Learning Word Representations by Jointly Modeling Syntagmatic and Paradigmatic Relations

Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and **Xueqi Cheng**
CAS Key Lab of Network Data Science and Technology
Institute of Computing Technology
Chinese Academy of Sciences, China
ofey.sunfei@gmail.com
{guojiafeng, lanyanyan, junxu, cxq}@ict.ac.cn

Abstract

Vector space representation of words has been widely used to capture fine-grained linguistic regularities, and proven to be successful in various natural language processing tasks in recent years. However, existing models for learning word representations focus on either syntagmatic or paradigmatic relations alone. In this paper, we argue that it is beneficial to jointly modeling both relations so that we can not only encode different types of linguistic properties in a unified way, but also boost the representation learning due to the mutual enhancement between these two types of relations. We propose two novel distributional models for word representation using both syntagmatic and paradigmatic relations via a joint training objective. The proposed models are trained on a public Wikipedia corpus, and the learned representations are evaluated on word analogy and word similarity tasks. The results demonstrate that the proposed models can perform significantly better than all the state-of-the-art baseline methods on both tasks.

1 Introduction

Vector space models of language represent each word with a real-valued vector that captures both semantic and syntactic information of the word. The representations can be used as basic features in a variety of applications, such as information retrieval (Manning et al., 2008), named entity recognition (Collobert et al., 2011), question answering (Tellex et al., 2003), disambiguation (Schütze, 1998), and parsing (Socher et al., 2011).

A common paradigm for acquiring such representations is based on the *distributional hypothesis* (Harris, 1954; Firth, 1957), which states that words occurring in similar contexts tend to have similar meanings. Based on this hypothesis, various models on learning word representations have been proposed during the last two decades.

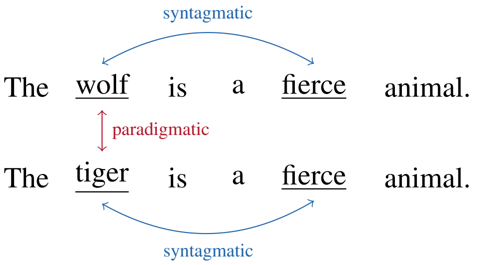

Figure 1: Example for syntagmatic and paradigmatic relations.

According to the leveraged distributional information, existing models can be grouped into two categories (Sahlgren, 2008). The first category mainly concerns the *syntagmatic relations* among the words, which relate the words that co-occur in the same text region. For example, “wolf” is close to “fierce” since they often co-occur in a sentence, as shown in Figure 1. This type of models learn the distributional representations of words based on the text region that the words occur in, as exemplified by Latent Semantic Analysis (LSA) model (Deerwester et al., 1990) and Non-negative Matrix Factorization (NMF) model (Lee and Seung, 1999). The second category mainly captures *paradigmatic relations*, which relate words that occur with similar contexts but may *not* cooccur in the text. For example, “wolf” is close to “tiger” since they often have similar context words. This type of models learn the word representations based on the surrounding words, as exemplified by the Hyperspace Analogue to Language (HAL) model (Lund et al., 1995), Continuous Bag-of-Words (CBOW) model and Skip-Gram (SG) model (Mikolov et al., 2013a).

In this work, we argue that it is important to take both syntagmatic and paradigmatic relations into account to build a good distributional model. Firstly, in distributional meaning acquisition, it is expected that a good representation should be able to encode a bunch of linguistic properties. For example, it can put semantically related words close (*e.g.*, "microsoft" and "office"), and also be able to capture syntactic regularities like "big is to bigger as deep is to deeper". Obviously, these linguistic properties are related to both syntagmatic and paradigmatic relations, and cannot be well modeled by either alone. Secondly, syntagmatic and paradigmatic relations are complimentary rather than conflicted in representation learning. That is relating the words that co-occur within the same text region (*e.g.*, "wolf" and "fierce" as well as "tiger" and "fierce") can better relate words that occur with similar contexts (*e.g.*, "wolf" and "tiger"), and vice versa.

Based on the above analysis, we propose two new distributional models for word representation using both syntagmatic and paradigmatic relations. Specifically, we learn the distributional representations of words based on the text region (*i.e.*, the document) that the words occur in as well as the surrounding words (*i.e.*, word sequences within some window size). By combining these two types of relations either in a parallel or a hierarchical way, we obtain two different joint training objectives for word representation learning. We evaluate our new models in two tasks, *i.e.*, word analogy and word similarity. The experimental results demonstrate that the proposed models can perform significantly better than all of the state-of-the-art baseline methods in both of the tasks.

2 Related Work

The distributional hypothesis has provided the foundation for a class of statistical methods for word representation learning. According to the leveraged distributional information, existing models can be grouped into two categories, *i.e.*, syntagmatic models and paradigmatic models.

Syntagmatic models concern combinatorial relations between words (*i.e.*, syntagmatic relations), which relate words that co-occur within the same text region (*e.g.*, sentence, paragraph or document).

For example, sentences have been used as the text region to acquire co-occurrence information by (Rubenstein and Goodenough, 1965; Miller and Charles, 1991). However, as pointed our by Picard (1999), the smaller the context regions are that we use to collect syntagmatic information, the worse the sparse-data problem will be for the resulting representation. Therefore, syntagmatic models tend to favor the use of larger text regions as context. Specifically, a document is often taken as a natural context of a word following the literature of information retrieval. In these methods, a words-by-documents co-occurrence matrix is built to collect the distributional information, where the entry indicates the (normalized) frequency of a word in a document. A low-rank decomposition is then conducted to learn the distributional word representations. For example, LSA (Deerwester et al., 1990) employs singular value decomposition by assuming the decomposed matrices to be orthogonal. In (Lee and Seung, 1999), non-negative matrix factorization is conducted over the words-by-documents matrix to learn the word representations.

Paradigmatic models concern substitutional relations between words (*i.e.*, paradigmatic relations), which relate words that occur in the same context but may not at the same time. Unlike syntagmatic model, paradigmatic models typically collect distributional information in a words-by-words co-occurrence matrix, where entries indicate how many times words occur together within a context window of some size.

For example, the Hyperspace Analogue to Language (HAL) model (Lund et al., 1995) constructed a high-dimensional vector for words based on the word co-occurrence matrix from a large corpus of text. However, a major problem with HAL is that the similarity measure will be dominated by the most frequent words due to its weight scheme. Various methods have been proposed to address the drawback of HAL. For example, the Correlated Occurrence Analogue to Lexical Semantic (COALS) (Rohde et al., 2006) transformed the co-occurrence matrix by an entropy or correlation based normalization. Bullinaria and Levy (2007), and Levy and Goldberg (2014b) suggested that positive pointwise mutual information (PPMI) is a good transformation. More recently, Lebret and Collobert (2014) obtained the word representations through a Hellinger PCA (HPCA) of the words-by-words co-occurrence matrix. Pennington et al. (2014) explicitly factorizes the words-by-words co-occurrence matrix to obtain the Global Vectors (GloVe) for word representation.

Alternatively, neural probabilistic language models (NPLMs) (Bengio et al., 2003) learn word representations by predicting the next word given previously seen words. Unfortunately, the training of NPLMs is quite time consuming, since computing probabilities in such model requires normalizing over the entire vocabulary. Recently, Mnih and Teh (2012) applied Noise Contrastive Estimation (NCE) to approximately maximize the probability of the softmax in NPLM. Mikolov et al. (2013a) further proposed continuous bag-of-words (CBOW) and skip-gram (SG) models, which use a simple single-layer architecture based on inner product between two word vectors. Both models can be learned efficiently via a simple variant of Noise Contrastive Estimation, *i.e.*, Negative sampling (NS) (Mikolov et al., 2013b).

3 Our Models

In this paper, we argue that it is important to jointly model both syntagmatic and paradigmatic relations to learn good word representations. In this way, we not only encode different types of linguistic properties in a unified way, but also boost the representation learning due to the mutual enhancement between these two types of relations.

We propose two joint models that learn the distributional representations of words based on both the text region that the words occur in (*i.e.*, syntagmatic relations) and the surrounding words (*i.e.*, paradigmatic relations). To model syntagmatic relations, we follow the previous work (Deerwester et al., 1990; Lee and Seung, 1999) to take document as a nature text region of a word. To model paradigmatic relations, we are inspired by the recent work from Mikolov et al. (Mikolov et al., 2013a; Mikolov et al., 2013b), where simple models over word sequences are introduced for efficient and effective word representation learning.

In the following, we introduce the notations used in this paper, followed by detailed model descriptions, ending with some discussions of the proposed models.

3.1 Notation

Before presenting our models, we first list the notations used in this paper. Let $D{=}\{d_1, \ldots, d_N\}$ denote a corpus of N documents over the word vocabulary W. The contexts for word $w_i^n{\in}W$ (*i.e.* i-th word in document d_n) are the words surrounding it in an L-sized window $(c_{i-L}^n, \ldots, c_{i-1}^n, c_{i+1}^n, \ldots, c_{i+L}^n) \in H$, where $c_j^n \in W, j{\in}\{i{-}L, \ldots, i{-}1, i{+}1, \ldots, i{+}L\}$. Each document $d \in D$, each word $w \in W$ and each context $c \in W$ is associated with a vector $\vec{d} \in \mathbb{R}^K$, $\vec{w} \in \mathbb{R}^K$ and $\vec{c} \in \mathbb{R}^K$, respectively, where K is the embedding dimensionality. The entries in the vectors are treated as parameters to be learned.

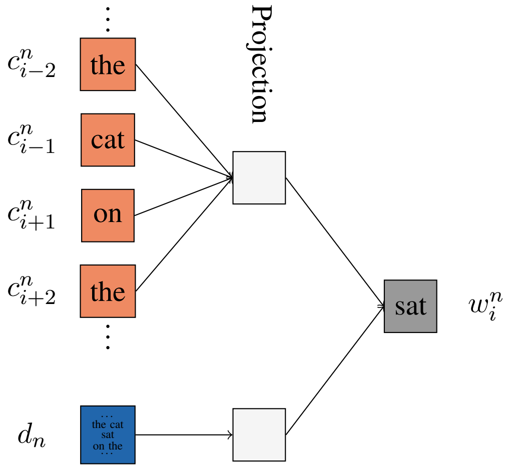

Figure 2: The framework for PDC model. Four words (“the”, “cat”, “on” and “the”) are used to predict the center word (“sat”). Besides, the document in which the word sequence occurs is also used to predict the center word (“sat”).

3.2 Parallel Document Context Model

The first proposed model architecture is shown in Figure 2. In this model, a target word is predicted by its surrounding context, as well as the document it occurs in. The former prediction task captures the paradigmatic relations, since words with similar context will tend to have similar representations. While the latter prediction task models the syntagmatic relations, since words co-occur in the same document will tend to have similar representations. More detailed analysis on this will be presented in Section 3.4. The model can be viewed as an extension of CBOW model (Mikolov et al., 2013a), by adding an extra document branch. Since both the context and document are parallel in predicting the target word, we call this model the Parallel Document Context (PDC) model.

More formally, the objective function of PDC model is the log likelihood of all words

$$\ell = \sum_{n=1}^{N} \sum_{w_i^n \in d_n} \big(\log p(w_i^n|h_i^n) + \log p(w_i^n|d_n)\big)$$

where h_i^n denotes the projection of w_i^n's contexts, defined as

$$h_i^n = f(c_{i-L}^n, \dots, c_{i-1}^n, c_{i+1}^n, \dots, c_{i+L}^n)$$

where $f(\cdot)$ can be sum, average, concatenate or max pooling of context vectors[1]. In this paper, we use average, as that of `word2vec` tool.

We use softmax function to define the probabilities $p(w_i^n|h_i^n)$ and $p(w_i^n|d_n)$ as follows:

$$p(w_i^n|h_i^n) = \frac{\exp(\vec{w_i^n} \cdot \vec{h_i^n})}{\sum_{w \in W} \exp(\vec{w} \cdot \vec{h_i^n})} \tag{1}$$

$$p(w_i^n|d_n) = \frac{\exp(\vec{w_i^n} \cdot \vec{d_n})}{\sum_{w \in W} \exp(\vec{w} \cdot \vec{d_n})} \tag{2}$$

where $\vec{h_i^n}$ denotes projected vector of w_i^n's contexts.

To learn the model, we adopt the negative sampling technique (Mikolov et al., 2013b) for efficient learning since the original objective is intractable for direct optimization. The negative sampling actually defines an alternate training objective function as follows

$$\begin{aligned}\ell = \sum_{n=1}^{N} \sum_{w_i^n \in d_n} \big(&\log \sigma(\vec{w_i^n} \cdot \vec{h_i^n}) + \log \sigma(\vec{w_i^n} \cdot \vec{d_n}) \\ &+ k \cdot \mathbb{E}_{w' \sim P_{\text{nw}}} \log \sigma(-\vec{w'} \cdot \vec{h_i^n}) \\ &+ k \cdot \mathbb{E}_{w' \sim P_{\text{nw}}} \log \sigma(-\vec{w'} \cdot \vec{d_n})\big)\end{aligned} \tag{3}$$

where $\sigma(x) = 1/(1 + \exp(-x))$, k is the number of "negative" samples, w' denotes the sampled word, and P_{nw} denotes the distribution of negative word samples. We use stochastic gradient descent (SGD) for optimization, and the gradient is calculated via back-propagation algorithm.

3.3 Hierarchical Document Context Model

Since the above PDC model can be viewed as an extension of CBOW model, it is natural to introduce the same document-word prediction layer into the SG model. This becomes our second model architecture as shown in Figure 3. Specifically, the document is used to predict a target word, and the target word is further used to predict its surrounding context words. Since the prediction is conducted in a hierarchical manner, we name this model the Hierarchical Document Context (HDC) model. Similar as the PDC model, the syntagmatic relation in HDC is modeled by the document-word prediction layer and the word-context prediction layer models the paradigmatic relation.

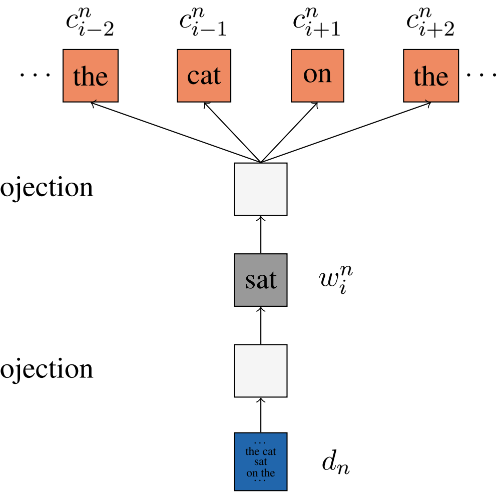

Figure 3: The framework for HDC model. The document is used to predict the target word ("sat"). Then, the word ("sat") is used to predict the surrounding words ("the", "cat", "on" and "the").

Formally, the objective function of HDC model is the log likelihood of all words:

$$\ell = \sum_{n=1}^{N} \sum_{w_i^n \in d_n} \Bigg(\sum_{\substack{j=i-L \\ j \neq i}}^{i+L} \log p(c_j^n|w_i^n) + \log p(w_i^n|d_n) \Bigg)$$

where $p(w_i^n|d_n)$ is defined the same as in Equation (2), and $p(c_j^n|w_i^n)$ is also defined by a softmax function as follows:

$$p(c_j^n|w_i^n) = \frac{\exp(\vec{c_j^n} \cdot \vec{w_i^n})}{\sum_{c \in W} \exp(\vec{c} \cdot \vec{w_i^n})}$$

Similarly, we adopt the negative sampling technique for learning, which defines the following

[1] Note that the context window size L can be a function of the target word w_i^n. In this paper, we use the same strategy as `word2vec` tools which uniformly samples from the set $\{1, 2, \cdots, L\}$.

training objective function

$$\begin{aligned}\ell =& \sum_{n=1}^{N} \sum_{w_i^n \in d_n} \Bigg(\sum_{\substack{j=i-L \\ j\neq i}}^{i+L} \Big(\log \sigma(\vec{c_j^n} \cdot \vec{w_i^n}) \\ &+ k \cdot \mathbb{E}_{c' \sim P_{\text{nc}}} \log \sigma(-\vec{c'} \cdot \vec{w_i^n}) \Big) \\ &+ \log \sigma(\vec{w_i^n} \cdot \vec{d_n}) + k \cdot \mathbb{E}_{w' \sim P_{\text{nw}}} \log \sigma(-\vec{w'} \cdot \vec{d_n}) \Bigg)\end{aligned}$$

where k is the number of the negative samples, c' and w' denotes the sampled context and word respectively, and P_{nc} and P_{nw} denotes the distribution of negative context and word samples respectively[2]. We also employ SGD for optimization, and calculate the gradient via back-propagation algorithm.

3.4 Discussions

In this section we first show how PDC and HDC models capture the syntagmatic and paradigmatic relations from the viewpoint of matrix factorization. We then talk about the relationship of our models with previous work.

As pointed out in (Sahlgren, 2008), to capture syntagmatic relations, the implementational basis is to collect text data in a words-by-documents co-occurrence matrix in which the entry indicates the (normalized) frequency of occurrence of a word in a document (or, some other type of text region, *e.g.*, a sentence). While the implementational basis for paradigmatic relations is to collect text data in a words-by-words co-occurrence matrix that is populated by counting how many times words occur together within the context window. We now take the proposed PDC model as an example to show how it achieves these goals, and similar results can be shown for HDC model.

The objective function of PDC with negative sampling in Equation (3) can be decomposed into the following two parts:

$$\begin{aligned}\ell_1 =& \sum_{w\in W} \sum_{h \in H} \big(\#(w,h) \cdot \log \sigma(\vec{w} \cdot \vec{h}) \\ & + k \cdot \#(h) \cdot p_{\text{nw}}(w) \log \sigma(-\vec{w} \cdot \vec{h}) \big)\end{aligned} \tag{4}$$

$$\begin{aligned}\ell_2 =& \sum_{d\in D} \sum_{w \in W} \big(\#(w,d) \cdot \log \sigma(\vec{w} \cdot \vec{d}) \\ & + k \cdot |d| \cdot p_{\text{nw}}(w) \log \sigma(-\vec{w} \cdot \vec{d}) \big)\end{aligned} \tag{5}$$

where $\#(\cdot,\cdot)$ denotes the number of times the pair $(\cdot,\cdot)$ appears in D, $\#(h) = \sum_{w\in W} \#(w,h)$, $|d|$ denotes the length of document d, the objective function ℓ_1 corresponds to the context-word prediction task and ℓ_2 corresponds to the document-word prediction task.

Following the idea introduced by (Levy and Goldberg, 2014a), it is easy to show that the solution of the objective function ℓ_1 follows that

$$\vec{w} \cdot \vec{h} = \log(\frac{\#(w,h)}{\#(h) \cdot p_{\text{nw}}(w)}) - \log k$$

and the solution of the objective function ℓ_2 follows that

$$\vec{w} \cdot \vec{d} = \log(\frac{\#(w,d)}{|d| \cdot p_{\text{nw}}(w)}) - \log k$$

It reveals that the PDC model with negative sampling is actually factorizing both a words-by-contexts co-occurrence matrix and a words-by-documents co-occurrence matrix simultaneously. In this way, we can see that the implementational basis of the PDC model is consistent with that of syntagmatic and paradigmatic models. In other words, PDC can indeed capture both syntagmatic and paradigmatic relations by processing the right distributional information. Please notice that the PDC model is not equivalent to direct combination of existing matrix factorization methods, due to the fact that the matrix entries defined in PDC model are more complicated than the simple co-occurrence frequency (Lee and Seung, 1999).

When considering existing models, one may connect our models to the Distributed Memory model of Paragraph Vectors (PV-DM) and the Distributed Bag of Words version of Paragraph Vectors (PV-DBOW) (Le and Mikolov, 2014). However, both of them are quite different from our models. In PV-DM, the paragraph vector and context vectors are averaged or concatenated to predict the next word. Therefore, the objective function of PV-DM can no longer decomposed as the PDC model as shown in Equation (4) and (5). In other words, although PV-DM leverages both paragraph and context information, it is unclear how these information is collected and used in this model. As for PV-DBOW, it simply leverages paragraph vector to predict words in the paragraph. It is easy to show that it only uses the words-by-documents co-occurrence matrix, and thus only captures syntagmatic relations.

Another close work is the Global Context-Aware Neural Language Model (GCANLM for

[2] P_{nc} is not necessary to be the same as P_{nw}.

short) (Huang et al., 2012). The model defines two scoring components that contribute to the final score of a (word sequence, document) pair. The architecture of GCANLM seems similar to our PDC model, but exhibits lots of differences as follows: (1) GCANLM employs neural networks as components while PDC resorts to simple model structure without non-linear hidden layers; (2) GCANLM uses weighted average of all word vectors to represent the document, which turns out to model words-by-words co-occurrence (*i.e.*, paradigmatic relations) again rather than words-by-documents co-occurrence (*i.e.*, syntagmatic relations); (3) GCANLM is a language model which predicts the next word given the preceding words, while PDC model leverages both preceding and succeeding contexts for prediction.

4 Experiments

In this section, we first describe our experimental settings including the corpus, hyper-parameter selections, and baseline methods. Then we compare our models with baseline methods on two tasks, *i.e.*, word analogy and word similarity. After that, we conduct some case studies to show that our model can better capture both syntagmatic and paradigmatic relations and how it improves the performances on semantic tasks.

4.1 Experimental Settings

We select Wikipedia, the largest online knowledge base, to train our models. We adopt the publicly available April 2010 dump[3] (Shaoul and Westbury, 2010), which is also used by (Huang et al., 2012; Luong et al., 2013; Neelakantan et al., 2014). The corpus in total has 3,035,070 articles and about 1 billion tokens. In preprocessing, we lowercase the corpus, remove pure digit words and non-English characters[4].

Following the practice in (Pennington et al., 2014), we set context window size as 10 and use 10 negative samples. The noise distributions for context and words are set as the same as used in (Mikolov et al., 2013a), $p_{\text{nw}}(w) \propto \#(w)^{0.75}$. We also adopt the same linear learning rate strategy described in (Mikolov et al., 2013a), where the initial learning rate of PDC model is 0.05, and HDC is 0.025. No additional regularization is used in our models.[5]

Table 1: Corpora used in baseline models.

model	corpus	size
C&W	Wikipedia 2007 + Reuters RCV1	0.85B
HPCA	Wikipedia 2012	1.6B
GloVe	Wikipedia 2014+ Gigaword5	6B
GCANLM, CBOW, SG PV-DBOW, PV-DM	Wikipedia 2010	1B

We compare our models with various state-of-the-art models including C&W (Collobert et al., 2011), GCANLM (Huang et al., 2012), CBOW, SG (Mikolov et al., 2013a), GloVe (Pennington et al., 2014), PV-DM, PV-DBOW (Le and Mikolov, 2014) and HPCA (Lebret and Collobert, 2014). For C&W, GCANLM[6], GloVe and HPCA, we use the word embeddings they provided. For CBOW and SG model, we reimplement these two models since the original `word2vec` tool uses SGD but *cannot* shuffle the data. Besides, we also implement PV-DM and PV-DBOW models due to (Le and Mikolov, 2014) has not released source codes. We train these four models on the same dataset with the same hyper-parameter settings as our models for fair comparison. The statistics of the corpora used in baseline models are shown in Table 1. Moreover, since different papers report different dimensionality, to be fair, we conduct evaluations on three dimensions (*i.e.*, 50, 100, 300) to cover the publicly available results[7].

4.2 Word Analogy

The word analogy task is introduced by Mikolov et al. (2013a) to quantitatively evaluate the linguistic regularities between pairs of word representations. The task consists of questions like "a is to b as c is to __", where __ is missing and must be guessed from the entire vocabulary. To answer such questions, we need to find a word vector $\vec{x}$, which is the closest to $\vec{b} - \vec{a} + \vec{c}$ according to the cosine similarity:

$$\arg \max_{\substack{x \in W, x \neq a \\ x \neq b,\, x \neq c}} (\vec{b} + \vec{c} - \vec{a}) \cdot \vec{x}$$

The question is judged as correctly answered only if x is exactly the answer word in the evaluation

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

[4] We ignore the words less than 20 occurrences during training.

[5] The source code and word vectors can be downloaded at `http://ofey.me/projects/wordrep/`.

[6] Here, we use GCANLM's single-prototype embedding.

[7] C&W and GCANLM only released the vectors with 50 dimensions, and HPCA released vectors with 50 and 100 dimensions.

Table 2: Results on the word analogy task. Underlined scores are the best within groups of the same dimensionality, while bold scores are the best overall.

model	size	dim	semantic	syntactic	total
C&W	0.85B	50	9.33	12.35	10.98
GCANLM	1B	50	2.6	10.94	7.34
HPCA	1.6B	50	3.36	10.42	7.2
GloVe	6B	50	48.46	44.36	46.22
CBOW	1B	50	54.38	50.04	52.01
SG	1B	50	53.73	45.15	49.04
PV-DBOW	1B	50	55.02	44.61	49.34
PV-DM	1B	50	45.08	43.57	44.25
PDC	1B	50	61.21	55.12	57.88
HDC	1B	50	57.8	49.76	53.41
HPCA	1.6B	100	4.16	16.37	10.79
GloVe	6B	100	65.34	61.26	63.11
CBOW	1B	100	70.73	63.67	66.87
SG	1B	100	67.66	59.95	63.45
PV-DBOW	1B	100	67.49	56.54	61.51
PV-DM	1B	100	57.72	59.06	58.45
PDC	1B	100	72.77	68.35	70.35
HDC	1B	100	69.57	64.26	66.67
GloVe	6B	300	77.44	67.0	71.7
CBOW	1B	300	76.2	69.22	72.39
SG	1B	300	78.9	66.05	71.88
PV-DBOW	1B	300	66.85	58.12	62.08
PV-DM	1B	300	56.88	68.79	63.39
PDC	1B	300	79.55	**70.78**	**74.76**
HDC	1B	300	**79.67**	67.69	73.13

set. The evaluation metric for this task is the percentage of questions answered correctly.

The dataset contains 5 types of semantic analogies and 9 types of syntactic analogies[8]. The semantic analogy contains 8869 questions, typically about people and place like “Beijing is to China as Paris is to France”, while the syntactic analogy contains 10,675 questions, mostly on forms of adjectives or verb tense, such as “good is to better as bad to worse”.

Result Table 2 shows the results on word analogy task. As we can see that CBOW, SG and GloVe are much stronger baselines as compare with C&W, GCANLM and HPCA. Even so, our PDC model still performs significantly better than these state-of-the-art methods (p-value < 0.01), especially with smaller vector dimensionality. More interestingly, by only training on 1 billion words, our models can outperform the GloVe model which is trained on 6 billion words. The results demonstrate that by modeling both syntagmatic and paradigmatic relations, we can learn better word representations capturing linguistic regularities.

Besides, CBOW, SG and PV-DBOW can be viewed as sub-models of our proposed models, since they use either context (*i.e.*, paradigmatic relations) or document (*i.e.*, syntagmatic relations) alone to predict the target word. By comparing with these sub-models, we can see that the PDC and HDC models can perform significantly better on both syntactic and semantic subtasks. It shows that by jointly modeling the two relations, one can boost the representation learning and better capture both semantic and syntactic regularities.

4.3 Word Similarity

Besides the word analogy task, we also evaluate our models on three different word similarity tasks, including WordSim-353 (Finkelstein et al., 2002), Stanford’s Contextual Word Similarities (SCWS) (Huang et al., 2012) and rare word (RW) (Luong et al., 2013). These datasets contain word paris together with human assigned similarity scores. We compute the Spearman rank correlation between similarity scores based on learned word representations and the human judgements. In all experiments, we removed the word pairs that cannot be found in the vocabulary.

Results Figure 4 shows results on three different word similarity datasets. First of all, our proposed PDC model always achieves the best performances on the three tasks. Besides, if we compare the PDC and HDC models with their corresponding sub-models (*i.e.*, CBOW and SG) respectively, we can see performance gain by adding syntagmatic information via document. This gain becomes even larger for rare words with low dimensionality as shown on RW dataset. Moreover, on the SCWS dataset, our PDC model using the single-prototype representations under dimensionality 50 can achieve a comparable result (65.63) to the state-of-the-art GCANLM (65.7 as the best performance reported in (Huang et al., 2012)) which uses multi-prototype vectors[9].

4.4 Case Study

Here we conduct some case studies to (1) gain some intuition on how these two relations affect

[8] `http://code.google.com/p/word2vec/source/browse/trunk/questions-words.txt`

[9] Note, in Figure 4, the performance of GCANLM is computed based on their released single-prototype vectors.

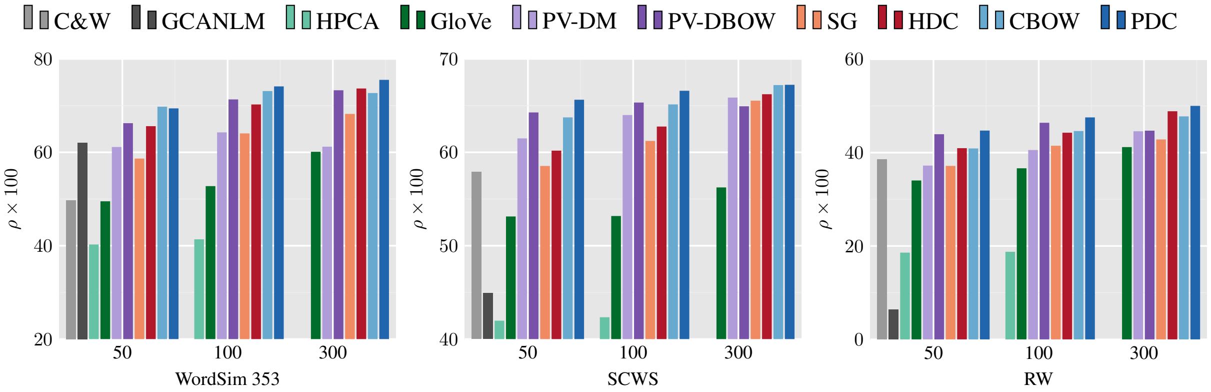

Figure 4: Spearman rank correlation on three datasets. Results are grouped by dimensionality.

Table 3: Target words and their 5 most similar words under different representations. Words in italic often co-occur with the target words, while words in bold are substitutable to the target words.

feynman	
CBOW	**einstein**, **schwinger**, **bohm**, **bethe** *relativity*
SG	**schwinger**, *quantum*, **bethe**, **einstein** *semiclassical*
PDC	*geometrodynamics*, **bethe**, *semiclassical* **schwinger**, *perturbative*
HDC	**schwinger**, *electrodynamics*, **bethe** *semiclassical*, *quantum*
PV-DBOW	*physicists*, *spacetime*, *geometrodynamics* *tachyons*, **einstein**
moon	
CBOW	**earth**, **moons**, **pluto**, **sun**, **nebula**
SG	**earth**, **sun**, **mars**, **planet**, **aquarius**
PDC	**sun**, **moons**, *lunar*, *heavens*, **earth**
HDC	**earth**, **sun**, **mars**, **planet**, *heavens*
PV-DBOW	*lunar*, **moons**, *celestial*, **sun**, *ecliptic*

the representation learning, and (2) analyze why the joint model can perform better.

To show how syntagmatic and paradigmatic relations affect the learned representations, we present the 5 most similar words (by cosine similarity with 50-dimensional vectors) to a given target word under the PDC and HDC models, as well as three sub-models, *i.e.*, CBOW, SG, and PV-DBOW. The results are shown in table 3, where words in italic are those often co-occurred with the target word (*i.e.*, syntagmatic relations), while words in bold are whose substitutable to the target word (*i.e.*, paradigmatic relation).

Clearly, top words from CBOW and SG models are more under paradigmatic relations, while those from PV-DBOW model are more under syntagmatic relations, which is quite consistent with the model design. By modeling both relations, the top words from PDC and HDC models become more diverse, *i.e.*, more syntagmatic relations than CBOW and SG models, and more paradigmatic relations than PV-DBOW model. The results reveal that the word representations learned by PDC and HDC models are more balanced with respect to the two relations as compared with sub-models.

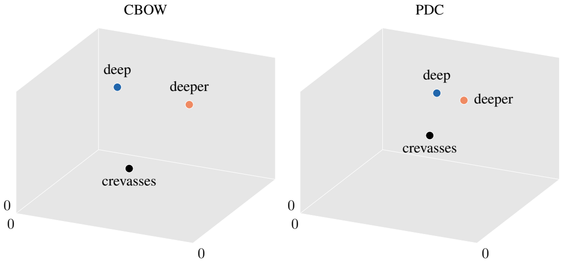

Figure 5: The 3-D embedding of learned word vectors of “deep”, “deeper” and “crevasses” under CBOW and PDC models.

The next question is why learning a joint model can work better on previous tasks? We first take one example from the word analogy task, which is the question “*big* is to *bigger* as *deep* is to __” with the correct answer as “deeper”. Our PDC model produce the right answer but the CBOW model fails with the answer “shallower”. We thus embedding the learned word vectors from the two models into a 3-D space to illustrate and analyze the reason.

As shown in Figure 5, we can see that by jointly modeling two relations, PDC model not only requires that “deep” to be close to “deeper” (in cosine similarity), but also requires that “deep” and “deeper” to be close to “crevasses”. The additional requirements further drag these three words closer as compared with those from the CBOW model, and this make our model outperform the CBOW model on this question. As for the word similarity tasks, we find that the word pairs are either syntagmatic (*e.g.*, “bank” and “money”) or paradigmatic (*e.g.*, “left” and “abandon”). It is, therefore, not surprising to see that a more balanced representation can achieve much better performance than a biased representation.

5 Conclusion

Existing work on word representations models either syntagmatic or paradigmatic relations. In this paper, we propose two novel distributional models for word representation, using both syntagmatic and paradigmatic relations via a joint training objective. The experimental results on both word analogy and word similarity tasks show that the proposed joint models can learn much better word representations than the state-of-the-art methods.

Several directions remain to be explored. In this paper, the syntagmatic and paradigmatic relations are equivalently important in both PDC and HDC models. An interesting question would then be whether and how we can add different weights for syntagmatic and paradigmatic relations. Besides, we may also try to learn the multi-prototype word representations for polysemous words based on our proposed models.

Acknowledgments

This work was funded by 973 Program of China under Grants No. 2014CB340401 and 2012CB316303, and the National Natural Science Foundation of China (NSFC) under Grants No. 61232010, 61433014, 61425016, 61472401 and 61203298. We thank Ronan Collobert, Eric H. Huang, Rémi Lebret, Jeffrey Pennington and Tomas Mikolov for their kindness in sharing codes and word vectors. We also thank the anonymous reviewers for their helpful comments.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan andGadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131, January.

J. R. Firth. 1957. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-59:1–32.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 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*, ACL ’12, pages 873–882, Stroudsburg, PA, USA. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196. JMLR Workshop and Conference Proceedings.

Rémi Lebret and Ronan Collobert. 2014. Word embeddings through hellinger pca. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–490. Association for Computational Linguistics.

Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, october.

Omer Levy and Yoav Goldberg. 2014a. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc., Montreal, Quebec, Canada.

Omer Levy and Yoav Goldberg, 2014b. *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, chapter Linguistic Regularities in Sparse and Explicit Word Representations, pages 171–180. Association for Computational Linguistics.

Kevin Lund, Curt Burgess, and Ruth Ann Atchley. 1995. Semantic and associative priming in a high-dimensional semantic space. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, pages 660–665.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop of ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language & Cognitive Processes*, 6(1):1–28.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1751–1758.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 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)*, pages 1059–1069, Doha, Qatar, October. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. 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*, pages 1532–1543.

Justin Picard. 1999. Finding content-bearing terms using term similarities. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, pages 241–244, Stroudsburg, PA, USA. Association for Computational Linguistics.

Douglas L. T. Rohde, Laura M. Gonnerman, and David C. Plaut. 2006. An improved model of semantic similarity based on lexical co-occurence. *Communications of the ACM*, 8:627–633.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8(10):627–633, October.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–54.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123, March.

Cyrus Shaoul and Chris Westbury. 2010. The westbury lab wikipedia corpus. *Edmonton, AB: University of Alberta*.

Richard Socher, Cliff C. Lin, Chris Manning, and Andrew Y. Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 129–136, New York, NY, USA. ACM.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 41–47, New York, NY, USA. ACM.