Phrase Embedding Learning Based on External and Internal Context with Compositionality Constraint

Minglei Li, Qin Lu, Dan Xiong, Yunfei Long*

Department of Computing,
The Hong Kong Polytechnic University, Hung Hom, Hong Kong

Abstract

Different methods are proposed to learn phrase embedding, which can be mainly divided into two strands. The first strand is based on the distributional hypothesis to treat a phrase as one non-divisible unit and to learn phrase embedding based on its external context similar to learn word embedding. However, distributional methods cannot make use of the information embedded in component words and they also face data spareness problem. The second strand is based on the principle of compositionality to infer phrase embedding based on the embedding of its component words. Compositional methods would give erroneous result if a phrase is non-compositional. In this paper, we propose a hybrid method by a linear combination of the distributional component and the compositional component with an individualized phrase compositionality constraint. The phrase compositionality is automatically computed based on the distributional embedding of the phrase and its component words. Evaluation on five phrase level semantic tasks and experiments show that our proposed method has overall best performance. Most importantly, our method is more robust as it is less sensitive to datasets.

Keywords: Phrase embedding, Compositionality, Distributional hypothesis, Composition model

*Corresponding author
Email address: yunfei.long@connect.polyu.hk (Yunfei Long)

1. Introduction

Phrases, as one kind of language units, play an important role in many NLP applications such as machine translation, web searching and sentiment analysis [1]. Generally speaking, phrases can be categorized as either **compositional** or **non-compositional**. For compositional phrases, such as *traffic light, swimming pool*, their semantics are composed from the semantics of its component words. We define component words as the **internal context** of a phrase. For non-compositional phrases, such as multiword expressions *couch potato* and *kick the bucket*, their semantics are generally not directly related to the semantics of their component words. According to [2], in a corpus with a collection of web pages, about 15% of word tokens belong to multiword expressions, 57% of sentences and 88% documents contain at least one multiword expression.

With the success of word embedding as a latent low dimensional vector [3] to represent words, embedding representation has been proposed for other areas, such as network embedding [4] and user embedding [5], etc. Different models are also proposed to learn phrase embedding. Phrase embedding uses two main approaches.

The first one is called the **distributional approach** which is developed based on the distributional hypothesis that words occurring in similar contexts tend to have similar meanings [6]. This kind of context is referred to as **external contexts**, which indicates the surrounding words of a phrase. We use the term **distributional embedding** to refer to embedding obtained by the distributional approach. Methods based on the distributional approach treat a phrase as one single unit and learn embeddings the same way as learning word embedding [7, 8, 9]. However, distributional embedding suffers from data sparseness problem. This is because distributional methods are based on the contexts of a target word. For words with lower frequency of occurrences, there are insufficient number of word-context pairs. Data sparseness problem is more serious at phrase level compared to that of word level. For phrases that are indeed compositional, the semantic information contained in component words are totally ignored. For example, both *traffic* and *light* are frequently used words and their embeddings can be very useful in forming the meaning of the phrase *traffic light*. But, non-compositional methods do not make use of such information.

The second approach, referred to as the **compositional approach**, is based on the principle of compositionality [10] that the meaning of an expression is composed from the meanings of its constituents and the internal structure. We use **compositional embedding** to refer to embeddings obtained by the compositional approach. This kind of methods compute phrase embedding from the embeddings of the component words based on some composition function [11, 12, 13, 14]. One problem with this approach is that the embedding learned for non-compositional phrases are incorrect, and thus this approach fails for non-compositional phrases. For example, the meaning of the phrase *monkey business* is not related to the meanings of *monkey* and *business*. Thus any composition function based on the embeddings of the component words will lead to erroneous results.

We argue that both the internal contexts and external contexts are useful for inferring phrase embedding. The usefulness of internal contexts depends on the compositionality of the phrases. If a phrase is compositional, both the internal contexts and external contexts should be used to take advantage of the all the information available for its representation. If a phrase is non-compositional, the representations of component words will not be useful and the phrase representation should be inferred from its external contexts only. The issue is that the choice of which approach to use is dependent on the proportion of compositional phrases in the dataset. This information, however, is not priori knowledge known to applications.

Based on the above analysis, we propose a hybrid model by a linear combination of both a distributional component and a compositional component with an individualized compositionality constraint. Compositionality is a value to indicate to what extent the semantics of a phrase can be inferred from that of its component words. The more compositional a phrase is, the larger is its compositionality value. For a non-compositional phrase, its compositionality should be low. Thus, in the hybrid model, its semantics should mainly be determined by its external contexts through the distributional component only. For a compositional phrase, its compositionality should be high. Both distributional component and compositional component can be used together. The hybrid model is designed to overcome the drawbacks of both distributional approach and compositional approach. The key for the hybrid model to work is how to learn an appropriate compositionality for each phrase. A constant value to all phrases obviously should not do the trick. In this work, we use two methods to learn the compositionality for each phrase using measures between distributional embeddings of a phrase and its component words.

To evaluate the performance of our proposed model, we applied our phrase embedding results in different down stream tasks using five datasets. Evaluations show that our model has a overall best performance. More importantly, our model is the most robust as it is less sensitive to datasets than the baseline methods.

The rest of the paper is organized as follows. Section 2 introduces related works. Section 3 presents our proposed hybrid model. Section 4 gives performance evaluation, and Section 5 concludes this paper.

2. Related Work

2.1. Embedding Representation

Representing objects in a latent space has a long history, such as Latent Semantic Analysis which represents a document as a latent vector [15]. Word embedding, as one kind of latent representation, represents a word as a low-dimensional and dense vector to encode semantic information. Methods for learning word embedding can either be count-based or prediction-based [16]. Count-based methods first build word-context as a statistic matrix where each entry in the matrix can be co-occurrence frequency, mutual information (MI), point-wise mutual information (PMI), and positive point-wise mutual information (PPMI), etc. Then the embedding representation of words can be obtained by matrix factorization. Different matrix factorization methods can be used, such as Singular Value Decomposition (SVD), QR factorization, etc [17]. Prediction-based methods use neural networks to predict a context word for a given target word or vice versa by maximizing the co-occurrence probability of the target word and its context. As proven in [18], one of the prediction-based methods, namely the Skip-Gram model which will be introduced in Section 3.1, is equivalent to a count-based method with each matrix entry as the PPMI by a constant shift. Inspired by the idea of modeling the relationships between a word and its context, other kinds of contexts are further studied, including context words under a specific syntactic dependency [19], context of words from different languages [20], context from a knowledge base [21], neighbor context in a semantic lexicon [22], substitute context [23], contrast context [24], path-based context [25], and morphological context [26, 22]. Further more, ensemble-based methods are also proposed to make use of multi-view contexts [27, 28, 29, 30]. For example, in [27], Rastogi et al. combine multi-view resources such as monolingual text from Wikipedia, word aligned bi-text, dependency relations, morphology and Frame relations through Generalized Canonical Correlation Analysis (GCCA). In [29], word definition as an intrinsic view and context as an extrinsic view are used. Given the current word, Chen et al. [29] maximizes the conditional probability of a context word and a definition word, which is similar to Wang et al. [21] that maximizes the conditional probability of a target word given a current word, where the current word is from a knowledge base. In [30], Speer et al. propose to combine word embedding from Skip-Gram, word embedding from matrix factorization, and word embedding from knowledge base ConceptNet [1] to obtain ensemble word embedding. The general conclusion is that more context information leads to better word embedding.

Similarly, embedding representation is also used in other areas, such as user and item representation based on user-item co-occurrence matrix in recommendation systems [31]. Inspired by prediction based methods, neural network based models are explored in other research areas to learn embedding representations, such as network embedding [4], and user embedding[5] etc.

2.2. Composition Model

One of the most important properties of a language is its compositionality. People communicate and parse complex information by combining single concepts through limited grammar rules. Semantic composition is studied in various disciplines such as psychology, linguistics, philosophy, neuroscience and computer science. Currently, however there is no consensus on how human combine simple concepts to obtain complex concepts [32]. In computer science, different mathematic composition models are

[1] http://conceptnet.io/

proposed to infer the representation of phrases based on the representation of words. For example, in [11], several basic composition models are proposed, including vector addition and vector multiplication of component words. Vector addition is the most widely used composition model because of its efficiency and performance. Different weighted addition versions are also proposed [33]. Baroni et al. [34] propose to represent an adjective as a matrix and a noun as a vector and use matrix-vector multiplication to obtain the representation of adjective-noun phrases. More complex composition models are proposed including recursive neural networks (RecNN) [35, 36, 37], recurrent neural networks (RNN) [38, 39], and convolutional neural networks (CNN) [40]. All of them are widely used deep learning models in natural language processing. Generally speaking, these complex composition models are based on the combination of some basic composition models, such as concatenation and matrix multiplication plus a non-linear transformation.

2.3. Compositionality Prediction

Compositionality indicates the extent of how the meaning of a phrase can be inferred from the meaning of its component words. Previous methods on compositionality prediction can be divided into two categories. The first type is based on the statistics between a phrase and its component words. It is known that for non-compositional phrases, their component words have stronger statistical associations. For example, Pedersen et al. propose to used t-score and PMI as the measure of compositionality [41]. However, for t-score and PMI, the generated statistic value range is hard to control and the obtained values can be very large. Statistical information is also used as features in supervised learning models. For example, Hashimoto et al. propose several syntactic features such as the word index, frequency and PMI of the phrase and component words [42]. Then the feature vector is multiplied by a weight vector to compute the compositionality value. However, for institutionalized compounds such as *traffic light, fresh air*, they also have strong statistical association, though they are compositional. The second type is based on semantic similarity of phrases and component words. Based on the contexts of a phrase, co-occurrence vector representation of the phrase and its component words can be obtained by extracting its external context words. Semantic similarity between a phrase and its component words can be computed by vector multiplication. For example, Baldwin et al. [43] propose to use semantic similarity between only one of the component words and phrase as compositionality. Reddy et al. [44] firstly compute similarities between a phrase and its two component words. Compositionality is then obtained from the two similarity values based on different functions such as addition and multiplication. Another semantic similarity based method first obtains the composed vector representation from the representations of component words based on some composition functions. Then compositionality is computed as the similarity between the composed vector and the co-occurrence vector [44]. Co-occurrence vectors are high-dimensional. Following the idea of composition, Salehi et al. [45] compute compositionality using cosine similarity between distributional phrase embedding and composed phrase embedding from component words based on composition functions. Observing that the semantic space of a phrase or a sentence is a subspace spanned by the word vectors of all component words, Gong et al. [46] propose to compute compositionality using cosine similarity between distributional phrase embedding and projected distributional phrase embedding on the subspace spanned by component words.

2.4. Phrase Embedding

Inspired by the success of word embedding, different models are proposed to learn embedding of phrases. As introduced in the Introduction part, there are mainly two kinds of approaches for learning phrase embedding. The first one is the distributional approach. For example, in [3, 8], Mikolov et al. and Yin et al. treat phrases as non-divisible units and learn phrase embedding the same way as learning word embedding. Not only considering the external context, Sun et al. argue that internal context is also useful and they use the same method to model internal contexts as that of external contexts [26]. The second one is the compositional approach which computes phrase embedding from component word embedding. Yu et al. [12] propose to obtain phrase representations by weighted sum of word vectors and weights are based on a list of lexical feature templates of phrase types. Zhao et al. [47] propose a tensor-based compositional model to learn phrase representations by vector-tensor-vector multiplication.

Huang et al. [48] propose to compute phrase embedding based on character and word embedding for Chinese through composition functions. However, compositionality is not considered in their work.

Different from the above methods, the work from [42] considers both external context and component words with a compositionality constraint,which is similar to our idea. However, the learning process in their work is task dependent and the proposed model only handles verb-noun phrases. In that work, predicate of a verb-noun phrase is represented as a matrix and the noun is represented as a vector. The compositional representation of verb-noun is obtained by matrix-vector multiplication. In addition, their compositionality prediction is based on manually defined features.

3. Proposed Framework

For a given phrase, our proposed model is shown in Figure 1, which consists of two parts: the distributional component based on the distributional hypothesis and the compositional component based on the principle of compositionality. The two parts are linearly combined with a fixed weight λ and a phrase specific compositionality weight t. λ is a hyper-parameter controlling the overall contribution of each component. The compositionality t is a value range from 0 to 1 where 0 indicates that the phrase is non-compositional and 1 indicates that the phrase is compositional. t is obtained by a phrase compositionality prediction model. The basic principle is that the more compositional a phrase is, the more contribution should be by the compositional component, namely, the more contribution from the component words. If a phrase is non-compositional, t should be close to zero and the semantic information comes only from the distributional component, namely the external context. We denote our proposed model as **D&C** (**D**istributional and **C**ompositional). D&C is similar to that of [42] which works on verb-noun phrases only. D&C extends [42] to handle general types of words.

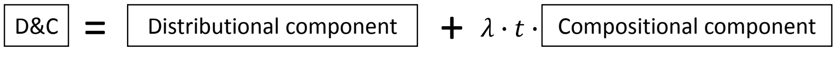

Figure 1: The framework of the proposed D&C model.

First we introduce some notations. Given a corpus S with a set of words $w \in$

V_W and their context $c \in V_C$ where V_W and V_C are word and context vocabularies. Note that the vocabularies of V_w and V_C are generally identical. The distinction is more for conceptual convenience only. The context of word w_i is defined as the words surrounding w_i in a window of size $2L$, namely w_{i-L}, $\cdots$, w_{i-1}, w_{i+1}, $\cdots$, w_{i+L}. Let $\#(w)$ denote the occurrence frequency of word w in S, and $\#(c)$ denote the occurrence frequency of context c. Let $\#(w, c)$ denote the frequency of a word-context pair (w, c). Furthermore, let V_M denote the set of given phrases where each phrase $m \in V_M$ consists of two words. t_m is used to denote the compositionality of the phrase m. The larger t_m is, the more compositional is the phrase m. Let D denote the set of (w, c) and (m, c) pairs. The objective is to learn a vector representation $\vec{w} \in \mathbb{R}^d$ for each $w \in V_W$, a vector representation $\vec{c} \in \mathbb{R}^d$ for each context $c \in V_C$, and a vector representation $\vec{m} \in \mathbb{R}^d$ for each $m \in V_M$. d is the vector dimension. The following subsections introduce the distributional component, the compositional component, the proposed hybrid model and compositionality prediction model in sequence.

3.1. Distributional Component

The distributional component makes use of the external contexts. The representation can be obtained from any word embedding learning model based on the distributional hypothesis. As introduced in Section 2.1, we can either use count-based methods or prediction-based methods for learning distributional phrase embedding. Compared to count-based methods, prediction-based methods do not need to perform a large matrix factorization, which is computation power demanding. In this work, we use a widely used prediction-based method, referred to as Skip-Gram with negative sampling (SGNS) model [7]. When applying SGNS to representation learning of phrases, $m_i \in V_M$ is treated as a single term and representation learning is carried out the same as word representation learning. To be specific, consider a phrase-context pair (m, c). Let $p(D = 1|m, c)$ be the probability that (m, c) comes from D and let $p(D = 0|m, c)$ be the probability that (m, c) does not comes from D. The basic assumption of SGNS is that the conditional probability of $p(D = 1|m, c)$ should be high if c is the context of phrase m in corpus D and $p(D = 0|m, c)$ should be high otherwise. $p(D = 1|m, c)$

is computed as:

$$p(D=1|m,c) = \sigma(\vec{m} \cdot \vec{c}) = \frac{1}{1+e^{-\vec{m}\cdot\vec{c}}}.$$

The basic idea behind this is that if phrase m and context c co-occur, their vectors should have close correlation, modeled by the element-wise multiplication $\vec{m} \cdot \vec{c}$. The objective of negative sampling is to maximize the conditional probability $p(D = 0|m, c_N) = \sigma(-\vec{m} \cdot \vec{c}_N)$ by randomly sampling negative context c_N of m from V_C. This can be translated to maximizing $\sigma(-\vec{m} \cdot \vec{c})$. So the objective for a single (m, c) pair is:

$$log(\sigma(\vec{m} \cdot \vec{c})) + k \cdot \mathbb{E}_{c_N \sim P_D}[log(\sigma(-\vec{m} \cdot \vec{c}_N))],$$

where k is the number of negative samples and P_D is the empirical unigram distribution $P_D(c) = \frac{\#(c)}{|D|}$. The final objective function for the whole phrase corpus is:

$$J_S = \sum_{m \in V_M} \sum_{c \in V_C} \#(m,c) \Big(log(\sigma(\vec{m} \cdot \vec{c})) + k \cdot \mathbb{E}_{c_N \sim P_D}[log(\sigma(-\vec{m} \cdot \vec{c}_N))] \Big). \tag{1}$$

Note that for compositional phrases, the SGNS component only takes information from external contexts. Internal contexts of component words are not directly taken into consideration.

3.2. Compositional Component

In the compositional component, the representation of a phrase is computed from the representations of its component words. Without loss of generality, we assume that phrases only have two component words, similar to discussions used in previous studies. Given a phrase m with two component words w_m^1 and w_m^2 and their embedding representations $\vec{w}_m^1$ and $\vec{w}_m^2$, the representation of m, denoted by $\vec{m}$, can be computed by a composition function f:

$$\vec{m} = f(\vec{w}_m^1, \vec{w}_m^2). \tag{2}$$

Different composition models are proposed for f in [49]. The weighted addition composition model with weights α and β is defined as a linear composition:

$$\vec{m} = \alpha \vec{w}_m^1 + \beta \vec{w}_m^2. \tag{3}$$

The multiplication composition model is defined by:

$$\vec{m} = \vec{w}_m^1 \cdot \vec{w}_m^2. \tag{4}$$

Note that in the above formulas, word embeddings are obtained in advance by any word embedding learning model. Compared to SGNS, the compositional component can make use of component words information. However, this model can produce erroneous representation for non-compositional phrases. For example, for the phrase *couch potato*, its meaning cannot be composed from its component words *couch* and *potato*.

3.3. The Hybrid Model

The distributional model using SGNS can suffer from data sparseness problem for phrases and cannot make use of component words information. The compositional model alone does not make full use of external context and is not appropriate for non-compositional phrases. Based on the distributional component and compositional component, our proposed hybrid model can modeled as:

$$\begin{aligned} J_S = \sum_{m \in V_M} \sum_{c \in V_C} \#(m,c) \Big(& log\sigma(\vec{m} \cdot \vec{c}) \quad + k \cdot \mathbb{E}_{c_N \sim P_D}[log(\sigma(-\vec{m} \cdot \vec{c}_N))] \\ & + \lambda t_m log\sigma\left(\vec{m} \cdot f(\vec{w}_m^1, \vec{w}_m^2)\right) \Big). \end{aligned} \tag{5}$$

In **Formula 5**, the first two parts forms the SGNS model and serve as the distributional component. The third part is the compositional component with a constant weight λ to balance the overall contributions of the two components. $f(\vec{w}_m^1, \vec{w}_m^2)$ can be any compositional model defined by **Formula 2**. $\sigma\left(\vec{m} \cdot f(\vec{w}_m^1, \vec{w}_m^2)\right)$ defines the correlation between the learned phrase embedding $\vec{m}$ and the composed phrase embedding. The more they are correlated, the larger contribution the third part is to J_S. t_m is the compositionality of m, which will be introduced in detail later.

Theoretically speaking, **Formula 5** has the following properties:

1. If the compositionality t_m is low (m being more non-compositional), the weight of the correlation between the phrase representation $\vec{m}$ and the composed representation $f(\vec{w}_m^1, \vec{w}_m^2)$ from its component words should be low. It means $\vec{m}$ is learned mainly based on SGNS, namely its external contexts.

2. If the compositionality t_m is high (m being more compositional), the weight of the correlation between $\vec{m}$ and $f(\vec{w}_m^1, \vec{w}_m^2)$ should be high and the objective function will force $\vec{m}$ to be similar to the composed $f(\vec{w}_m^1, \vec{w}_m^2)$. It means $\vec{m}$ should consider both the external contexts and component words.

By setting λ to zero, the model degrades to SGNS. By setting t_m to a constant, the model changes to a fix-weighted model.

3.4. Compositionality Prediction

One of the most important elements of D&C is the compositionality value t. The compositionality prediction model aims to predict the compositionality of a phrase. Phrase compositionality has the property of continuum [44]. For example, the compositionality of phrase *bus driver* is 1.0, which means this phrase is compositional and the meaning of it can be composed from the component words *bus* and *driver*. The compositionality of phrase *coach potato* is 0, which means this phrase is non-compositional and the meaning of it cannot be inferred from the component words *coach* and *potato*. The compositionality of the phrase *silver screen* is 0.6, which indicates that its semantics cannot be totally obtained from the component words because the first word *silver* loses its original meaning in the phrase while the second word *screen* can reflect the phrase' meaning. In this section, we introduce two models for predicting individual compositionality of phrases.

The first model is from [45], which computes the compositionality of a phrase based on the consine similarity between the distributional embedding and the compositional embedding of the phrase defined as:

$$t_m = cosine\left(\vec{m}, \vec{w}_m^1 + \vec{w}_m^2\right), \tag{6}$$

where $\vec{m}$, $\vec{w}_m^1$ and $\vec{w}_m^2$ are obtained by SGNS in advance. Formula 9 means that the more similar between $\vec{m}$ and $\vec{w}_m^1 + \vec{w}_m^2$, the more compositional the phrase is. We label this compositionality prediction model as **C1**.

The second model is inspired by the work from [46] which is based on the geometry of word embedding. They find that the semantic space of larger text units (such as phrases and sentences) is spanned by the subspace of the consisting word vectors and the subspace can be obtained through dimension reduction such as Principle Component Analysis (PCA). Inspired by this, we propose to compute phrase compositionality by computing the cosine similarity between the distributional embedding and the projected vector on the subspace spanned by the component word embeddings. The process is shown in Figure 2. Given a phrase m consisting of two words w_m^1 and w_m^2, $\vec{m}$ is the distributional phrase embedding and $\vec{v}_m^1$ and $\vec{v}_m^2$ are the distributional component word embedding, obtained by distributional methods. $\vec{m}_p$ is the projected vector of $\vec{m}$ on the space spanned by $\vec{v}_m^1$ and $\vec{v}_m^2$. Let $A = [\vec{v}_m^{1T}, \vec{v}_m^{2T}]$. $\vec{m}_p$ is computed as:

$$\vec{m}_p = A(A^T A)^{-1} A^T \vec{m}. \tag{7}$$

The compositionality is computed as:

$$t_m = cosine(\vec{m}, \vec{m}_p). \tag{8}$$

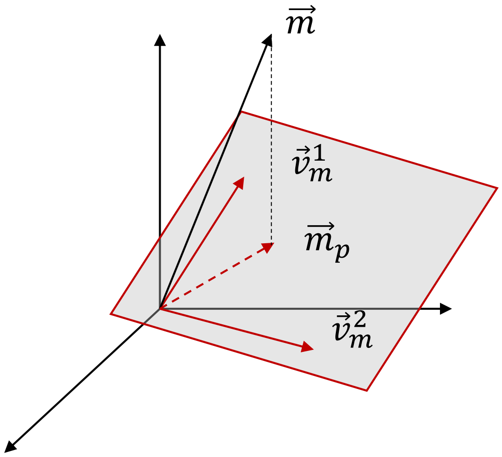

Figure 2: The C2 model for compositionality prediction.

270 We label this compositionality prediction model as **C2**. Compared to C1, C2 assumes that if a phrase is compositional, its phrase representation is the subspace spanned by its component words. The more the distributional embedding is close to the subspace, the more compositional of the phrase. If the distributional embedding is perpendicular to the subspace, the phrase is non-compositional.

Generally speaking, the two compositionality models C1 and C2 can be easily extended to include phrases with length K where K is the number of component words. For a phrase m consisting of K words, C1 can be extended to compute the compositionality by:

$$t_m = cosine\left(\vec{m}, \sum_{i=1}^{K} \vec{w}_m^i\right). \tag{9}$$

For C2, the distributional phrase embedding $\vec{m}$ is projected to the subspace spanned by component word vectors to obtain vector $\vec{m}_p$. Compositionality can then be computed as:

$$t_m = cosine(\vec{m}, \vec{m}_p). \tag{10}$$

After obtaining the compositionality value, the proposed model can be used for longer phrases just the same as as that of bigrams.

Theoretically speaking, any phrase compositionality model can be used in our proposed framework. Note that the compositionality values of phrases are computed based on the distributional embedding before training the model.

Details of the training procedure of the hybrid model is shown in **Algorithm 1**. Our model can be trained through stochastic gradient descent (SGD) by maximizing Formula 5 suggested by [3]. The gradient can be directly calculated for each training sample. Both the word embeddings and phrase embeddings are randomly initialized as what is used by Mikolov et al [3].

4. Experiment

In this section, we evaluate the representations by the proposed phrase embedding learning model on five different phrase level semantic tasks including both English and Chinese. For all experiments based on English text, Wikipedia August 2016 dump[2] is used as our training corpus. In pre-processing, pure digits and punctuations are removed and all English words are converted to lowercase. The final corpus consists of about 3.2 billion words. During training, only words that occur more than 100

[2] https://dumps.wikimedia.org/enwiki/latest/

Algorithm 1 The procedure of training phrase embedding based on the hybrid model.

```
• Input:
$M = [m_1, \cdots, m_n]$: phrase list of size n.
$S$: text corpus.
$\lambda$: overall weight hyper-parameter.
$d$: embedding dimension.
$win$: window size.
• Output:
$\vec{w}, \vec{m}$: the learned word embeddings and phrase embeddings.
• Procedure:
1. Extract word-context and phrase context pairs from $S$ based on $win$.
2. Using SGNS model to train distributional word and phrase embeddings, $\vec{w}, \vec{m}$.
3. Compute phrase compositionality values $t_m$ using C1 and C2 based on $\vec{w}, \vec{m}$.
4. Using D&C to obtain the final word and phrase embeddings $\vec{w}, \vec{m}$ based on $t_m$.
5. Return $\vec{w}, \vec{m}$.
```

times are kept, resulting in a vocabulary of 204,981 words. The list of phrases used in the evaluation are from 5 sources: (1) the set of 2,180 phrases in the Noun-Modifier Composition dataset [50], (2) the DISCo set of 349 phrases for the 2011 shared task in Distributional Semantics and Compositionality [51], (3) the set of 8,105 phrases from the SemEval 2013 Task 5A [52], (4) the set of 1,042 phrases from [53], and (5) the set of 56,850 phrases from [8]. The consolidated phrase list has a total of 60,315 phrases after removal of duplicates. For experiments using Chinese data, the training corpus is from Baidu Baike[3] with 1.8 billion tokens after performing word segmentation using the HIT LTP tool.[4] The Chinese phase list is from [14]. The distributional embeddings of the phrases and words are first learned based on SGNS for computing compositionality using C1 and C2 models. Then the compositionality values are used in our D&C model to obtain the final phrase embeddings.

[3] http://www.nlpcn.org/resource/list/2

[4] www.ltp-cloud.com/

4.1. Evaluation Tasks

The proposed model is evaluated on five tasks. The first task is from the **SemEval 2013 Task 5**. The dataset for this task, denoted as **SemEval**, is prepared to judge whether a given bigram-unigram pair is semantically related or not [52]. For example, the bigram *newborn infant* is semantically related to the unigram *neonate*. So, the gold answer for this pair is *(newborn infant, neonate, 1), where the label 1 indicates their relatedness*. On the other hand, the bigram *stable condition* is not related to the unigram *interview*, So, in the gold answer, the entry is *(stable condition, interview, 0)*. The officially released data in SemEval contains 7,814 test samples and 11,722 training samples.[5] Since only 15,973 samples are contained in Wikipedia, they are used in our evaluation[6]. SemEval 2013 Task 5 is a binary classification problem. The cosine similarity between a learned bigram embedding and a unigram embedding is used as the feature. Support Vector Machine (SVM) is used to as the classifier to perform 5-fold cross-validation classification. Accuracy, precision, recall and F-score are used as the evaluation metrics.

The second task is called **Phrase Similarity** [11]. Since this is an English dataset, we denote it is as **PS-En**. This task provides a phrase pair similarity dataset with 324 samples[7] constructed using manually rated scores from 1 to 7 with 7 being the most similar. For example, the phrase pair *(hot weather, cold air)* has a similarity score 2.22. The dataset contains three types of phrases: adjective-nouns, noun-nouns, and verb-objects with 108 samples for each type. All 324 samples are used in the evaluation. Cosine similarity is used to compare two phrase vectors and Spearmans ρ correlation coefficient between estimated similarities and gold similarities is used as the evaluation metric.

The third task is on phrase similarity for Chinese, denoted as **PS-Ch**. The evalua-

[5] https://www.cs.york.ac.uk/semeval-2013/task5.html.

[6] Theoretically speaking, compositional models are not limited that one phrase has to occur in the training corpus because it requires only the component words occurring in the corpus. It is an advantage of compositional models over distributional models. Removing the phrase not occurred in the corpus is beneficial for distributional models. However, this is not our focus in this paper and our model is not limited by this.

[7] http://homepages.inf.ed.ac.uk/mlap/index.php?page=resources

tion dataset is from Wang et al. [14].[8] Similarity annotation ranges from 1 to 6 with 6 being most similar. The final evaluation size is 240. Note that all the phrases are compositional. The evaluation method is the same as that for PS-En.

The fourth task is labeled as **Turney-5** [50]. The dataset in this task is a 7-choice Noun-Modifier Question dataset built from WordNet with 2,180 question groups. For example, in the sample *(small letter, lowercase, small, letter, little, missive, ploughman, debt)*, the first bigram *small leter* is the question and the latter 7 unigrams are the candidate answers. The task is to select the most similar unigram as the answer, which should be *lowercase* in this example. To remove the bias towards component words by following Yu's suggestion [12], both two component words are removed to construct a 5 choice single word questions to form our evaluation dataset, denoted as **T-5**. Again, by removing samples that are not contained in the Wikipedia training corpus, the final evaluation data contains 669 questions. The cosine similarity is used to measure the semantic closeness of a bigram phrase and the unigrams. The one with the highest similarity score is chosen as the answer. Accuracy is used as the evaluation metric.

The fifth task is to predict the sentiment score of phrases proposed by this work. The phrase list is extracted from the Stanford Sentiment Treebank (SST) [37]. In SST, every sentence is syntactically parsed and every node is annotated with a sentiment score from 0 to 1 through crowdsourcing, where 0 indicates the most negative and 1 indicates the most positive. We extract the noun-noun and adjective-noun phrases in the parsed trees and the overlapping set of phrases in SST and our phrase list is 772. The obtained phrase embedding is treated as latent feature representation and the target is to predict the sentiment score, which is a regression problem. This dataset is denoted as **SST**. The evaluation metrics include Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and the Kendall rank correlation coefficient (τ). For RMSE and MAE, the smaller of the value, the better of the performance, and vice versa for τ. The Ridge Regression model is used to predict the sentiment score of the phrases from the phrase embedding.

Note that all the datasets contain only bigram phrases. As explained in **Section 3.4**,

[8] https://github.com/wangshaonan/Phrase-representation

the proposed model is not limited to bigrams. However, multi-word phrase datasets with longer length are not available. Similar to other works, performance evaluations here are all based on bigram datasets used in other reported works.

4.2. Baselines and Experiment Settings

The proposed hybrid model is compared with the following baselines:

1. **SGNS**: the original word representation learning model that takes a phrase as a non-divisible unit [3, 8];
2. **SEING**: a modified SGNS model by treating component words as the context of a phrase and perform the same constraint on component words as the external contexts [26]. This will force the phrase vector to be similar to both the vectors of its component words regardless of compositionality of the phrase;
3. **Comp-Add**: the addition composition model to use the average of the vectors of the component words to obtain the vector of a phrase.
4. **Comp-Mul**: the multiplication composition model to use the multiplication of the two components vectors to obtain the vector of a phrase.
5. **Comp-W1**: a composition model to use the vector of the first component word directly as the vector of a phrase;
6. **Comp-W2**: a composition model to use the vector of the second component word directly as the vector of a phrase;

The proposed D&C model has three settings for compositionality t_m. The first one directly sets t_m as a constant, $t_m = 1$, denoted as **D&C-***C*. This means the compositionality of all phrases is set fixed as an identical and fixed number. The second one uses automatically computed t_m by model C1, denoted as **D&C-***C1*. The third one uses automatically obtained t_m by model C2, denoted as **D&C-***C2*. Both D&C-C1 and D&C-C2 estimate compositionality for each phrase individually.

The size of the context window for all the models is set to 10, negative samples size is 5, and the embedding dimension is 300. For λ, we evaluate different values on SemEval based on 5-fold cross validation and select the best value 8. For other datasets, we use the same value for λ. For the composition model in Formula 2, we empirically evaluate several combinations such as the addition model with α and β as 1, or the multiplication model. Experiments show that the addition model achieves the best result. So only the results using the addition composition model are reported here. To obtain compositionality t_m, the representations of a phrase is first trained using SGNS and its compositionality is computed based on model C1 and C2, respectively.

Model	SemEval				PS-En	PS-Ch	T-5	SST		
	(2.5%)				(2.5%)	(0%)	(10%)	(30%)		
	Acc	**Pre**	**Rec**	**F**	ρ	ρ	**Acc**	**rmse**	**mae**	τ
SGNS	.629	.728	.412	.526	.155	.075	.535	.094	.063	.218
SEING	.586	.562	.773	.651	.056	.531	.576	.089	.061	.269
Comp-Add	**.795**	**.826**	**.748**	**.785**	**.622**	**.784**	.603	.090	.066	.283
Comp-Mul	.506	.506	.483	.494	410	.647	.227	.098	.063	.219
Comp-W1	.737	.771	.672	.718	.450	.648	.499	.092	.065	.211
Comp-W2	.759	.796	.697	.743	.500	.682	.463	.100	.071	.113
D&C-C	**.779**	.808	**.731**	**.767**	.595	**.786**	**.683**	.089	**.061**	**.301**
D&C-*C1*	.764	.794	.711	.750	.580	.776	**.681**	**.087**	**.060**	**.310**
D&C-*C2*	.776	**.**841	.681	.753	**.623**	.765	.677	**.088**	.061	.293

Table 1: Performance of different phrase representation learning models. The top two performers are in bold and the best performer is also underlined.

4.3. Performance Evaluation and Analysis

The evaluation result on the five datasets under different evaluation metrics is shown in Table 1. In this table, the first two models are distributional methods, the middle four models are compositional methods and the last three models are three variants of our proposed model. The percentage after each dataset name is the proportion of non-compositional phrases in that dataset. The percentage is obtained by randomly sample 30 phrases in each data set and then manually verify their compositionality. We can see that different datasets do have different proportions of non-compositional phrases and this should have effects on the performance of different methods.

4.3.1. General Analysis

Comparison between distributional methods and compositional methods shows that compositional methods achieve much better result than distributional methods. For example, on SemEval, Comp-Add achieves a relative improvement of 49.2% under F-score compared to SGNS. In other words, the semantics of phrase expressions are not fully recognized by using only external context. Treating phrases as a non-divisible units obviously loses some semantic information carried by the component words. This also indicates that in a real application, compositional models are a better choice compared to a distributional approach for phrase embedding learning. Comparing between distributional models, SEING performs better than SGNS on SemEval, PS-Ch, T-5 and SST. But, SEING performs worse than SGNS on PS-En. Further analysis of SEING on PS-En indicates that the cosine similarities of many phrase pairs in PS are negative. The average frequency of PS-Ch is only 461, much smaller than the average frequency 1,297 of PS-En. That is why SGNS performs much worse on PS-Ch. Among the four baseline compositional methods, Comp-Add performs much better than other compositional methods. Comp-Mul performs the worst. This means that element-wise multiplication can introduce more noise than information. Comp-W1 and Comp-W2 have similar performance with Comp-W2 performing slightly better on SemEval, PS-En and PS-Ch and Comp-W1 performing better on T-5 and SST. Among all the models, Comp-Add performs the best on the SemEval dataset while our proposed model D&C performs the best on PS-En, PS-Ch, T-5 and SST. Specifically, on T-5, the best performer D&C achieves a relative improvement of 13.3% over Comp-Add. This indicates the effectiveness of our proposed model. Among the three variants of D&C, no one is overall best. D&C-C performs the best on SemEval and T-5, while D&C-C2 performs the best on PS-En, D&C-C1 performs the best on SST under rmse, mae and τ. Overall, our proposed model achieves the most robust result since D&C is always the top two performer on all datasets and in fact top performer in four out of five datasets.

Further analysis indicates that the performances of different models are dataset dependent, especially dependent on the proportion of non-compositional phrases. As shown in Table 1, the proportions of non-compositional phrases are 2.5%, 2.5%, 0%, 10%, and 30% in SemEval, PS-En, PS-Ch, T-5 and SST, respectively. Because compositional models are more suitable for compositional phrases, Comp-Add performs much better than SGNS on SemEval. However, the gap decreases on T-5 between SGNS and Comp-Add as the proportion of non-compositional phrases increases. Performance of Comp-Add indicates that the combined use of the vectors of two component words is more comprehensive than using external contexts for compositional phrases. On T-5 and SST datasets, the proportions of the non-compositional phrases are larger than in the other two sets. So, there are more phrases which would not work using compositional methods. That is why the performance of SGNS increases and D&C outperforms Comp-Add.

4.3.2. Compositionality Analysis

To further explore the effects of compositionality on different methods, the proportion of non-compositional phrases are further analyzed based on the SemEval semantic relation task. 20 non-compositional phrases are manually selected from Farahmand's list which has 1,042 phrases manually annotated with compositionality values [53]. Each phrase is annotated by four annotators with 1 indicating non-compositional and 0 as compositional. Based on the 20 phrases, 20 positive (semantically related) bigram-unigram pairs and 20 negative (not semantically related) bigram-unigram pairs are constructed to form a balanced non-compositional sample set for the SemEval task, denoted as **N-Sem**. 60 samples from the original SemEval dataset are also taken to form a compositional sample set, denoted as **C-Sem**. In the evaluation, the non-compositional phrases from N-Sem are added to C-Sem to increase the proportion of non-compositional phrases until all the non-compositional phrases are used up (total of 100 samples). Then the compositional portion is reduced so that the non-compositional proportion reaches about 70% of the total set (57 samples). The two distributional models, SGNS and SEING, are selected for evaluation. Since Comp-Add performs much better than the other three compositional models, only Comp-Add is included for comparison. For comparison, we introduce another variant of D&C, **D&C-M**, which uses manually annotated compositionality as t_m, which is obtained as follows. We first obtain the sum the four annotation values as a and convert a by $t_m = (4 - a)/4$ to obtain t_m as the gold compositionality value. t_m is in the range of [0,1] and is consistent with our definition of compositionality (namely 1 indicates compositional, 0 indicates non-compositional). F-score is used as the evaluation metric. Because of the limited data size, each model is run 10 times and the average is used.

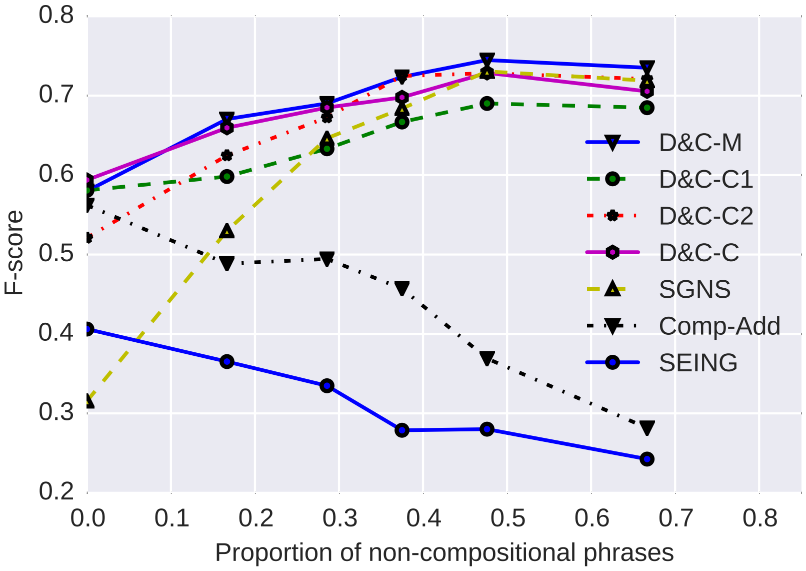

Figure 3: Performance of different models when increasing the proportion of non-compositional phrases.

The result is shown in **Figure 3**. This figure shows that when the proportion of non-compositional phrases is small, Comp-Add performs much better than SGNS, consistent with the result in **Table 1**. As the non-compositional portion increases, the performance of Comp-Add degrades gradually whereas in contrast, the performance of SGNS increases gradually. This indicates that external context is indeed useful for non-compositional phrases and the compositional model is ill-suited for non-compositional phrases. The performance of SEING indicates that the constraint to force a phrase's vector to be similar to both of its components can actually bring adverse effect for non-compositional phrases.

Note that even though D&C-C assumes all phrases are compositional so that t_m

is set to constant 1, the performance of D&-C in **Figure 3** does not decrease like Comp-Add and SEING. Further analysis reveals that the average frequency of the non-compositional phrases is 1,914.7 while the average frequency of compositional phrases is only 328.4. One possible reason is that higher frequency of non-compositional phrases generally leads to better distributional embeddings compared to that of the the compositional phrases. Even though the compositional part can introduce noise, performance can still improve when the benefits from the distributional part is larger than the noise introduced by the compositional part. As the noise accumulates, the performance of D&C-C begins to decrease when the proportion achieves about 0.5.

Over the whole spectrum, D&C gives a much more stable performance and is the overall top performer in all the automatic methods. D&C-M, which uses manually annotated compositionality, gives the best performance. The better performance of D&C-M over D&C-*C1* and S&C-*C2* indicates that there is still room for improvement in compositionality estimation. To validate this, a selected group of phrases are evaluated from Farahmand's list [53]. The overlapping of the phrase list with our phrase list is 408. We use the 408 phrases to evaluate the performance of the two compositionality prediction models. The estimated compositionality values by model D&C-C1 and D&C-C2 are compared with the gold compositionality by calculating Spearman's ρ correlation between the golden compositionality and the estimated compositionality. The result shows that ρ only achieves 0.227 and 0.200 for compositionality prediction model C1 and C2 respectively, which means the current method for compositionality estimation still has much room for improvement. Even though inaccurate t_m means that the use of compositional component may be less accurate and may introduce noise, but it still brings benefits compared to the baselines. The improvement of the hybrid model results from the combination of distributional component and compositional component so that the model makes use of more information (both external context and component words).

4.3.3. Hyper-parameter Analysis

To investigate the effects of the hyper-parameter λ, Figure 4 shows the performance of D&C-C on the four datasets when varying λ. The evaluation metrics are F-score,

ρ, accuracy, τ for SemVal, PS-En, T-5 and SST respectively. The result indicates that D&C-C achieves the best performance when λ equals about 8.

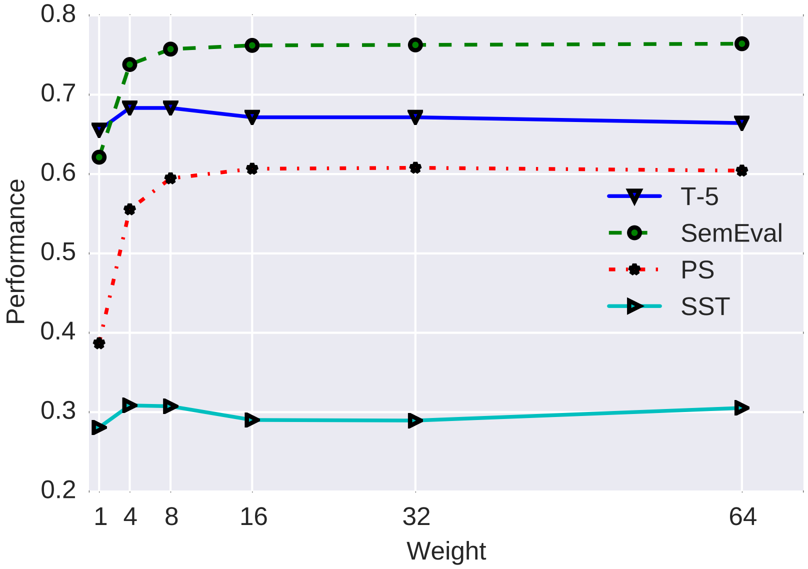

Figure 4: Performance of D&C-C with different λ values.

4.3.4. Case Study

To examine the performance of each model more closely, we select four phrases to extract the top 5 most similar words by different models. The phrases are selected based on the occurrence frequency in the Wikipedia corpus and the compositionality values. They are annotated in [44] with compositionality values from 0 to 5 with 0 indicating the most non-compositional and 5 indicating the most compositional. The statistics of the four phrases are shown in Table 2. **Frequency** is the occurrence frequency in our Wikipedia corpus and **Compositionality** is the annotated value by [44]. As shown in Table 2, the first phrase, *swimming pool* is highly compositional with high frequency. The second phrase, *game plan*, is highly compositional but low frequency. The third phrase, *melting pot*, is low in compositionality and high in frequency. The last phrase, *rat run*, is low in compositionality and also low in frequency. We list the top 5 most similar words/phrases based on the cosine similarity between the phrase embedding and the word/phrase embedding. The result of selected words/phrases based on different models is shown in Table 3. Overall, Comp-Mul gives the worst unreasonable results. Comp-W1 and Comp-W2 give results similar to the first component word and the second component word, respectively. So we can put them aside in our discussions.

	swimming pool	game plan	melting pot	rat run
Frequency	17794	116	6119	4
Compositionality	4.87	3.83	0.54	0.79

Table 2: Statistics of the selected example phrases.

Firstly, for the high compositionality and high frequency phrase, *swimming pool*, all the models give reasonable results that are semantically similar to *swimming pool*. Secondly, for the high compositionality and low frequency phrase, *game plan*, the results from SGNS are not reasonable. For example, all the given phrases of *fool up, run book, make book luck out*, and *times sign* are not closely related to *game plan*. This validates the claim that SGNS can not perform well when the occurrence frequency is low. SEING gives reasonable results because it constrains a phrase to be semantically related to its component words. Comp-Add also gives semantically related words/phrases and most of the them are related to the word of game. The three variants of the proposed D&C model all give similar results, including the most reasonable phrase *game plans*. Thirdly, for the low compositionality and high frequency phrase, *melting pot*, SGNS gives reasonable and similar results, which are all related to politics. On the contrary, both SEING and Comp-Add gives unreasonable cases, which are all related to either component word *pot* or *melting* but not related to *melting pot*. Again, the three variants of our proposed D&C model give reasonable results, which are all related to *melting pot*. Fourthly, for the low compositionality and low frequency phrase, *rat run*, all the results given by all the models are unreasonable. This is because all distributional models fail under low frequency whereas compositional models fail because of non-compositionality. However, our proposed models still give a semantically related phrase *rat running*. In conclusion, this case study validates that distributional models will fail when the occurrence frequency of a phrase is low and compositional models will fail when a phrase is non-compositional. Our proposed model gives the most robust answers. However, none of the models perform well when a phrase is non-compositional with low occurrence frequency.

To conclude, the distributional model performs better than compositional model when the proportion of non-compositional phrase is large and the compositional model performs better than distributional model when the proportion of non-compositional phrase is small. However, in practice, we do not have prior knowledge on the proportion of non-compositional phrases. This is why our proposed method has advantage over both models individually as our method learns compositionality for individual phrases. Thus, D&C is less sensitive to datasets, especially the proportion of non-compositional phrases. Becasue of this, it gives an overall better and more robust performance no mater what proportion of non-compositional phrases are. In addition, the fact that D&C-M gives the best performance highlights the need for a more accurate estimation of compositionality.

Models	**swimming pool**	**game plan**	**melting pot**	**rat run**
SGNS	*swimming pools, squash courts, tennis courts, climbing wall, basketball courts*	*fool up, run book, make book, luck out, times sign*	*diasporic, middle eastern, mestizaje, caribbeans, ethnicities*	*holds true, faster computers, improve understanding, fuzzy set, molecular entity*
SEING	*swimming pools, pool hall, pool halls, tennis courts, wading pool*	*strategy game, arcade game, saved game, strategy games, game board*	*cooking pot, pot luck, pot roast, coffee pot, pot shots*	*hog line, hoosier state, blade roast, w byrd, running dog*

Comp-Add	*swimming, swimming pool, squash courts, pools, swimming pools*	*game, the game, plans, a game, strategy game*	*pot, melt, cooking pot, saucepan, boiling*	*rat, brown rat, roof rat, black rat, giant kangaroo*
Comp-Mul	*weberian, individuation, apparatuses, cope, internalization*	*negatives, barb, stag, andersons, smallville*	*pot, cooking pot, talgai, pocket knife, pinfold*	*controversially, sion, furthered, controversy, tahiti*
Comp-W1	*swimming pool, aquatics, swim, synchronized swimming, squash courts*	*the game, games, card game, video game, wiiware*	*melt, melts, melted, melting point, eutectic*	*rats, rodent, rattus, mole rat, muridae*
Comp-W2	*pools, swimming pool, squash courts, wading pool, swimming pools*	*plans, planning, master plan, planned, proposal*	*pots, cooking pot, saucepan, pourri, ladle*	*running, runs, ran, run in, run on*

D&C-C	*swimming pools, tennis courts, squash courts, basketball courts, fitness center*	*game plans, a game, saved game, end game, waiting game*	*diasporic, mestizaje, caribbeans, middle eastern, folk culture*	*rat running, rat through, rat on, rat trap, young rat*
D&C-C1	*swimming pools, squash courts, tennis courts, basketball courts, fitness center*	*game plans, end game, waiting game, saved game, game board*	*diasporic, mestizaje, caribbeans, diasporas, folk culture*	*rat running, rat through, rat on, rat race, rat trap*
D&C-C2	*swimming pools, tennis courts, squash courts, basketball courts, indoor pool*	*game plans, the game, a game, strategy game, board game*	*diasporic, mestizaje, caribbeans, ethnicities, diasporas*	*rat running, rat through, rat on, rat, rat trap*

Table 3: The top 5 similar words of four kinds of phrases.

5. Conclusion and Future Work

In this paper, a hybrid model, D&C, is proposed to learn the representation of phrases from both their external contexts and internal contexts through a weighted linear combination with a phrase specific constraint. Instead of a simple combination of the two kinds of information, the individualized compositionality measures from lexical semantics are used to serve as the constraint. Evaluations on five phrase semantic analysis tasks show that the proposed hybrid model performs better than other models in four out of five datasets. Our model is the most robust on both compositional and non-compositional phrases without any knowledge of the dataset in terms of proportion of non-compositional phrases. This also indicates that incorporating more semantic information properly brings benefits for representation learning.

Even though the model gives a theoretically sound solution, the compositionality estimation method still has room for improvement. Firstly, more study on appropriate compositionality estimation model can be investigated as future work. Secondly, acquisition of longer phrase datasets should be conducted to see how different models work on longer phrases and which method can deal with data sparseness issue more effectively.

References

[1] A. Moreno-Ortiz, C. Prez-Hernndez, M. . Del-Olmo, Managing Multiword Expressions in a Lexicon-Based Sentiment Analysis System for Spanish, in: Proceedings of the 9th Workshop on Multiword Expressions, MWE@NAACL-HLT, Vol. 13, Atlanta, Georgia, USA, 2013, pp. 1–10.

[2] N. Schneider, S. Onuffer, N. Kazour, E. Danchik, M. T. Mordowanec, H. Conrad, N. A. Smith, Comprehensive Annotation of Multiword Expressions in a Social Web Corpus, in: Proceedings of the Ninth International Conference on Language Resources and Evaluation, Reykjavik, Iceland, 2014, pp. 455–461.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed Representations of Words and Phrases and Their Compositionality, in: Proceedings of Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, United States, 2013, pp. 3111–3119.

[4] A. Grover, J. Leskovec, node2vec: Scalable Feature Learning for Networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 855–864. doi:10.1145/2939672.2939754.

[5] Y. Yu, X. Wan, X. Zhou, User Embedding for Scholarly Microblog Recommendation, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 449–453. doi:10.18653/v1/P16-2073.

[6] Z. S. Harris, Distributional structure, Word 10 (2-3) (1954) 146–162.

[7] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, CoRR abs/1301.3781.

[8] W. Yin, H. Schtze, An Exploration of Embeddings for Generalized Phrases, in: Proceedings of the ACL, Student Research Workshop, 2014, pp. 41–47.

[9] W. Yin, H. Schtze, Discriminative Phrase Embedding for Paraphrase Identification, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 1368–1373. doi:10.3115/v1/N15-1154.

[10] G. Frege, The Foundations of Arithmetic, 2nd Edition, Vol. Trans. J. L. Austin, Northwestern University Press, Evanston, Illinois, 1884.

[11] J. Mitchell, M. Lapata, Composition in Distributional Models of Semantics, Cognitive Science 34 (8) (2010) 1388–1429. doi:10.1111/j.1551-6709.2010.01106.x.

[12] M. Yu, M. Dredze, Learning Composition Models for Phrase Embeddings, Transactions of the Association for Computational Linguistics 3 (2015) 227–242.

[13] R. Socher, D. Chen, C. D. Manning, A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, United States, 2013, pp. 926–934.

[14] S. Wang, C. Zong, Comparison Study on Critical Components in Composition Model for Phrase Representation, ACM Trans. Asian Low-Resour. Lang. Inf. Process. 16 (3) (2017) 16:1–16:25. doi:10.1145/3010088.

[15] D. M. Blei, Build, Compute, Critique, Repeat: Data Analysis with Latent Variable Models, Annual Review of Statistics and Its Application 1 (1) (2014) 203–232. doi:10.1146/annurev-statistics-022513-115657.

[16] M. Baroni, G. Dinu, G. Kruszewski, Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 238–247. doi:10.3115/v1/P14-1023.

[17] O. Levy, Y. Goldberg, I. Dagan, Improving Distributional Similarity with Lessons Learned from Word Embeddings, Transactions of the Association of Computational Linguistics 3 (2015) 211–225.

[18] O. Levy, Y. Goldberg, Neural Word Embedding as Implicit Matrix Factorization, in: Advances in Neural Information Processing Systems, Montreal, Quebec, Canada, 2014, pp. 2177–2185.

[19] O. Levy, Y. Goldberg, Dependency-Based Word Embeddings, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 302–308. doi:10.3115/v1/P14-2050.

[20] M. Faruqui, C. Dyer, Improving Vector Space Word Representations Using Multilingual Correlation, in: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Gothenburg, Sweden, 2014, pp. 462–471. doi:10.3115/v1/E14-1049.

[21] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge Graph and Text Jointly Embedding, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1591–1601. doi:10.3115/v1/D14-1167.

[22] M. Faruqui, Diverse Context for Learning Word Representations, Ph.D., University of Trento (2016).

[23] O. Melamud, D. McClosky, S. Patwardhan, M. Bansal, The Role of Context Types and Dimensionality in Learning Word Embeddings, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 1030–1040. `doi:10.18653/v1/N16-1118`.

[24] K. A. Nguyen, S. Schulte im Walde, N. T. Vu, Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 454–459. `doi:10.18653/v1/P16-2074`.

[25] V. Shwartz, Y. Goldberg, I. Dagan, Improving Hypernymy Detection with an Integrated Path-based and Distributional Method, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, Berlin, Germany, 2016, pp. 2389–2398.

[26] F. Sun, J. Guo, Y. Lan, J. Xu, X. Cheng, Inside Out: Two Jointly Predictive Models for Word Representations and Phrase Representations., in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA, 2016, pp. 2821–2827.

[27] P. Rastogi, B. Van Durme, R. Arora, Multiview LSA: Representation Learning via Generalized CCA, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 556–566. `doi:10.3115/v1/N15-1058`.

[28] S. L. Hyland, T. Karaletsos, G. Rtsch, A Generative Model of Words and Relationships from Multiple Sources, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona, USA., 2016, pp. 2622–2629.

[29] J. Chen, K. Chen, X. Qiu, Q. Zhang, X. Huang, Z. Zhang, Learning Word Embeddings from Intrinsic and Extrinsic Views, CoRR abs/1608.05852.

[30] R. Speer, J. Chin, C. Havasi, ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA., 2017, pp. 4444–4451.

[31] H. Ma, H. Yang, M. R. Lyu, I. King, Sorec: social recommendation using probabilistic matrix factorization, in: Proceedings of the 17th ACM conference on Information and knowledge management, ACM, Napa Valley, California, USA, 2008, pp. 931–940.

[32] M. Werning, W. Hinzen, E. Machery (Eds.), The Oxford Handbook of Compositionality, 1st Edition, Oxford University Press, Oxford ; New York, NY, 2012.

[33] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings, in: Preceedings of 5th International Conference on Learning Representations, Toulon, France, 2017.

[34] M. Baroni, R. Zamparelli, Nouns are Vectors, Adjectives are Matrices: Representing Adjective-Noun Constructions in Semantic Space, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, 2010, pp. 1183–1193.

[35] R. Socher, C. C. Lin, C. Manning, A. Y. Ng, Parsing natural scenes and natural language with recursive neural networks, in: Proceedings of the 28th International Conference on Machine Learning, Bellevue, Washington, USA, 2011, pp. 129–136.

[36] R. Socher, B. Huval, C. D. Manning, A. Y. Ng, Semantic compositionality through recursive matrix-vector spaces, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, Jeju Island, Korea, 2012, pp. 1201–1211.

[37] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, 2013, pp. 1631–1642.

[38] T. Mikolov, M. Karafit, L. Burget, J. Cernock, S. Khudanpur, Recurrent neural network based language model, in: Proceedings of 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, 2010, pp. 1045–1048.

[39] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[40] Y. Kim, Convolutional Neural Networks for Sentence Classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, A meeting of SIGDAT, a Special Interest Group of the ACL, Doha, Qatar, 2014, pp. 1746–1751.

[41] T. Pedersen, Identifying Collocations to Measure Compositionality: Shared Task System Description, in: Proceedings of the Workshop on Distributional Semantics and Compositionality, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 33–37.

[42] K. Hashimoto, Y. Tsuruoka, Adaptive Joint Learning of Compositional and Non-Compositional Phrase Embeddings, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 205–215. doi:10.18653/v1/P16-1020.

[43] T. Baldwin, Compositionality and Multiword Expressions: Six of One, Half a Dozen of the Other?, in: Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties, Association for Computational Linguistics, Sydney, Australia, 2006, p. 1.

[44] S. Reddy, D. McCarthy, S. Manandhar, An Empirical Study on Compositionality in Compound Nouns, in: Proceedings of 5th International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, Chiang Mai, Thailand, 2011, pp. 210–218.

[45] B. Salehi, P. Cook, T. Baldwin, A Word Embedding Approach to Predicting the Compositionality of Multiword Expressions, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado, USA, 2015, pp. 977–983. doi:10.3115/v1/N15-1099.

[46] H. Gong, S. Bhat, P. Viswanath, Geometry of Compositionality, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, 2017.

[47] Y. Zhao, Z. Liu, M. Sun, Phrase Type Sensitive Tensor Indexing Model for Semantic Composition, in: Proceedings of the Twenty-Ninth {AAAI} Conference on Artificial Intelligence, Austin, Texas, USA, 2015, pp. 2195–2202.

[48] J. Huang, D. Ji, S. Yao, W. Huang, B. Chen, Learning Phrase Representations Based on Word and Character Embeddings, in: Neural Information Processing, Lecture Notes in Computer Science, Springer, Cham, 2016, pp. 547–554. doi:10.1007/978-3-319-46681-1_65.

[49] J. Mitchell, M. Lapata, Vector-based Models of Semantic Composition., in: Proceedings of ACL-08: HLT, Association for Computational Linguistics, Columbus, Ohio, 2008, pp. 236–244.

[50] P. D. Turney, Domain and Function: A Dual-Space Model of Semantic Relations and Compositions, CoRR abs/1309.4035.

[51] C. Biemann, E. Giesbrecht, Distributional Semantics and Compositionality 2011: Shared Task Description and Results, in: Proceedings of the Workshop on Dis-

tributional Semantics and Compositionality, DiSCo '11, Stroudsburg, PA, USA, 2011, pp. 21–28.

[52] I. Korkontzelos, T. Zesch, F. M. Zanzotto, C. Biemann, SemEval-2013 Task 5: Evaluating Phrasal Semantics, in: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Vol. 2, Association for Computational Linguistics, Atlanta, Georgia, USA, 2013, pp. 39–47.

[53] M. Farahmand, A. Smith, J. Nivre, A Multiword Expression Data Set: Annotating Non-Compositionality and Conventionalization for English Noun Compounds, in: In Proceedings of the 11th Workshop on Multiword Expressions, NAACL, Denver, Colorado, 2015, pp. 29–33. `doi:10.3115/v1/W15-0904`.