Geometric Relationship between Word and Context Representations

Jiangtao Feng, Xiaoqing Zheng
School of Computer Science, Fudan University, Shanghai, China
Shanghai Key Laboratory of Intelligent Information Processing
{fengjt16, zhengxq}@fudan.edu.cn

Abstract

Pre-trained distributed word representations have been proven to be useful in various natural language processing (NLP) tasks. However, the geometric basis of word representations and their relations to the representations of word's contexts has not been carefully studied yet. In this study, we first investigate such geometric relationship under a general framework, which is abstracted from some typical word representation learning approaches, and find out that only the directions of word representations are well associated to their context vector representations while the magnitudes are not. In order to make better use of the information contained in the magnitudes of word representations, we propose a hierarchical Gaussian model combined with maximum a posteriori estimation to learn word representations, and extend it to represent polysemous words. Our word representations have been evaluated on multiple NLP tasks, and the experimental results show that the proposed model achieved promising results, comparing to several popular word representations.

Introduction

Much research has been devoted to distributed word representation learning, such as (Bengio et al. 2003), (Mikolov et al. 2013a), (Pennington, Socher, and Manning 2014). In these approaches, words are mapped to dense vectors in a low-dimensional latent embedded space, and these word vectors keep meaningful linguistic characteristics that words sharing similar meanings aggregate together whereas dissimilar words repel each other. Many empirical results show that such pre-trained word representations can enhance the supervised models on a variety of NLP tasks (Collobert et al. 2011; Socher et al. 2011). One of the possible explanation why such semi-supervised systems work is that pre-trained word representations act as a regularizer by constraining parts of the parameters in an appropriate region, which leads to better generalization (Erhan et al. 2010).

Word representation learning algorithms often follow Harris's distributed hypothesis that word meanings are determined by their contexts (Harris 1954). However, the relationship between the learned word and context representations has not been carefully studied in mathematics or geometry yet.

In this paper, we study the nature of word representation learning algorithms under a general framework, and find out that the learned representation of a target word belongs to the *conic hull* formed by the representations of its contexts, which means that the directions of word representations are strongly correlated with context representations while their magnitudes are relatively neglected. Such observation can give an explanation on why word similarity measured by cosine similarity usually achieves significantly higher correlation with gold standard than that measured by Euclidean distance-based similarity measures. Inspired by this observation, we explore the feasibility to learn the word vectors whose directions and magnitudes are both taken into account by combining a hierarchical Gaussian model with maximum a posteriori estimation (MAP). The proposed model is also extended to represent a polysemous word under a specific context approximately by moving a word representation closer to the representation of the given context. Finally, our approaches were validated on several NLP tasks, and achieved promising results. Especially in word similarity tasks, our results showed strong improvements using Euclidean distance-based similarity measure.

Background & Notation

Word representation learning algorithms are usually designed to predict the central word w_i in a given context $c = \{w_0, ..., w_{i-1}, w_{i+1}, ..., w_l\}, c \in \mathcal{C}$, where l is the range of a context, and $\mathcal{C}$ is the context vocabulary. The word vocabulary $\mathcal{V}$ is usually associated with two different look-up tables $\mathcal{F}, \mathcal{G} \in \mathbb{R}^{d\times|\mathcal{V}|}$, where d is the dimensionality of the embedded vector space, and $|\mathcal{V}|$ is the vocabulary size, in order to transform a word w to its vector representations ($u_w, v_w \in \mathbb{R}^d$) respectively. The first vector ($u_w = f_{\mathcal{F}}(w)$) represents a context word, and is used to generate a dense context representation $u_c \in \mathbb{R}^d$, i.e.,

$$\begin{aligned} u_c &= f_{\mathcal{M}}(c) \\ &= f_{\mathcal{M}}(\{u_{w_0}, ..., u_{w_{i-1}}, u_{w_{i+1}}, ..., u_{w_l}\}) \end{aligned} \tag{1}$$

where $\mathcal{M}$ denotes the parameters of the context model $f_{\mathcal{M}}$. The second vector ($v_w = f_{\mathcal{G}}(w)$) represents the target word, and is used to compute the similarity $u_c^\top v_w$ between a word w and a context c, which is further used to estimate the likelihood of the word w occurring in the given context c, namely the conditional probability $P(w|c)$,

$$P(w|c) = softmax(u_c^\top v_w) = \frac{\exp(u_c^\top v_w)}{\sum_{w' \in \mathcal{V}} \exp(u_c^\top v_{w'})} \tag{2}$$

The loss function of each co-occurred word-context pair in the corpus is defined as the cross-entropy between predicted distribution and target one. By summing it up over the corpus, the loss function with parameters $\theta = \{\mathcal{F}, \mathcal{G}, \mathcal{M}\}$ can be derived,

$$L(\theta) = -\sum_{w \in \mathcal{V}} \sum_{c \in C_w} \sum_{w' \in \mathcal{V}} \delta_{ww'} \ln P(w'|c) \tag{3}$$

$$\delta_{ww'} = \begin{cases} 1 & w = w' \\ 0 & w \neq w' \end{cases} \tag{4}$$

where C_w consists of all contexts in which the word w can occur, and $\delta_{ww'}$ is Kronecker delta. Word representations can be obtained by minimizing $L(\theta)$.

Several popular distributed word representation learning models, such as NNLM (Bengio et al. 2003) and word2vec (Mikolov et al. 2013a) including continuous bag-of-word (CBOW) and skip-gram (SG), can be taken as special cases of the presented prediction-based word representation learning framework. For NNLM, $f_\mathcal{M}$ is an n-gram model, and the context representation is generated by the weighted sum of the preceding $n-1$ word vectors. For CBOW, $f_\mathcal{M}$ averages the embeddings of words in a window, and for SG, $f_\mathcal{M}$ takes only one of the surrounding word embeddings to represent the context. Thus we believe that the framework could be taken as general one, and we will discuss the geometric basis of learned word representations under this framework.

Another family of models learns word representations by factorizing a word-word matrix (Pennington, Socher, and Manning 2014; Levy and Goldberg 2014b), such as pointwise mutual information (PMI) matrix and shifted positive pointwise mutual information (SPPMI) matrix. The relationship between prediction-based models and matrix factorization-based ones has been partly discussed in some literatures, which bridges the gap between two learning philosophies. Levy and Goldberg (2014b) showed that SG with negative sampling (Mikolov et al. 2013b) is implicitly factorizing a PMI matrix, and noise-contrastive estimation (NCE) (Mnih and Kavukcuoglu 2013) is implicitly factorizing a shifted log conditional probability. The relationship between SG and GloVe was also discussed by (Pennington, Socher, and Manning 2014). In the further discussion, we will focus on the prediction-based models under the presented framework. The geometric relationships derived from matrix factorization-based models could be inferred from the conclusions of (Pennington, Socher, and Manning 2014; Levy and Goldberg 2014b) and ours, and we leave it to future work.

Sampling-Based Estimation of Softmax

It is a time-consuming task to optimize Eq. [3] because computing Eq. [2] normally involves a large vocabulary. Many methods were proposed to accelerate training process, and we here focus on two effective methods in practice: negative sampling (Mikolov et al. 2013b) and Blackout (Ji et al. 2015).

Negative Sampling Negative sampling provides an efficient way to estimate $softmax$ by sampling k words with respect to their frequencies and optimizing the similarity between a given context and each candidate word independently. The probability distribution of selecting a word w is modeled as

$$Q(w) = \frac{n(w)^\gamma}{\sum_{w' \in \mathcal{V}} n(w')^\gamma} \tag{5}$$

where $n(w)$ is the frequency of the word w, and γ is a hyperparameter to smooth the distribution. A word with higher frequency is more likely to be chosen than that with lower frequency. Equipped with negative sampling, Eq. [3] is approximated with

$$\widetilde{L}_{NS}(\theta) = -\sum_{w \in \mathcal{V}} \sum_{c \in C_w} \big[\ln \sigma(u_c^\top v_w) - \sum_{w' \in neg(w)} \ln \sigma(u_c^\top v_{w'}) \big] \tag{6}$$

where $neg(w)$ contains k negative samples. In the negative sampling, the loss function is computed from two parts, $U_w = \{u_c | c \in C_w\}$ and $\bar{U}_w = \{u_c | c \notin C_w\}$. The first part U_w contains the contexts co-occurred with the target word w, whereas the second part $\bar{U}_w$ contains the rest, which is used to differentiate data from noise (Mikolov et al. 2013b). However, the learning objective is quite different from that in Eq. [3], because Eq. [6] no longer aims to predict the target word of a given context but to judge how well a word fits that context.

Blackout Another effective method is Blackout, a variant of negative sampling by approximating the conditional probability $P(w|c)$ with

$$\widetilde{P}_B(w|c) = \frac{q(w) e^{u_c^\top v_w}}{q(w) e^{u_c^\top v_w} + \sum_{w' \in neg(w)} q(w') e^{u_c^\top v_{w'}}} \tag{7}$$

where $q(w) = Q(w)^{-1}$ is the prior probability of selecting a word w. Negative samples are retrieved as Eq. [5], and $softmax$ is computed within the scores of one positive and k negative samples, each of which is weighted by a $q(\cdot)$.

Geometric Relationship between Word and Context Representations

In order to better understand the relationship between word and context representations, we shed light on their quantitative relationship when the learning algorithm converges, or namely $L(\theta)$ reaches one of its local minima. A regularization term is usually added to constrain the magnitude of the word representations. The following loss function can be derived with some simplifications,

$$L(\theta) = -\sum_{w \in \mathcal{V}} \sum_{c \in C_w} \ln \frac{e^{u_c^\top v_w}}{\sum_{w' \in \mathcal{V}} e^{u_c^\top v_{w'}}} + \frac{\beta}{2} \sum_{w \in \mathcal{V}} v_w^\top v_w \tag{8}$$

Then the $L(\theta)$ is differentiated with respect to v_w, leading to

$$\begin{aligned}\frac{\partial L}{\partial v_w} =& - \sum_{c\in C_w} \big(1 - P(w|c)\big)u_c \\ &+ \sum_{c\notin C_w} P(w|c)u_c + \beta v_w\end{aligned} \tag{9}$$

which can be divided into two context terms (U_w and $\bar{U}_w$) and a regularization term, where U_w and $\bar{U}_w$ reflect the relationship to the related and unrelated contexts respectively for the word w. By comparing Eq. [6] and Eq. [9], we find that the relationship between word and context representations can be observed in the loss function of the negative sampling, and it can also be derived from the general loss function Eq. [3, 8] by taking partial derivation with respect to v_w. Since $\bar{U}_w$ is usually used to differentiate data from noise and to obtain nontrivial solutions in negative sampling, it could be inferred that $\bar{U}_w$ plays a similar role in Eq. [9]. Below we show that $\bar{U}_w$ can be eliminated in mathematics.

Assuming that the mean of context vectors equals to zero, namely $\mathbb{E}_{c\in\mathcal{C}}[u_c] = 0$, we can derive

$$\Big|\sum_{c\in C_w} u_c\Big| = \Big|\sum_{c\notin C_w} u_c\Big| \tag{10}$$

where $|\cdot|$ indicates the norm of a vector. Let $\mathbb{E}_{c\in C_w}[P(w|c)]$ be $1 - t_w, t_w \in [0, 1]$, where t_w is the prediction error, and is assumed to be in the same scale for each word, and then we can obtain

$$\begin{aligned}\mathbb{E}_{c\notin C_w}[P(w|c)] \approx& \frac{1}{|\mathcal{V}|-1}\mathbb{E}_{w\in\mathcal{V}}[t_w] \\ &\ll t_w = \mathbb{E}_{c\in C_w}[1 - P(w|c)]\end{aligned} \tag{11}$$

since the vocabulary size $|V|$ is normally large. By combining Eq. [10] and Eq. [11], the quantitative relationships between two context terms in Eq. [9] can be finally derived,

$$\Big|\sum_{c\notin C_w} P(w|c)u_c\Big| \ll \Big|\sum_{c\in C_w} \big(1 - P(w|c)\big)u_c\Big| \tag{12}$$

Note that Eq. [12] is not a strict consequence but a reasonable result induced from Eq. [10] and Eq. [11]. Therefore, the second term in Eq. [9] can be neglected, and it will be eliminated in the further discussion.

By setting $\frac{\partial L}{\partial v_w}$ to 0, the quantitative relationship between word and context representations can be expressed by

$$v_w \approx \sum_{c\in C_w} \frac{1}{\beta}\big(1 - P_0(w|c)\big)u_c \tag{13}$$

where $P_0(w|c)$ is the reached local minima of $P(w|c)$, showing that the trained vector representation of a target word is a linear combination of those of all its contexts with positive weights. From geometric perspectives, v_w belongs to the *conic hull* of U_w, which means that the directions of vector representations are emphasized while the magnitudes are not constrained through the context vectors.

Since the properties on the directions of the learned word vectors are emphasized while those on magnitudes are not, the direction-oriented word similarity measures such as cosine similarity are naturally favored. It could be the reason why word similarity score computed by cosine similarity usually achieves higher correlation with gold standard than that computed by Euclidean distance-based similarity which measures both directions and magnitudes[1].

To put stronger constraints on their geometric relationships, an ideal representation for a word would be the central point of the representations of its contexts,

$$v_w = \frac{1}{|C_w|}\sum_{c\in C_w} u_c \tag{14}$$

which means that v_w represents the average or general meaning of its contexts as Harris hypothesis suggests. In the next section, we will show that such geometric properties can be approximately achieved by a hierarchical Gaussian model combined with maximum a posteriori estimation.

Our Approach

As we discussed above, the magnitudes of word vectors learned from the framework are not well constrained by context vectors. Thus we describe here a new model, aiming to learn word and context representations simultaneously and to take both direction and magnitude information into account. The proposed model contains a context representation generator, which try to capture the compositionality of words in a context, a posterior probability estimator based on Bayesian approach, and a loss function defined by cross-entropy. In the rest of this section, we first investigate and discuss the derived geometric relationships. Next, an efficient method is presented to approximate Bayesian posterior probability, and finally our model is extended to represent multi-sense words.

Given an unlabeled textual corpus $\mathcal{D}$, the target word w_i and the context $c = \{w_{i-L}, ..., w_{i-1}, w_{i+1}, ..., w_{i+L}\}, c \in C_{w_i}$ is generated by an L-sized window. Each word (e.g. w_j) in the context is transformed to its vector representation (u_{w_j}) by the looking up table $\mathcal{F}$, and then a weighted sum function with a set of weight matrices $\mathcal{M} = (M_{i-L}, ..., M_{i-1}, M_{i+1}, ..., M_{i+L}) \in \mathbb{R}^{2L\times d\times d}$ and a bias vector $b \in \mathbb{R}^d$ is used to generate the context vector u_c,

$$u_c = \sum_{j=i-L, j\neq i}^{i+L} M_j u_{w_j} + b. \tag{15}$$

Diagonal matrices are used for $\mathcal{M}$ to reduce the computational complexity of context generator, which also speeds up the computation and training process.

After the context representation u_c is generated, the next question is how to build the relationship between words and contexts. For each target word w, its representation (v_w) is fetched from another look-up table $\mathcal{G}$. We model the conditional probability function $p(c|w)$ as a Gaussian distribution,

$$p(c|w) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{1}{2\sigma^2}\|u_c - v_w\|_2^2) \tag{16}$$

[1] It was shown by our word similarity experiments.

where σ^2 indicates the variance and is chosen as a hyperparameter. The Eq. [16] embodies a fundamental assumption of our model that for each word, representations of its contexts are drawn from a Gaussian distribution whose expectation is the representation of this word, i.e., $\forall c \in C_w, u_c \sim \mathcal{N}(v_w, \sigma^2 I)$, where $I \in \mathbb{R}^{d \times d}$ is an identity matrix. Speaking geometrically, v_w is the center of $u_c \in U_w$ as Eq. [14] suggests.

As a prediction-based model, our model aims to predict the target word under a given context by estimating the posterior probability $P(w|c)$ by Bayesian approach,

$$P(w|c) = \frac{p(c|w)P(w)}{\sum_{w' \in \mathcal{V}} p(c|w')P(w')} \tag{17}$$

where $P(w)$ is the prior probability, and is set to a discrete uniform distribution[2]. Finally, the loss function is defined by the cross-entropy,

$$L(\theta) = -\sum_{w \in \mathcal{V}} \sum_{c \in C_w} \ln P(w|c) \tag{18}$$

By minimizing $L(\theta)$, word representations $(\mathcal{F}, \mathcal{G})$ as well as the way to generate context representations $(\mathcal{M})$ can be obtained.

Note that now the proposed model is translation-invariant to word and context vectors, which means that the value of the loss function will not be changed by adding a constant bias to all representations. Thus, it is necessary to use a regularization term to constrain the magnitudes of representations, and make them zero-biased. Therefore, our model can be described as a hierarchical Gaussian model,

$$\begin{aligned} \forall w \in \mathcal{V}, & v_w \sim \mathcal{N}(0, I) \\ \forall c \in C_w, & u_c \sim \mathcal{N}(v_w, \sigma^2 I) \end{aligned} \tag{19}$$

indicating that word vectors follow a Gaussian distribution $\mathcal{N}(0, I)$ whereas the vector representations of contexts of the word w also follow a Gaussian distribution $\mathcal{N}(v_w, \sigma^2 I)$.

Geometric Relationships

To investigate the geometric properties of word representations learned by our model, Eq. [18] is rewritten to following form by replacing $P(w|c)$ according to Eq. [16, 17] and adding a regularization term,

$$\begin{aligned} L(\theta) = & -\sum_{w \in \mathcal{V}} \sum_{c \in C_w} \ln \frac{e^{R(w)}}{\sum_{w' \in \mathcal{V}} e^{R(w')}} \\ & + \frac{\beta}{2} \sum_{w \in \mathcal{V}} v_w^\top v_w \end{aligned} \tag{20}$$

$$R(w) = -\frac{1}{2\sigma^2} \|u_c - v_w\|_2^2 + \ln P(w) \tag{21}$$

When $L(\theta)$ is differentiated with respect to v_w, we obtain

$$\begin{aligned} \frac{\partial L}{\partial v_w} = & \sum_{c \in C_w} \big(1 - P(w|c)\big) v_w \\ & - \sum_{c \in C_w} \big(1 - P(w|c)\big) u_c \\ & + \sum_{c \notin C_w} P(w|c) u_c + \Big(\beta - \sum_{c \notin C_w} P(w|c)\Big) v_w \end{aligned} \tag{22}$$

As discussed in the previous section, the third term is used to obtain nontrivial solutions, and is relatively small comparing to the second one. The forth term can be approximated to 0 by choosing an appropriate regularization rate β. By neglecting the third and the forth terms and setting $\frac{\partial L}{\partial v_w}$ to 0, the quantitative relationship between word and context representations can be derived,

$$v_w \approx \sum_{c \in C_w} \frac{1 - P_0(w|c)}{\sum_{c' \in C_w} 1 - P_0(w|c')} u_c \tag{23}$$

where $P_0(w|c)$ is a reached local minima. The Eq. [23] shows that v_w is a linear combination of $u_c \in U_w$ with positive weights whose sum equals to 1, leading to a geometric property that v_w belongs to the *convex hull* formed by U_w. Assuming that every conditional probability $P_0(w|c)$, under ideal situation, takes the same value for all $c \in C_w$, it is obvious that v_w is the expectation of u_c, which implies that the word representation v_w is the center of the clustering formed by representations of its contexts.

Efficient Posterior Probability Approximation

Inspired by Blackout, the Bayesian probability in Eq. [17] can be approximated by sampling k negative samples and normalizing within positive and negative samples,

$$\widetilde{P}(w|c) = \frac{p(c|w)P(w)}{p(c|w)P(w) + \sum_{w' \in neg(w)} p(c|w')P(w')} \tag{24}$$

because the target distribution $P(w|c)$ is one-hot, indicating that the probability of the positive sample is significantly larger than that of negative ones, and approximating $P(w|c)$ with k negative samples does not change its value severely. The Eq. [24] provides a general method to approximate Bayesian posterior probability no matter which form $p(c|w)$ takes. Note that Blackout is a special case of our method by setting $p(c|w)$ to $e^{u_c^\top v_w}$ and $P(w)$ to $Q(w)^{-1}$.

Multi-Sense Extension

We here propose a method to extend our model to represent a word w under a specific context c with

$$v_{w|c} = (1 - \lambda) v_w + \lambda u_c \tag{25}$$

where λ is a hyperparameter to weight the influence of v_w and u_c. The philosophy behind Equation [25] is that the meaning of a word is influenced by its contexts as suggested by Harris hypothesis. The first term contains *general* information by averaging meaning of its contexts whereas

[2] Two different methods were tried on determining $P(w)$. The first method was to take $P(w)$ as parameters learned in the training, and the second was to set $P(w)$ to $Q(w)^{-1}$ as Blackout does. However, it was showed in our preliminary experiments that both of the methods were not helpful empirically.

the second term contains *specific* information which make it more sensitive to the given context. From a geometrical perspective, the Eq. [25] can be interpreted as moving the word vector, namely a point in the latent embedded space, closer to the vector of the given context, reflecting both general characteristics of the word and the specific influence of the context. The experimental results of the nearest neighbors in the next section demonstrated that $v_{w|c}$ is able to capture the phenomena of polysemy to some extent by representing a polysemous word approximately.

Experiments

We conducted four sets of experiments, including word similarity, nearest neighbors, word analogy and downstream sequence labeling tasks, to evaluate our approach. The goal of the first experiment was to investigate the performance of the proposed algorithm in several word similarity datasets. The second experiment was designed to show whether its multi-sense extension really captures the phenomena of polysemy. The third experiment was to explore the capability of the learned word representations in finding analogue words. The last experiment aimed to show how well the performance of supervised learning model was enhanced by our word embeddings.

English Wikipedia documents were used to train word representation models, and its vocabulary was reduced to $50,023$ by replacing infrequent words with an "UNKNOWN" token[3]. The compared models were trained with the toolkits provided by their authors. For our approach, we use notification "Ours" for word representations (v_w) and "Ours-MS" for multi-sense extension ($v_{w|c}$). In the training process, the dimensionality of word vectors was set to 300, the window size L to 5, the variance σ^2 to 0.5, the number of negative samples to 5, regularization rate to 10^{-3} and γ in negative sampling to 0.75. Like word2vec, sub-sampling was applied with rate 10^{-5}. The weight coefficient λ was set to 0.2[4]. The stochastic gradient decent was used to minimize the loss function with 0.025 learning rate. All results reported were averaged over ten runs.

Word Similarity

Word similarity tasks were experimented on WordSim353 (Finkelstein et al. 2002), SimLex-999 (Hill, Reichart, and Korhonen 2016) and Stanford Contextual Word Similarity (SCWS) (Huang et al. 2012). For WordSim353 and SimLex-999, CBOW, SG and GloVe were compared; for SCWS, (Huang et al. 2012), (Neelakantan et al. 2014), (Iacobacci, Pilehvar, and Navigli 2015), (Mu, Bhat, and Viswanath 2016), (Zheng et al. 2017) were compared and their best reported performances were excerpted from their papers.

Model	WordSim353		SimLex-999	
	Cos	ED	Cos	ED
CBOW	0.7095	0.3787	**0.4275**	0.2726
SG	0.7003	0.4143	0.3712	0.2622
GloVe	0.6077	0.4919	0.3698	0.3231
Ours	**0.7269**	**0.7049**	0.4072	**0.4028**

Table 1: Spearman coefficients on WordSim353 and SimLex-999.

Model	SCWS	
	Cos	ED
(Huang et al. 2012)	0.6570	-
(Neelakantan et al. 2014)	0.6910	-
(Iacobacci, Pilehvar, and Navigli 2015)	0.6240	-
(Mu, Bhat, and Viswanath 2016)	0.6367	-
(Zheng et al. 2017)	0.6990	-
Ours-MS	**0.7042**	**0.6970**

Table 2: Spearman coefficients on SCWS. "-" denotes the data that were not reported by their authors.

To evaluate whether word representation models can capture both the direction and magnitude information, the cosine similarity (Cos) and Euclidean distance-based similarity (ED), defined as

$$sim(w_1, w_2) = \left(1 + \|v_{w_1} - v_{w_2}\|_2^2\right)^{-1} \tag{26}$$

were also used to compute the similarity of each pair of words. Then the Spearman's coefficient was calculated to measure performance of word representation, namely the correlation between predicted similarities and golden similarities rated by human annotators over the test set.

The results in Table 1 and Table 2 show that our model achieved state-of-the-art performance on all the word similarity tasks except for SimLex-999 using the cosine similarity measure. It is also worth noting that, for other competitive word representations (CBOW, SG and GloVe) on WordSim353 and SimLex-999 tasks, their performance degraded severely if ED was used as a word similarity measure whereas the performance of our approach only dropped slightly, which indicates that our approach is more capable of capturing both direction and magnitude information of word representations than other compared models as predicted by the previous theoretical analysis.

Nearest Neighbors

The nearest neighbors experiment was conducted to test whether our multi-sense extension is able to model multi-sense words as it is supposed to be, and the some examples were shown in Table 3. Instead of Cos, ED was used as the measure for computing word similarity because ED take both of direction and magnitudes of vectors into account. First, we chose several typical words having multiple senses, and searched the sample sentences for each sense of those words on a dictionary website. Then their contexts were randomly sampled from these sentences. For each word w in the context c, its extensive representation $v_{w|c}$ was computed

[3] In word2vec, infrequent words, which appear less than 5 times, are ignored. Here another strategy was used to pre-process the textual corpus by replacing infrequent words with an "UNKNOWN" token. A higher threshold of frequency was also set to reduce the size of word embedding lookup tables.

[4] The weight coefficient was determined on Stanford Contextual Word Similarity tasks (Huang et al. 2012), and the experimental results show that performance remained similar for $\lambda \in [0.1, 0.3]$.

Target	Context	Nearest Neighbors
blackberry	Could I try some of that jam? Jam? That **blackberry** jam. Oh, of course, darling. Actually, it's sort of a fish berry jam. It's called caviar.	apple, strawberry, jam, mango, cherry
	He said the technology would be compatible with most smartphones, including the iPhone, **BlackBerry** and Android phone.	iphone, smartphone, android, apple, app
show	He laughs: "So you know we lost a bit of money, but we had a great time and the **show** was awesome."	sketch, episode, showcase, sitcom, premiere
	Instead, I point you to the sample application for my latest book and post snippets of that example to **show** what Spring can do for you.	talk, tell, explain, showcase, reveal
kind	If there are too many results to easily manage, use one of these buttons to see only the **kind** of file you are interested in.	sort, type, thing, moment, instance
	You bet I am proud, but what really matters to me is that she grew up to be warm and **kind**, with an easygoing, unassuming demeanor.	sort, gentle, generous, ideal, nice

Table 3: Nearest neighbors under specific contexts.

by Eq. [25], and the derived representation was used to retrieve its nearest word vectors by computing the similarity scores to all the words in the vocabulary. Finally we illustrated three representative words, "blackberry" (noun/noun), "show" (noun/verb) and "kind"(noun/adjective), in Table 3. As the table suggests, the word "blackberry", for example, was closer to other fruits under a fruit-related context whereas a smartphone-related context was given, its nearest neighbors became the words related to smartphones. Similar phenomena were also observed in the cases of "show" and "kind". The experiment demonstrates that the representation $v_{w|c}$ can truly reflect the meaning of a multi-sense word under a specific context.

Word Analogy

Google (Mikolov et al. 2013a) and MSR dataset (Mikolov, Yih, and Zweig 2013) were used to evaluate our word representation on word analogy tasks, which aims to answer the question "a is a^* as b is to [b^*]". We used two methods to compute the target word b^*, 3CosAdd (Mikolov, Yih, and Zweig 2013) and 3CosMul (Levy and Goldberg 2014a). For 3CosAdd, b^* is predicted by

$$\arg\max_{b^* \in \mathcal{V}} (cos(b^*, b - a + a^*)) \tag{27}$$

whereas for the 3CosMul, it is predicted by

$$\arg\max_{b^* \in \mathcal{V}} (cos(b^*, b) - cos(b^*, a) + cos(b^*, a^*)) \tag{28}$$

Following (Mikolov, Yih, and Zweig 2013), the vocabulary of candidate words was generated as the intersection of the vocabulary of the learned word representations and that of evaluation dataset.

Model	Google		MSR	
	3CosAdd	3CosMul	3CosAdd	3CosMul
CBOW	0.7590	0.7639	0.6762	0.6846
SG	0.7563	0.7627	0.6477	0.6599
GloVe	0.6120	0.6484	0.6717	0.6807
Ours	**0.7870**	**0.7860**	**0.7512**	**0.7527**

Table 4: Accuracy on Google and MSR datasets.

The results in Table 4 showed that our model boosted the performance on both datasets of the word analogy tasks. On Google dataset, the accuracy achieved by our model was $2.80\%/2.21\%$ higher than the maximum accurracy achieved by other models with respect to two prediction methods, whereas on MSR dataset, our model outperformed other competitive models with a significant margin around $7.50\%/6.81\%$.

Downstream Sequence Labeling

Two NLP tasks (POS-tagging and chunking) are performed to compare the performance of different pre-trained word representations. We are interested in how well the word embeddings can improve the performance of the supervised learning model rather than whether state-of-the-art results can be achieved on these tasks.

We advocate the following criterion $imp(\cdot)$ to quantify the improvement caused by pre-trained word representations,

$$imp(e) = \frac{s(g_e) - s(g_r)}{1 - s(g_r)}, s(\cdot) \in [0, 1] \tag{29}$$

where $s(g_e)$ indicates the performance of the supervised learning model g with initialized word representations e. Note that e can be pre-trained word representations or random initialization r. For example, the imp of a model with randomly initialized word embeddings, which is usually taken as a baseline model, equals to 0; for a perfect model g_p ($s(g_p) = 1$), the imp equals to 1. The Eq. [29] is inspired by *measure of belief* (Ihara 1987), and the imp measure acts as a normalizer of original performance scores, aiming to reflect the real impact of word representations on supervised learning for NLP tasks, eliminating the influence brought by the supervised model itself.

For the POS-tagging, we used the Wall Street Journal benchmark (Toutanova et al. 2003), and the performances were reported in per-word accuracy (PWA). For the chunking, the CoNLL 2000 shared task[5] was used, and performances were evaluated with the standard F1-score, the harmonic mean of precision and recall. Besides, "IOBES" tagging scheme was applied for chunking.

[5] `www.cnts.ua.ac.be/conll2000/chunking`.

We implemented the window approach networks with the sentence-level log-likelihood (WNN-SLL) of (Collobert et al. 2011) as the supervised learning model. In the experiments, WNN-SLL with random initialization was taken as the baseline system.

Model	POS-tagging		Chunking	
	PWA	*imp*	F1	*imp*
baseline	0.9419	0.0000	0.8826	0.0000
CBOW	0.9584	0.2840	0.9071	0.2087
SG	0.9597	0.3064	0.9068	0.2061
GloVe	0.9592	0.2978	0.9002	0.1499
Ours	0.9610	0.3287	0.9247	0.3586
Ours-MS	**0.9642**	**0.3838**	**0.9289**	**0.3944**

Table 5: Performance on sequence labeling tasks.

Table 5 shows that our model outperformed other compared word embedding learning algorithms on both of POS-tagging and Chunking tasks, which means that word representations pre-trained by our approach provide better initialization for the supervised learning model, leading to better generalization. It also can be seen that the highest score was achieved by our multi-sense extension. By comparing the performance of *Ours* and *Our-MS*, it demonstrated that the extension gains further improvement with a significant margin to original ones because the multi-sense version can better model multi-sense words as expected.

Related Work

After the pioneer NNLM (Bengio et al. 2003) were proposed to learn distributed word representations, many methods have been proposed for word representation learning, such as word2vec (Mikolov et al. 2013a) and GloVe (Pennington, Socher, and Manning 2014). These methods offer a creative way to learn word embeddings and achieve high performance on word similarity benchmarks. However, the geometric relationships between word and context representations underlying their methods has not been carefully studied yet. Our paper aims to investigate such relationship in mathematics when the algorithm reaches one of its local minima, and proposes an improved approach to train word representations according to our findings.

In order to obtain word representations with multiple senses, many studies have been devoted to multi-prototype models, including (Reisinger and Mooney 2010; Huang et al. 2012; Neelakantan et al. 2014; Iacobacci, Pilehvar, and Navigli 2015; Zheng et al. 2017). However, multi-prototype models suffer from a problem that it is hard to determine the number of prototypes for each word. In our model, a sound method is proposed to approximately represent a polysemous word under a specific context, which is more efficient in usage because the computational intense word disambiguation process is not required.

The motivation of our paper is similar to a recent paper (Mu, Bhat, and Viswanath 2016) which also discussed the geometry of word representations. However, they focus on the polysemy representations whereas we pay attention to word representations in a more general case.

The most relevant work was proposed by Vilnis and McCallum (2014) that focuses on representing words with Gaussian distributions whereas in our hierarchical model, words and contexts are represented by points, and Gaussian distribution is used for modeling their relationship. Besides, their model is learned by optimizing the energy of co-occurred word-word pairs in the corpus whereas our model targets to predict the central word with a set of words in a given context based on Bayesian approach. The way to generate context representations is also learned at the same time in the learning process. Furthermore, in their work, each word is associated with only one representation and unable to model polysemy, but we proposed a novel extension to represent multi-sense words.

Conclusions

We studied the geometric characteristics of word and context representations and their relationship. The quantitative relationship between word and context representations has been investigated under a general framework abstracted from some typical word embedding learning algorithms, such as NNLM, CBOW and SG. We proved that the representation of each word learned by their approaches belongs to the conic hull formed by representations of its contexts, indicating that the directions of word vectors are well constrained by the context representations while their magnitudes are not. Inspired by such observation, we proposed a joint word and context representation learning approach based on the combination of hierarchical Gaussian model and maximum a posteriori estimation. In contrast to the existing typical approaches, the representation of each word learned by our model is in the convex hull of representations of its contexts, which puts constraints on the word vectors both in direction and magnitude. The geometric characteristics brought by the convex hull allow us to easily extend our model in the ability to represent polysemous words, and such ability was demonstrated by the nearest neighbors experiment. The study on the geometric basis of word representations also gives a possible explanation of why the existing typical models usually achieve higher performances on word similarity benchmarks using the cosine similarity as a similarity measure than Euclidean distance-based one. The experiments on word similarity tasks confirmed such explanation empirically and also showed that our approach performed well using both the similarity measures. Besides, the experiments on multiple downstream sequence labeling tasks have demonstrated that our word representations can improve the performance of the neural network-based NLP systems more than other competitors can. The source code of our model is available at `https://github.com/JiangtaoFeng/HGM-MAP`.

Acknowledgments

We would like to thank anonymous reviewers for their valuable comments, and we also thank Mengxiao Lin, Yi Zhou and Haoyuan Peng for their helpful discussions. This work was partly supported by a grant from Shanghai Municipal Natural Science Foundation (No.15511104303).

References

Bengio, Y.; Ducharme, R.; Vincent, P.; and Janvin, C. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3(6):1137–1155.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.-A.; and Vincent, P. 2010. Why does unsupervised pre-training help deep learning. *Journal of Machine Learning Research* 11:625–660.

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2002. Placing search in context: the concept revisited. *ACM Transactions on Information Systems* 20(1):116–131.

Harris, Z. S. 1954. Distributional structure. *Word* 10(2-3):146–162.

Hill, F.; Reichart, R.; and Korhonen, A. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 873–882. Association for Computational Linguistics.

Iacobacci, I.; Pilehvar, M. T.; and Navigli, R. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 95–105. Beijing, China: Association for Computational Linguistics.

Ihara, J. 1987. Extension of conditional probability and measures of belief and disbelief in a hypothesis based on uncertain evidence. *IEEE transactions on pattern analysis and machine intelligence* (4):561–568.

Ji, S.; Vishwanathan, S.; Satish, N.; Anderson, M. J.; and Dubey, P. 2015. Blackout: Speeding up recurrent neural network language models with very large vocabularies. *arXiv preprint arXiv:1511.06909*.

Levy, O., and Goldberg, Y. 2014a. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 171–180. Ann Arbor, Michigan: Association for Computational Linguistics.

Levy, O., and Goldberg, Y. 2014b. Neural word embedding as implicit matrix factorization. In Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N. D.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc. 2177–2185.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. 3111–3119.

Mikolov, T.; Yih, W.-t.; and Zweig, G. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–751. Atlanta, Georgia: Association for Computational Linguistics.

Mnih, A., and Kavukcuoglu, K. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc. 2265–2273.

Mu, J.; Bhat, S.; and Viswanath, P. 2016. Geometry of polysemy. *CoRR* abs/1610.07569.

Neelakantan, A.; Shankar, J.; Passos, A.; and McCallum, A. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1059–1069. Doha, Qatar: Association for Computational Linguistics.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, volume 14, 1532–1543.

Reisinger, J., and Mooney, R. J. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 109–117. Association for Computational Linguistics.

Socher, R.; Lin, C. C.; Manning, C.; and Ng, A. Y. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, 129–136.

Toutanova, K.; Klein, D.; Manning, C. D.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 173–180. Association for Computational Linguistics.

Vilnis, L., and McCallum, A. 2014. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*.

Zheng, X.; Feng, J.; Chen, Y.; Peng, H.; and Zhang, W. 2017. Learning context-specific word/character embeddings. In *AAAI Conference on Artificial Intelligence*.