j=1}^{|t_i|} intra\_dis(e_{w_{ij}}, ct^i)\\ &-\sum_{i=1}^{|T|}\sum_{k=1}^{|NEG|*|t_i|} E_{w_k \sim P_n(w)} inter\_dis(e_{w_k}, c^i)\end{aligned} \tag{8}$$

For specific term-document pair *(w_a,t_b)*, the objective is:

$$\begin{aligned}&c(w_a, t_b) * intra\_dis(e_{w_a}, ct^b)\\ &-|t_b| * |NEG| * P_n(w_a) * inter\_dis(e_{w_a}, ct^b)\end{aligned} \tag{9}$$

where *c(w_a,t_b)* is the number of times word w_a appears in text t_b . From equation 9, we can see more clearly that our model utilizes no more information than term-document co-occurrence matrices and some other basic statistics, such as the length of texts and words distribution. Next, we rewrite the equation by replacing *intra_dis* and *inter_dis* with concrete distance measures:

$$\begin{aligned}&c(w_a, t_b) * (\sigma({e_{w_a}}^T ct^b) - 1)^2\\ &-|t_b| * |NEG| * P_n(w_a) * (-(\sigma({e_{w_a}}^T ct^b) - 0)^2)\end{aligned} \tag{10}$$

$$\begin{aligned}&c(w_a, t_b) * log(1/\sigma({e_{w_a}}^T ct^b))\\ &-|t_b| * |NEG| * P_n(w_a) * log(\sigma(-{e_{w_a}}^T ct^b))\end{aligned} \tag{11}$$

Following the work done by Levy and Goldberg [13], we assume that the objectives of different term-document pairs are independent to each other. Therefore we can directly optimize the objective of each specific pair. Without loss of generality, *n* is chosen to be *1*. We take derivatives of objectives in equation 10 and 11 with respect to $w_a^T ct^b$ and compare them to zero. In both cases, the objectives are optimized when the inner product of specific term-document pairs equals to the shifted pointwise mutual information (SPMI) of them:

$$ {w_a}^T ct^b = log(\frac{c(w_a, t_b)}{|t_b| * P_1(w_a)}) - log(|NEG|) \tag{12}$$

Therefore, optimizing equation 8 is implicitly factorizing a SPMI matrix of term-document type. For PCD, we can easily prove that it utilizes no more information than term-term matrix and it is implicitly factorizing SPMI matrix of term-term type.

It is worth mentioning that assuming the objectives of different term-document pairs are independent is not realistic, especially in term-document case. A word may occur in many texts and a text always contains multiple words. A word can affect many objectives, so does a text. The independence of the objectives is a hypothesis that is far from the real situation. However, the analysis above inspires us to factorize this matrix to obtain improved text representations.

3.2 Shifted Positive PMI Matrix of Term-document Type Factorization

Shifted PMI matrix can not be directly factorized since it contains too many $-\infty$ (*log0*) values, which correspond to the term-document pairs that are never observed in the dataset. A well-known substitution for PMI matrix is positive PMI (PPMI). We factorize shifted positive PMI (SPPMI) matrix of term-document type and it is defined as follows:

$$ max(PMI(w, t) - log(|NEG|), 0) \tag{13}$$

Levy and Goldberg [13] and Levy et al. [14] factorize SPPMI term-term matrix via SVD for acquiring dense word and context vectors. Since all negative values are replaced by zeros, SPPMI term-term matrix lose the information about which term pairs are negatively associated and to what extent.

However, it is not the case for SPPMI matrix of term-document type. We find that PMI term-document matrix usually contains rare negative values besides $-\infty$. Moreover, it is better to assume term-document pairs are uninformative rather than negatively correlated if they are not found in the dataset, because a text only includes hundreds of words, which is small compared to vocabulary size. Viewed from this point, SPPMI is a relatively ideal matrix to be factorized. Experimental results show that text representation obtained by factorizing this novel co-occurrence matrix can compete with or even outperform state-of-the-art baselines.

4 Word Analogy Experiment

4.1 Datasets and Experimental Setup

The word analogy dataset proposed by Mikolov et al. [18] is to evaluate linguistic regularities of word representations. Questions in this dataset are in the form: ‘a is to b as c is to _?’, which are answered by finding the nearest neighbor of e_a-e_b+e_c. Training corpus used for word analogy task varies among different published results, and we choose a comparatively widely used corpora Wikipedia2010 [4] as the training data. Pre-processing includes tokenization, lowercasing and substituting number with special character.

Stochastic gradient descent (SGD) is used for objective optimization. We find that two distance measures in table 1 can be used interchangeably as long as they are used with suitable hyper-parameters, such as learning rate and epoches. Here, distance measure 1 is used for PCD and distance measure 2 is used for CCD. Two state-of-the-art local context embedding models, skip-gram (SG) and continues bag-of-words (CBOW), are used as alternatives for integration. The above training protocols are applied to all experiments in this paper.

[4] http://nlp.stanford.edu/data/WestburyLab.wikicorp.201004.txt.bz2

4.2 Integrated Model vs Local Model

As shown in table 2, when the global information is introduced into the models, significant improvements are obtained on semantic analogy questions. Intuitively, global information can hardly provide any information for capturing syntactic regularities. In this sense, global information is the noise and may hurt the models performance in syntactic analogy questions. However, to our surprise, accuracies on syntactic analogy questions do not decline when a certain degree of global information is introduced. Overall, significant improvements on total accuracies are obtained.

To further understand why global information is beneficial for capturing semantic analogy regularities, we analyze some mistakes made by local models. We discover that local models give the wrong answers mainly for the reason that they fail to distinguish words which have similar semantic meanings. Take an analogy question 'son, daughter, grandfather, _?' for example. The correct answer is 'grandmother', but the local model returns the wrong answer 'granddaughter'. We notice that when the model is trained in local manner, the embedding of these two words are very close. Local information is not enough to distinguish these two words. However, more information is available when global information is introduced. For example, 'aged', 'life', 'maternal' frequently occur in the global contexts of 'grandmother', while they seldom occur in the global contexts of 'granddaughter'. These different global contexts can help to distinguish the semantics of these two words.

Table 2. Comparison of the local and integrated models. For PCD, λ and $|POS|$ are set to be 0.1 and 5 respectively. For CCD, λ is set to be 0.6. Hyper-parameter settings of the local embedding models follow the word2vec toolkit.

Dim.	Model	Sem.	Syn.	Model	Sem.	Syn.
50	CBOW	55.7	59.9	SG	45.9	50.7
	+PCD	+4.0	-0.6	+PCD	+3.7	+0.3
	+CCD	+3.8	+1.6	+CCD	+4.4	+2.5
100	CBOW	69.5	71.0	SG	62.7	66.0
	+PCD	+3.9	+0.1	+PCD	+3.7	+0.2
	+CCD	+5.1	+1.5	+CCD	+4.1	-0.1

4.3 Comparison of Word Embedding Models

Different state-of-the-art word embedding models are compared in table 3. The corpus size has been shown to be a minor factor compared to the embedding dimensions. Therefore, we group results according to the dimensions. Here, we still list the corpus size for keeping consistent with other researches.

We can observe that CBOW has provided strong baselines on word analogy dataset. By introducing global information upon CBOW, more competitive results are achieved. We can observe that our models perform consistently better than previous state-of-the-art approaches in all dimension settings.

PDC and HDC also introduce global information into word embeddings. The source of superiority of our models to PDC and HDC comes from the choice of λ values, which controls the degrees of global information utilized during the training. Suitable λ can enhance the accuracy in semantic questions significantly without hurting the accuracy in syntactic questions. We also find that different types of word analogy tasks favor different λ, which will be further explored in our future work.

Table 3. Comparison of different word embedding models on word analogy task. The results are grouped according to the dimensions of word embedding. The best methods in each group are underlined and the best in the whole table are also in bold

Model	Dim.	Size	Sem.	Syn.	Tot.
C&W[18]	50	0.66B	9.3	12.3	11.0
GCANLM[18]	50	1B	13.3	11.6	12.3
GLOVE[20]	50	6B	48.5	44.4	46.2
CBOW	50	1B	55.7	59.9	58.3
SG	50	1B	45.9	50.7	48.9
PDC[22]	50	1B	61.2	55.1	57.9
HDC[22]	50	1B	57.8	49.8	53.4
CCD_{CBOW}	50	1B	59.5	61.5	60.7
CCD_{SG}	50	1B	50.3	53.2	51.8
GLOVE[20]	100	1.6B	67.5	54.3	60.3
CBOW	100	1B	69.5	71.0	70.4
SG	100	1B	62.7	66.0	64.7
PDC[22]	100	1B	72.8	68.4	70.4
HDC[22]	100	1B	69.6	64.3	66.7
CCD_{CBOW}	100	1B	74.6	72.5	73.3
CCD_{SG}	100	1B	66.8	65.9	66.3
GLOVE[20]	300	6B	77.4	67.0	71.7
GLOVE[20]	300	42B	81.9	69.3	75.0
CBOW	300	1B	74.6	74.0	74.2
PDC[22]	300	1B	79.6	70.5	74.8
HDC[22]	300	1B	79.7	67.7	73.1
CCD_{CBOW}	300	1B	**82.5**	**75.4**	**78.1**

5 Sentiment Analysis Experiment

5.1 Datasets and Experimental Setup

Four sentiment analysis datasets are used to evaluate the effectiveness of our models. Datasets RT-s and Subj include sentence-level texts while IMDB and RT-2k include document-level texts. Since RT-s and RT-2k datasets only contain limited snippets or documents, additional texts in IMDB dataset are added to them during the training process.

Text embeddings obtained by our models can be regarded as texts features and then fed to logistic regression classifier [6]. 10% of the training set is selected as the validation set to identify optimal hyper-parameters, such as learning rate, $|POS|$ and λ. IMDB dataset has train/test split. The rest three datasets are evaluated by 10-fold cross-validation.

5.2 Words Semantic and Syntactic Relatedness Analysis

We evaluate the quality of word embeddings by judging if the top k nearest neighbors are semantic or syntactic related to the target word. Models are trained on a movie review dataset, IMDB. Both qualitative and quantitative results are presented, which shed some light on the reason why global information is preferred for sentiment analysis tasks.

From Table 4, we can observe the local model tends to return syntactic related words, while the global model tends to return semantic related words. For example, word 'best' is the neighbor of word 'worst' when the local model is used. Both words are superlative adjectives, but they have opposite sentiment polarities. When trained in global manner, word 'worst' has the neighbor word '0/10', which indicates the lowest user rating score in a movie review. Though they have different POS tags, they share exactly the same sentiment tendency.

In addition to just giving several examples and understanding them intuitively, word embedding properties are further analyzed quantitatively. Two widely used evaluation criteria in information retrieval literature, average mean precision and DCG@10, are respectively used to evaluate the syntactic and semantic relevance of ranked neighbor lists to the target words. Specifically, 100 target words are sampled in the corpus, and top 30 neighbors of each word are obtained. Whether two words are syntactically related are judged by checking if they have the same POS tags. The semantic relatedness between two words are obtained from the average of judgments from 5 persons. Figure 3 demonstrates that when global information is increasingly introduced into the model, the embeddings reflect more semantic information and are less constrained by syntactic regularities.

Table 4. Illustration of the nearest neighbors of the target words.

Target Word	Neighbors	
	By Local Model	By Global Model
amazing	great, wonderful	10/10, amazingly
worst	dumbest, best	0/10, zero
worthless	talentless, untalented	1/10, lowest

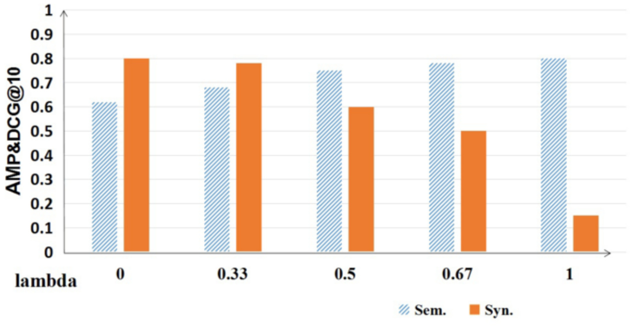

Figure 3. Evaluating the semantic and syntactic information contained in word embedding quantitatively.

5.3 Sentiment Analysis Prefer Global Information

As discussed in Section 2.4, two simple neural bag-of-words methods, PV and VecAvg, are used for generating text embeddings. From Figure 4, we can observe that more global information is preferred for document-level sentiment analysis tasks. Nearly 5 percent improvements are witnessed when global information is introduced. In fact, only about 2 percent improvements are obtained when word order information is taken into consideration in [24]. In this sense global information is of vital importance for document-level sentiment analysis. The performances of PCD and CCD are almost the same on document-level dataset. Therefore, Figure 3 only shows accuracies in the CCD case for the sake of space saving. For sentence-level datasets, introducing global information can not improve accuracy significantly since local windows usually already cover most of the sentences. However, strong results are obtained by using the cluster-driven models standalone, which requires less training time and computational resources.

From Figure 4, we can also observe that PV does not perform better than VecAvg. In contrast to the conclusion from [12], we discover that PV is not superior to VecAvg. In fact, they are both essentially bag-of-words models, where order information is totally ignored.

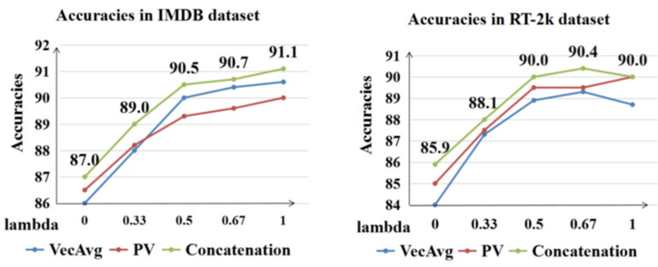

Figure 4. Accuracies on two document level datasets when global information is introduced in different degrees. Concatenation of PV and VecAvg performs better.

Table 5. Results from row 3 and 4 are from Mesnil et al [17]. Their work publishes the source code[5] and argues that the results provided by Le and Mikolov [12] can not be reproduced. For document-level datasets, integrated models are used while for sentence-level datasets, the cluster-driven models are used standalone. VecAvg is used to generate text embedding from word embedding. CON. (row 9) represents the concatenation of VecAvg and PV. The models are grouped according to how they exploit information in the text. The best methods in each group are underlined and the best in the whole table are also in bold.

Category	Model	RTs	Subj	IMDB	RT2k
bag-of-words	SVM-uni[24]	76.2	90.8	87.0	86.3
	NBSVM-uni[24]	78.1	92.4	88.3	87.8
	PV-DM[17]	76.9	91.7	89.6	88.8
	PV-DBOW[17]	76.1	90.1	89.1	88.7
	DAN-RAND[9]	77.3	-	88.8	-
	DAN[9]	80.3	-	89.4	-
	PCD	78.0	92.4	90.4	89.7
	CCD	75.4	90.9	90.6	90.1
	CON.	78.5	92.6	91.1	**90.4**
words order	SVM-bi[24]	77.7	91.7	89.2	87.4
	NBSVM-bi[24]	79.4	93.2	91.2	89.5
	NBSVM-tri[17]	-	-	91.9	-
	RNN-LM[17]	-	-	86.6	-
	Ensemble[17]	-	-	92.6	-
	SA-LSTM[3]	-	-	**92.8**	-
	CNN[10]	**81.5**	**93.6**	-	-
complex structure	DCNN[5]	-	-	89.4	-
	RecNN[21]	77.7	-	-	-
	RecNN-RNN[15]	-	-	87.0	-
	WNN[15]	77.8	-	90.2	-
	BENN[15]	77.2	-	91.0	-

5.4 Comparison of Sentiment Analysis Models

In Table 5, our models are compared with state-of-the-art sentiment analysis techniques, which are categorized according to how they exploit information in the text. One of the simplest representations is bag-of-words (BOW), where order information is totally discarded. Though BOW seems to be oversimplified, it still enjoys the advantages of being efficient, robust and concise. Word order is often important for text understanding. Bag-of-ngrams models use n-grams as features to capture words order in short context. CNNs use convolutional filters to extract n-gram information from texts. RNNs model texts sequentially and in theory can capture long-distance patterns in natural languages. Beyond word orders, more complex information such as syntax, relations among sentences is considered to train better text representations. Though information such as order and syntax is important for understanding texts, it always comes at a cost. We surprisingly observe that, even though our models are essentially bag-

[5] http://github.com/mesnilgr/iclr15

of-words models, they can even compete with models which exploit complex information of texts. Since our models ignore word order and syntactic information, they require less training time and computational resources compared to other state-of-the-art approaches.

Our models are also robust and concise. They perform well on both sentence and document level datasets. In contrast, models like CNNs and RecNNs are hard to extend to document-level dataset. Besides that, neural networks or their combinations usually have a large number of hyper-parameters and require careful hyper-parameter tuning. Their performance also closely rely on several sub-tasks, such as pretrained word embedding and parsing.

5.5 Embedding Models vs. Count-based Models

Almost all the recent works on sentiment analysis take count-based methods as poor baselines. However, work in section 3 inspires us to factorize SPPMI matrix of term-document type. The hyperparameter includes shifted-constant $|NEG|$ and the threshold for removing low frequency words, which are chosen by validation set. We compare four approaches which utilize exact the same source of information: term-document co-occurrence matrix. As shown in table 6, the novel count-based method can achieve comparable accuracies with state-of-the-art embedding methods such as PV-DBOM and C-CD, and is even more robust when dataset is small. We can observe that the performance of embedding models is poor on RT-2k dataset, unless additional unlabeled data is included.

Table 6. Comparison between count-based and embedding methods for sentiment analysis. Results of LSA are from Maas et al. [16]. Results of CCD are different from table 5 since results in table 5 are obtained by using both local and global information. Results of CCD in table 6 only utilize global information, where only term-document matrix information is taken into consideration.

Model	IMDB	RT2k	RT2k+Unlabeled
SPPMI	89.6	89.2	89.7
LSA	84.0	82.8	-
PV-DBOW	89.6	85.4	89.5
CCD	90.5	85.7	90.0

6 Conclusion

In this paper, we introduce the cluster-driven models to exploit global information to learn better word and text embeddings. When the models are used standalone, trained word embeddings can capture rich semantics. The models can also be integrated into existing local embedding models to introduce global information of different degrees. Besides that, analyzing the model in co-occurrence matrix perspective inspires us to factorize SPPMI matrix of term-document type to obtain text representations. From experimental results we can obtain several conclusions:

- Global information enriches the semantic information contained in word embeddings. Improvements are witnessed on all experiments by introducing global information into the models.
- Bag-of-words models can still compete with complex deep neural networks when global information is exploited. We also discover that the superiority of PV comes from the introduction of global information. Training text embedding in prediction manner (PV) is not superior to word embedding average (VecAvg).
- Count-based models are not inferior to embedding models. Strong results on sentiment analysis are achieved by factorizing a novel term-document matrix.

ACKNOWLEDGEMENTS

This work is supported by the National Fundamental Research and Development Program of China (973 Program) under grant 2012CB316205, National Natural Science Foundation of China (61472428, 61003204), the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China No. 14XNLQ06 and Tencent company.

REFERENCES

[1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin, 'A neural probabilistic language model', *Journal of Machine Learning Research*, **3**, 1137–1155, (2003).

[2] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa, 'Natural language processing (almost) from scratch', *Journal of Machine Learning Research*, **12**, 2493–2537, (2011).

[3] Andrew M. Dai and Quoc V. Le, 'Semi-supervised sequence learning', in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 3079–3087, (2015).

[4] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman, 'Indexing by latent semantic analysis', *JASIS*, **41**(6), 391–407, (1990).

[5] Misha Denil, Alban Demiraj, Nal Kalchbrenner, Phil Blunsom, and Nando de Freitas, 'Modelling, visualising and summarising documents with a single convolutional neural network', *CoRR*, **abs/1406.3830**, (2014).

[6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, 'LIBLINEAR: A library for large linear classification', *Journal of Machine Learning Research*, **9**, 1871–1874, (2008).

[7] Yoav Goldberg, 'A primer on neural network models for natural language processing', *CoRR*, **abs/1510.00726**, (2015).

[8] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng, 'Improving word representations via global context and multiple word prototypes', in *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pp. 873–882, (2012).

[9] Mohit Iyyer, Varun Manjunatha, Jordan L. Boyd-Graber, and Hal Daumé III, 'Deep unordered composition rivals syntactic methods for text classification', in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 1681–1691, (2015).

[10] Yoon Kim, 'Convolutional neural networks for sentence classification', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1746–1751, (2014).

[11] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao, 'Recurrent convolutional neural networks for text classification', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 2267–2273, (2015).

[12] Quoc V. Le and Tomas Mikolov, 'Distributed representations of sentences and documents', in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1188–1196, (2014).

[13] Omer Levy and Yoav Goldberg, 'Neural word embedding as implicit matrix factorization', in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2177–2185, (2014).

[14] Omer Levy, Yoav Goldberg, and Ido Dagan, 'Improving distributional similarity with lessons learned from word embeddings', *TACL*, **3**, 211–225, (2015).

[15] Jiwei Li, 'Feature weight tuning for recursive neural networks', *CoRR*, **abs/1412.3714**, (2014).

[16] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts, 'Learning word vectors for sentiment analysis', in *The 49th Annual Meeting of the Association for*

Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA, pp. 142–150, (2011).

[17] Grégoire Mesnil, Tomas Mikolov, Marc'Aurelio Ranzato, and Yoshua Bengio, 'Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews', *CoRR*, **abs/1412.5335**, (2014).

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, 'Efficient estimation of word representations in vector space', *CoRR*, **abs/1301.3781**, (2013).

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean, 'Distributed representations of words and phrases and their compositionality', in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pp. 3111–3119, (2013).

[20] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, 'Glove: Global vectors for word representation', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543, (2014).

[21] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning, 'Semi-supervised recursive autoencoders for predicting sentiment distributions', in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 151–161, (2011).

[22] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng, 'Learning word representations by jointly modeling syntagmatic and paradigmatic relations', in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 136–145, (2015).

[23] Duyu Tang, Bing Qin, and Ting Liu, 'Document modeling with gated recurrent neural network for sentiment classification', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1422–1432, (2015).

[24] Sida I. Wang and Christopher D. Manning, 'Baselines and bigrams: Simple, good sentiment and topic classification', in *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pp. 90–94, (2012).