Adaptive Joint Learning of Compositional and Non-Compositional Phrase Embeddings

Kazuma Hashimoto and Yoshimasa Tsuruoka
The University of Tokyo, 3-7-1 Hongo, Bunkyo-ku, Tokyo, Japan
{hassy,tsuruoka}@logos.t.u-tokyo.ac.jp

Abstract

We present a novel method for jointly learning compositional and non-compositional phrase embeddings by adaptively weighting both types of embeddings using a compositionality scoring function. The scoring function is used to quantify the level of compositionality of each phrase, and the parameters of the function are jointly optimized with the objective for learning phrase embeddings. In experiments, we apply the adaptive joint learning method to the task of learning embeddings of transitive verb phrases, and show that the compositionality scores have strong correlation with human ratings for verb-object compositionality, substantially outperforming the previous state of the art. Moreover, our embeddings improve upon the previous best model on a transitive verb disambiguation task. We also show that a simple ensemble technique further improves the results for both tasks.

1 Introduction

Representing words and phrases in a vector space has proven effective in a variety of language processing tasks (Pham et al., 2015; Sutskever et al., 2014). In most of the previous work, phrase embeddings are computed from word embeddings by using various kinds of composition functions. Such composed embeddings are called *compositional embeddings*. An alternative way of computing phrase embeddings is to treat phrases as single units and assigning a unique embedding to each candidate phrase (Mikolov et al., 2013; Yazdani et al., 2015). Such embeddings are called *non-compositional embeddings*.

Relying solely on non-compositional embeddings has the obvious problem of data sparsity (i.e. rare or unknown phrase problems). At the same time, however, using compositional embeddings is not always the best option since some phrases are inherently non-compositional. For example, the phrase "bear fruits" means "to yield results"[1] but it is hard to infer its meaning by composing the meanings of "bear" and "fruit". Treating all phrases as compositional also has a negative effect in learning the composition function because the words in those idiomatic phrases are not just uninformative but can serve as noisy samples in the training. These problems have motivated us to adaptively combine both types of embeddings.

Most of the existing methods for learning phrase embeddings can be divided into two approaches. One approach is to learn compositional embeddings by regarding all phrases as compositional (Pham et al., 2015; Socher et al., 2012). The other approach is to learn both types of embeddings separately and use the better ones (Kartsaklis et al., 2014; Muraoka et al., 2014). Kartsaklis et al. (2014) show that non-compositional embeddings are better suited for a phrase similarity task, whereas Muraoka et al. (2014) report the opposite results on other tasks. These results suggest that we should not stick to either of the two types of embeddings unconditionally and could learn better phrase embeddings by considering the compositionality levels of the individual phrases in a more flexible fashion.

In this paper, we propose a method that jointly learns compositional and non-compositional embeddings by adaptively weighting both types of phrase embeddings using a compositionality scoring function. The scoring function is used to quantify the level of compositionality of each phrase

[1]The definition is found at http://idioms.thefreedictionary.com/bear+fruit.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 205–215,
Berlin, Germany, August 7-12, 2016.

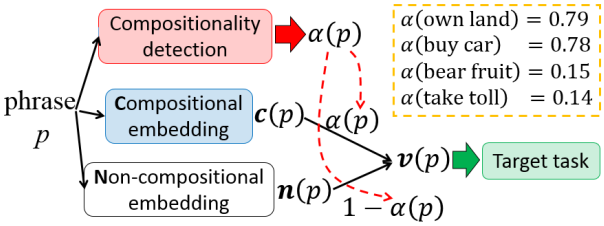

Figure 1: The overview of our method and examples of the compositionality scores. Given a phrase p, our method first computes the compositionality score $\alpha(p)$ (Eq. (3)), and then computes the phrase embedding $\boldsymbol{v}(p)$ using the compositional and non-compositional embeddings, $\boldsymbol{c}(p)$ and $\boldsymbol{n}(p)$, respectively (Eq. (2)).

and learned in conjunction with the target task for learning phrase embeddings. In experiments, we apply our method to the task of learning transitive verb phrase embeddings and demonstrate that it allows us to achieve state-of-the-art performance on standard datasets for compositionality detection and verb disambiguation.

2 Method

In this section, we describe our approach in the most general form, without specifying the function to compute the compositional embeddings or the target task for optimizing the embeddings.

Figure 1 shows the overview of our proposed method. At each iteration of the training (i.e. gradient calculation) of a certain target task (e.g. language modeling or sentiment analysis), our method first computes a compositionality score for each phrase. Then the score is used to weight the compositional and non-compositional embeddings of the phrase in order to compute the expected embedding of the phrase which is to be used in the target task. Some examples of the compositionality scores are also shown in the figure.

2.1 Compositional Phrase Embeddings

The compositional embedding $\boldsymbol{c}(p) \in \mathbb{R}^{d \times 1}$ of a phrase $p = (w_1, \cdots, w_L)$ is formulated as

$$\boldsymbol{c}(p) = f(\boldsymbol{v}(w_1), \cdots, \boldsymbol{v}(w_L)), \qquad (1)$$

where d is the dimensionality, L is the phrase length, $\boldsymbol{v}(\cdot) \in \mathbb{R}^{d \times 1}$ is a word embedding, and $f(\cdot)$ is a composition function. The function can be simple ones such as element-wise addition or multiplication (Mitchell and Lapata, 2008). More complex ones such as recurrent neural networks (Sutskever et al., 2014) are also commonly used. The word embeddings and the composition function are jointly learned on a certain target task. Since compositional embeddings are built on word-level (i.e. unigram) information, they are less prone to the data sparseness problem.

2.2 Non-Compositional Phrase Embeddings

In contrast to the compositional embedding, the non-compositional embedding of a phrase $\boldsymbol{n}(p) \in \mathbb{R}^{d \times 1}$ is independently parameterized, i.e., the phrase p is treated just like a single word. Mikolov et al. (2013) show that non-compositional embeddings are preferable when dealing with idiomatic phrases. Some recent studies (Kartsaklis et al., 2014; Muraoka et al., 2014) have discussed the (dis)advantages of using compositional or non-compositional embeddings. However, in most cases, a phrase is neither completely compositional nor completely non-compositional. To the best of our knowledge, there is no method that allows us to jointly learn both types of phrase embeddings by incorporating the levels of compositionality of the phrases as real-valued scores.

2.3 Adaptive Joint Learning

To simultaneously consider both compositional and non-compositional aspects of each phrase, we compute a phrase embedding $\boldsymbol{v}(p)$ by adaptively weighting $\boldsymbol{c}(p)$ and $\boldsymbol{n}(p)$ as follows:

$$\boldsymbol{v}(p) = \alpha(p)\boldsymbol{c}(p) + (1 - \alpha(p))\boldsymbol{n}(p), \qquad (2)$$

where $\alpha(\cdot)$ is a scoring function that quantifies the compositionality levels, and outputs a real value ranging from 0 to 1. What we expect from the scoring function is that large scores indicate high levels of compositionality. In other words, when $\alpha(p)$ is close to 1, the compositional embedding is mainly considered, and vice versa. For example, we expect $\alpha(\text{buy car})$ to be large and $\alpha(\text{bear fruit})$ to be small as shown in Figure 1.

We parameterize the scoring function $\alpha(p)$ as logistic regression:

$$\alpha(p) = \sigma(\boldsymbol{W} \cdot \boldsymbol{\phi}(p)), \qquad (3)$$

where $\boldsymbol{\phi}(p) \in \mathbb{R}^{N \times 1}$ is a feature vector of the phrase p, $\boldsymbol{W} \in \mathbb{R}^{N \times 1}$ is a weight vector, N is the number of features, and $\sigma(\cdot)$ is the logistic function. The weight vector $\boldsymbol{W}$ is jointly optimized in conjunction with the objective J for the target task of learning phrase embeddings $\boldsymbol{v}(p)$.

Updating the model parameters Given the partial derivative $\boldsymbol{\delta}_p = \frac{\partial J}{\partial \boldsymbol{v}(p)} \in \mathbb{R}^{d \times 1}$ for the target task, we can compute the partial derivative for updating $\boldsymbol{W}$ as follows:

$$\delta_\alpha = \alpha(p)(1-\alpha(p))\{\boldsymbol{\delta}_p \cdot (\boldsymbol{c}(p) - \boldsymbol{n}(p))\} \quad (4)$$

$$\frac{\partial J}{\partial \boldsymbol{W}} = \delta_\alpha \boldsymbol{\phi}(p). \quad (5)$$

If $\boldsymbol{\phi}(p)$ is not constructed by static features but is computed by a feature learning model such as neural networks, we can propagate the error term δ_α into the feature learning model by the following equation:

$$\frac{\partial J}{\partial \boldsymbol{\phi}(p)} = \delta_\alpha \boldsymbol{W}. \quad (6)$$

When we use only static features, as in this work, we can simply compute the partial derivatives of J with respect to $\boldsymbol{c}(p)$ and $\boldsymbol{n}(p)$ as follows:

$$\frac{\partial J}{\partial \boldsymbol{c}(p)} = \alpha(p)\boldsymbol{\delta}_p \quad (7)$$

$$\frac{\partial J}{\partial \boldsymbol{n}(p)} = (1-\alpha(p))\boldsymbol{\delta}_p. \quad (8)$$

As mentioned above, Eq. (7) and (8) show that the non-compositional embeddings are mainly updated when $\alpha(p)$ is close to 0, and vice versa. The partial derivative $\frac{\partial J}{\partial \boldsymbol{c}(p)}$ is used to update the model parameters in the composition function via the backpropagation algorithm. Any differentiable composition functions can be used in our method.

Expected behavior of our method The training of our method depends on the target task; that is, the model parameters are updated so as to minimize the cost function as described above. More concretely, $\alpha(p)$ for each phrase p is adaptively adjusted so that the corresponding parameter updates contribute to minimizing the cost function. As a result, different phrases will have different $\alpha(p)$ values depending on their compositionality. If the size of the training data were almost infinitely large, $\alpha(p)$ for all phrases would become nearly zero, and the non-compositional embeddings $\boldsymbol{n}(p)$ are dominantly used (since that would allow the model to better fit the data). In reality, however, the amount of the training data is limited, and thus the compositional embeddings $\boldsymbol{c}(p)$ are effectively used to overcome the data sparseness problem.

3 Learning Verb Phrase Embeddings

This section describes a particular instantiation of our approach presented in the previous section, focusing on the task of learning the embeddings of transitive verb phrases.

3.1 Word and Phrase Prediction in Predicate-Argument Relations

Acquisition of selectional preference using embeddings has been widely studied, where word and/or phrase embeddings are learned based on syntactic links (Bansal et al., 2014; Hashimoto and Tsuruoka, 2015; Levy and Goldberg, 2014; Van de Cruys, 2014). As with language modeling, these methods perform word (or phrase) prediction using (syntactic) contexts.

In this work, we focus on verb-object relationships and employ a phrase embedding learning method presented in Hashimoto and Tsuruoka (2015). The task is a plausibility judgment task for predicate-argument tuples. They extracted Subject-Verb-Object (*SVO*) and SVO-Preposition-Noun (*SVOPN*) tuples using a probabilistic HPSG parser, *Enju* (Miyao and Tsujii, 2008), from the training corpora. Transitive verbs and prepositions are extracted as predicates with two arguments. For example, the extracted tuples include (S, V, O) = ("importer", "make", "payment") and (SVO, P, N) = ("importer make payment", "in", "currency"). The task is to discriminate between observed and unobserved tuples, such as the (S, V, O) tuple mentioned above and (S, V', O) = ("importer", "*eat*", "payment"), which is generated by replacing "make" with "eat". The (S, V', O) tuple is unlikely to be observed.

For each tuple (p, a_1, a_2) observed in the training data, a cost function is defined as follows:

$$\begin{aligned} -\log\sigma(s(p,a_1,a_2)) &- \log\sigma(-s(p',a_1,a_2)) \\ &- \log\sigma(-s(p,a_1',a_2)) \\ &- \log\sigma(-s(p,a_1,a_2')), \end{aligned} \quad (9)$$

where $s(\cdot)$ is a plausibility scoring function, and p, a_1 and a_2 are a predicate and its arguments, respectively. Each of the three unobserved tuples (p', a_1, a_2), (p, a_1', a_2), and (p, a_1, a_2') is generated by replacing one of the entries with a random sample.

In their method, each predicate p is represented with a matrix $\boldsymbol{M}(p) \in \mathbb{R}^{d \times d}$ and each argument a with an embedding $\boldsymbol{v}(a) \in \mathbb{R}^{d \times 1}$. The matrices and embeddings are learned by minimizing the cost function using *AdaGrad* (Duchi et al., 2011). The scoring function is parameterized as

$$s(p, a_1, a_2) = \boldsymbol{v}(a_1) \cdot (\boldsymbol{M}(p)\boldsymbol{v}(a_2)), \quad (10)$$

and the VO and SVO embeddings are computed as

$$\boldsymbol{v}(VO) = \boldsymbol{M}(V)\boldsymbol{v}(O) \quad (11)$$

$$\boldsymbol{v}(SVO) = \boldsymbol{v}(S) \odot \boldsymbol{v}(VO), \quad (12)$$

as proposed by Kartsaklis et al. (2012). The operator $\odot$ denotes element-wise multiplication. In summary, the scores are computed as

$$s(V, S, O) = \boldsymbol{v}(S) \cdot \boldsymbol{v}(VO) \quad (13)$$

$$s(P, SVO, N) = \boldsymbol{v}(SVO) \cdot (\boldsymbol{M}(P)\boldsymbol{v}(N)). \quad (14)$$

With this method, the word and composed phrase embeddings are jointly learned based on co-occurrence statistics of predicate-argument structures. Using the learned embeddings, they achieved state-of-the-art accuracy on a transitive verb disambiguation task (Grefenstette and Sadrzadeh, 2011).

3.2 Applying the Adaptive Joint Learning

In this section, we apply our adaptive joint learning method to the task described in Section 3.1. We here redefine the computation of $\boldsymbol{v}(VO)$ by first replacing $\boldsymbol{v}(VO)$ in Eq. (11) with $\boldsymbol{c}(VO)$ as,

$$\boldsymbol{c}(VO) = \boldsymbol{M}(V)\boldsymbol{v}(O), \quad (15)$$

and then assigning VO to p in Eq. (2) and (3):

$$\boldsymbol{v}(VO) = \alpha(VO)\boldsymbol{c}(VO) + (1 - \alpha(VO))\boldsymbol{n}(VO), \quad (16)$$

$$\alpha(VO) = \sigma(\boldsymbol{W} \cdot \boldsymbol{\phi}(VO)). \quad (17)$$

The $\boldsymbol{v}(VO)$ in Eq. (16) is used in Eq. (12) and (13). We assume that the candidates of the phrases are given in advance. For the phrases not included in the candidates, we set $\boldsymbol{v}(VO) = \boldsymbol{c}(VO)$. This is analogous to the way a human guesses the meaning of an idiomatic phrase she does not know. We should note that $\boldsymbol{\phi}(VO)$ can be computed for phrases not included in the candidates, using partial features among the features described below. If any features do not fire, $\boldsymbol{\phi}(VO)$ becomes 0.5 according to the logistic function.

For the feature vector $\boldsymbol{\phi}(VO)$, we use the following simple binary and real-valued features:

- indices of V, O, and VO
- frequency and Pointwise Mutual Information (PMI) values of VO.

More concretely, the first set of the features (indices of V, O, and VO) is the concatenation of traditional one-hot vectors. The second set of features, frequency and PMI (Church and Hanks, 1990) features, have proven effective in detecting the compositionality of transitive verbs in McCarthy et al. (2007) and Venkatapathy and Joshi (2005). Given the training corpus, the frequency feature for a VO pair is computed as

$$freq(VO) = \log(count(VO)), \quad (18)$$

where $count(VO)$ counts how many times the VO pair appears in the training corpus, and the PMI feature is computed as

$$\mathrm{PMI}(VO) = \log \frac{count(VO)count(*)}{count(V)count(O)}, \quad (19)$$

where $count(V)$, $count(O)$, and $count(*)$ are the counts of the verb V, the object O, and all VO pairs in the training corpus, respectively. We normalize the frequency and PMI features so that their maximum absolute value becomes 1.

4 Experimental Settings

4.1 Training Data

As the training data, we used two datasets, one small and one large: the British National Corpus (BNC) (Leech, 1992) and the English Wikipedia. More concretely, we used the publicly available data[2] preprocessed by Hashimoto and Tsuruoka (2015). The BNC data consists of 1.38 million SVO tuples and 0.93 million SVOPN tuples. The Wikipedia data consists of 23.6 million SVO tuples and 17.3 million SVOPN tuples. Following the provided code[3], we used exactly the same train/development/test split (0.8/0.1/0.1) for training the overall model. As the third training data, we also used the concatenation of the two data, which is hereafter referred to as *BNC-Wikipedia*.

We applied our adaptive joint learning method to verb-object phrases observed more than K times in each corpus. K was set to 10 for the BNC data and 100 for the Wikipedia and BNC-Wikipedia data. Consequently, the non-compositional embeddings were assigned to 17,817, 28,933, and 30,682 verb-object phrase types in the BNC, Wikipedia, and BNC-Wikipedia data, respectively.

[2] http://www.logos.t.u-tokyo.ac.jp/~hassy/publications/cvsc2015/

[3] https://github.com/hassyGo/SVOembedding

4.2 Training Details

The model parameters consist of d-dimensional word embeddings for nouns, non-compositional phrase embeddings, $d \times d$-dimensional matrices for verbs and prepositions, and a weight vector $\boldsymbol{W}$ for $\alpha(VO)$. All the model parameters are jointly optimized. We initialized the embeddings and matrices with zero-mean gaussian random values with a variance of $\frac{1}{d}$ and $\frac{1}{d^2}$, respectively, and $\boldsymbol{W}$ with zeros. Initializing $\boldsymbol{W}$ with zeros forces the initial value of each $\alpha(VO)$ to be 0.5 since we use the logistic function to compute $\alpha(VO)$.

The optimization was performed via mini-batch AdaGrad (Duchi et al., 2011). We fixed d to 25 and the mini-batch size to 100. We set candidate values for the learning rate ε to $\{0.01, 0.02, 0.03, 0.04, 0.05\}$. For the weight vector $\boldsymbol{W}$, we employed L2-norm regularization and set the coefficient λ to $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 0\}$. For selecting the hyperparameters, each training process was stopped when the evaluation score on the development split decreased. Then the best performing hyperparameters were selected for each training dataset. Consequently, ε was set to 0.05 for all training datasets, and λ was set to 10^{-6}, 10^{-3}, and 10^{-5} for the BNC, Wikipedia, and BNC-Wikipedia data, respectively. Once the training is finished, we can use the learned embeddings and the scoring function in downstream target tasks.

5 Evaluation on the Compositionality Detection Function

5.1 Evaluation Settings

Datasets First, we evaluated the learned compositionality detection function on two datasets, **VJ'05**[4] and **MC'07**[5], provided by Venkatapathy and Joshi (2005) and McCarthy et al. (2007), respectively. VJ'05 consists of 765 verb-object pairs with human ratings for the compositionality. MC'07 is a subset of VJ'05 and consists of 638 verb-object pairs. For example, the rating of "buy car" is 6, which is the highest score, indicating the phrase is highly compositional. The rating of "bear fruit " is 1, which is the lowest score, indicating the phrase is highly non-compositional.

[4] http://www.dianamccarthy.co.uk/downloads/SVAJ2005compositionality_rating.txt

[5] http://www.dianamccarthy.co.uk/downloads/emnlp2007data.txt

Method	MC'07	VJ'05
Proposed method (Wikipedia)	0.508	0.514
Proposed method (BNC)	0.507	0.507
Proposed method (BNC-Wikipedia)	0.518	0.527
Proposed method (Ensemble)	0.550	0.552
Kiela and Clark (2013) w/ WordNet	n/a	0.461
Kiela and Clark (2013)	n/a	0.420
DSPROTO (McCarthy et al., 2007)	0.398	n/a
PMI (McCarthy et al., 2007)	0.274	n/a
Frequency (McCarthy et al., 2007)	0.141	n/a
DSPROTO+ (McCarthy et al., 2007)	0.454	n/a
Human agreement	0.702	0.716

Table 1: Compositionality detection task.

Evaluation metric The evaluation was performed by calculating Spearman's rank correlation scores[6] between the averaged human ratings and the learned compositionality scores $\alpha(VO)$.

Ensemble technique We also produced the result by employing an *ensemble* technique. More concretely, we used the averaged compositionality scores from the results of the BNC and Wikipedia data for the ensemble result.

5.2 Results and Discussion

5.2.1 Result Overview

Table 1 shows our results and the state of the art. Our method outperforms the previous state of the art in all settings. The result denoted as *Ensemble* is the one that employs the ensemble technique, and achieves the strongest correlation with the human-annotated datasets. Even without the ensemble technique, our method performs better than all of the previous methods.

Kiela and Clark (2013) used window-based co-occurrence vectors and improved their score using WordNet hypernyms. By contrast, our method does not rely on such external resources, and only needs parsed corpora. We should note that Kiela and Clark (2013) reported that their score did not improve when using parsed corpora. Our method also outperforms DSPROTO+, which used a small amount of the labeled data, while our method is fully unsupervised.

We calculated confidence intervals ($P < 0.05$) using bootstrap resampling (Noreen, 1989). For example, for the results using the BNC-Wikipedia data, the intervals on MC'07 and VJ'05 are (0.455, 0.574) and (0.475, 0.579), respectively. These results show that our method significantly outperforms the previous state-of-the-art results.

[6] We used the Scipy 0.12.0 implementation in Python.

Phrase		Gold standard	(a) BNC	(b) Wikipedia	BNC-Wikipedia	Ensemble ((a)+(b))×0.5
(A)	buy car	6	0.78	0.71	0.80	0.74
	own land	6	0.79	0.73	0.76	0.76
	take toll	1.5	0.14	0.11	0.06	0.13
	shed light	1	0.21	0.07	0.07	0.14
	bear fruit	1	0.15	0.19	0.17	0.17
(B)	make noise	6	0.37	0.33	0.30	0.35
	have reason	5	0.26	0.39	0.33	0.33
(C)	smoke cigarette	6	0.56	0.90	0.78	0.73
	catch eye	1	0.48	0.14	0.17	0.31

Table 2: Examples of the compositionality scores.

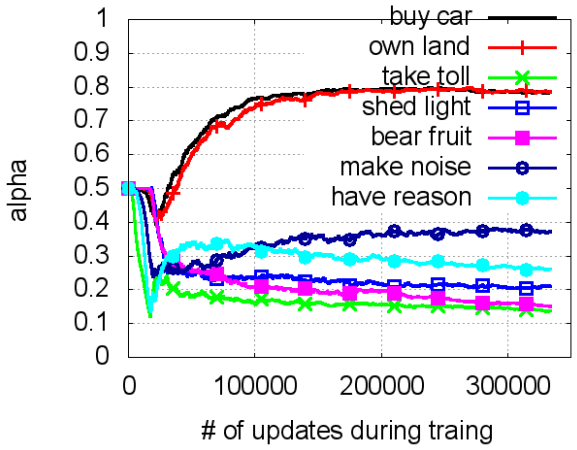

Figure 2: Trends of $\alpha(VO)$ during the training on the BNC data.

Highest average scores		Lowest average scores	
approve	0.83	bear	0.37
reject	0.72	play	0.38
discuss	0.71	have	0.38
visit	0.70	make	0.39
want	0.70	break	0.40
describe	0.70	take	0.40
involve	0.69	raise	0.41
own	0.68	reach	0.41
attend	0.68	gain	0.42
reflect	0.67	draw	0.42

Table 3: The 10 highest and lowest average compositionality scores with the corresponding verbs on the BNC data.

5.2.2 Analysis of Compositionality Scores

Figure 2 shows how $\alpha(VO)$ changes for the seven phrases during the training on the BNC data. As shown in the figure, starting from 0.5, $\alpha(VO)$ for each phrase converges to its corresponding value. The differences in the trends indicate that our method can adaptively learn compositionality levels for the phrases. Table 2 shows the learned compositionality scores for the three groups of the examples along with the gold-standard scores given by the annotators. The group (A) is considered to be consistent with the gold-standard scores, the group (B) is not, and the group (C) shows examples for which the difference between the compositionality scores of our results is large.

Characteristics of light verbs The verbs "take", "make", and "have" are known as *light verbs* [7], and the scoring function tends to assign low scores to light verbs. In other words, our method can recognize that the light verbs are frequently used to form idiomatic (i.e. non-compositional) phrases. To verify the assumption, we calculated the average compositionality score for each verb by averaging the compositionality scores paired with its candidate objects. Here we used 135 verbs which take more than 30 types of objects in the BNC data. Table 3 shows the 10 highest and lowest average scores with the corresponding verbs. We see that relatively low scores are assigned to the light verbs as well as other verbs which often form idiomatic phrases. As shown in the group (B) in Table 2, however, light verb phrases are not always non-compositional. Despite this, the learned function assigns low scores to compositional phrases formed by the light verbs. These results suggest that using a more flexible scoring function may further strengthen our method.

Context dependence Both our method and the two datasets, VJ'05 and MC'07, assume that the compositionality score can be computed for each phrase with no contextual information. However, in general, the compositionality level of a phrase depends on its contextual information. For example, the meaning of the idiomatic phrase "bear

[7] In Section 5.2.2 in Newton (2006), the term *light verb* is used to refer to verbs which can be used in combination with some other element where their contribution to the meaning of the whole construction is reduced in some way.

fruit" can be compositionaly interpreted as "to yield fruit" for a plant or tree. We manually inspected the BNC data to check whether the phrase "bear fruit" is used as the compositional meaning or the idiomatic meaning ("to yield results"). As a result, we have found that most of the usage was its idiomatic meaning. In the model training, our method is affected by the majority usage and fits the evaluation datasets where the phrase "bear fruit" is regarded as highly non-compositional. Incorporating contextual information into the compositionality scoring function is a promising direction of future work.

5.2.3 Effects of Ensemble

We used the two different corpora for constructing the training data, and our method achieves the state-of-the-art results in all settings. To inspect the results on VJ'05, we calculated the correlation score between the outputs from our results of the BNC and Wikipedia data. The correlation score is 0.674 and that is, the two different corpora lead to reasonably consistent results, which indicates the robustness of our method. However, the correlation score is still much lower than perfect correlation; in other words, there are disagreements between the outputs learned with the corpora. The group (C) in Table 2 shows such two examples. In these cases, the ensemble technique is helpful in improving the results as shown in the examples.

Another interesting observation in our results is that the result of the ensemble technique outperforms that of the BNC-Wikipedia data as shown in Table 1. This shows that separately using the training corpora of different nature and then performing the ensemble technique can yield better results. By contrast, many of the previous studies on embedding-based methods combine different corpora into a single dataset, or use multiple corpora just separately and compare them (Hashimoto and Tsuruoka, 2015; Muraoka et al., 2014; Pennington et al., 2014). It would be worth investigating whether the results in the previous work can be improved by ensemble techniques.

6 Evaluation on the Phrase Embeddings

6.1 Evaluation Settings

Dataset Next, we evaluated the learned embeddings on the transitive verb disambiguation dataset **GS'11**[8] provided by Grefenstette and Sadrzadeh (2011). GS'11 consists of 200 pairs of transitive verbs and each verb pair takes the same subject and object. For example, the transitive verb "run" is known as a polysemous word and this task requires one to identify the meanings of "run" and "operate" as similar to each other when taking "people" as their subject and "company" as their object. In the same setting, however, the meanings of "run" and "move" are not similar to each other. Each pair has multiple human ratings indicating how similar the phrases of the pair are.

Evaluation metric The evaluation was performed by calculating Spearman's rank correlation scores between the human ratings and the cosine similarity scores of $\boldsymbol{v}(SVO)$ in Eq. (12). Following the previous studies, we used the gold-standard ratings in two ways: averaging the human ratings for each SVO tuple (GS'11a) and treating each human rating separately (GS'11b).

Ensemble technique We used the same ensemble technique described in Section 5.1. In this task we produced two ensemble results: *Ensemble A* and *Ensemble B*. The former used the averaged cosine similarity from the results of the BNC and Wikipedia data, and the latter further incorporated the result of the BNC-Wikipedia data.

Baselines We compared our adaptive joint learning method with two baseline methods. One is the method in Hashimoto and Tsuruoka (2015) and it is equivalent to fixing $\alpha(VO)$ to 1 in our method. The other is fixing $\alpha(VO)$ to 0.5 in our method, which serves as a baseline to evaluate how effective the proposed adaptive weighting method is.

6.2 Results and Discussion

6.2.1 Result Overview

Table 4 shows our results and the state of the art, and our method outperforms almost all of the previous methods in both datasets. Again, the ensemble technique further improves the results, and overall, Ensemble B yields the best results.

The scores in Hashimoto and Tsuruoka (2015), the baseline results with $\alpha(VO) = 1$ in our method, have been the best to date. As shown in Table 4, our method outperforms the baseline results with $\alpha(VO) = 0.5$ as well as those

[8] http://www.cs.ox.ac.uk/activities/compdistmeaning/GS2011data.txt

	Proposed method		$\alpha(VO) = 1$	$\alpha(VO) = 0.5$
take toll	$\alpha(\text{take toll}) = 0.11$	put strain place strain cause strain have affect exacerbate injury	deplete division necessitate monitoring deplete pool create pollution deplete field	put strain cause lack befall army exacerbate weakness cause strain
catch eye	$\alpha(\text{catch eye}) = 0.14$	catch attention grab attention make impression lift spirit become favorite	catch ear catch heart catch e-mail catch imagination catch attention	grab attention make impression catch attention become legend inspire playing
bear fruit	$\alpha(\text{bear fruit}) = 0.19$	accentuate effect enhance beauty enhance atmosphere rejuvenate earth enhance habitat	bear herb bear grain bear spore bear variety bear seed	increase richness reduce biodiversity fuel boom enhance atmosphere worsen violence
make noise	$\alpha(\text{make noise}) = 0.33$	attack intruder attack trespasser avoid predator attack diver attack pedestrian	make sound do beating get bounce get pulse lose bit	burn can kill monster wash machine lightn flash cook raman
buy car	$\alpha(\text{buy car}) = 0.71$	buy bike buy machine buy motorcycle buy automobile purchase coins	buy truck buy bike buy automobile buy motorcycle buy vehicle	buy bike buy instrument buy chip buy scooter buy motorcycle

Table 5: Examples of the closest neighbors in the learned embedding space. All of the results were obtained by using the Wikipedia data, and the values of $\alpha(VO)$ are the same as those in Table 2.

Method	GS'11a	GS'11b
Proposed method (Wikipedia)	0.598	0.461
Proposed method (BNC)	0.595	0.463
Proposed method (BNC-Wikipedia)	0.623	0.483
Proposed method (Ensemble A)	0.661	0.511
Proposed method (Ensemble B)	0.680	0.524
$\alpha(VO) = 0.5$ (Wikipedia)	0.491	0.386
$\alpha(VO) = 0.5$ (BNC)	0.599	0.462
$\alpha(VO) = 0.5$ (BNC-Wikipedia)	0.610	0.477
$\alpha(VO) = 0.5$ (Ensemble A)	0.612	0.474
$\alpha(VO) = 0.5$ (Ensemble B)	0.638	0.495
$\alpha(VO) = 1$ (Wikipedia)	0.576	n/a
$\alpha(VO) = 1$ (BNC)	0.574	n/a
Milajevs et al. (2014)	0.456	n/a
Polajnar et al. (2014)	n/a	0.370
Hashimoto et al. (2014)	0.420	0.340
Polajnar et al. (2015)	n/a	0.330
Grefenstette and Sadrzadeh (2011)	n/a	0.210
Human agreement	0.750	0.620

Table 4: Transitive verb disambiguation task. The results for $\alpha(VO) = 1$ are reported in Hashimoto and Tsuruoka (2015).

with $\alpha(VO) = 1$. We see that our method improves the baseline scores by adaptively combining compositional and non-compositional embeddings. Along with the results in Table 1, these results show that our method allows us to improve the composition function by jointly learning non-compositional embeddings and the scoring function for compositionality detection.

6.2.2 Analysis of the Learned Embeddings

We inspected the effects of adaptively weighting the compositional and non-compositional embeddings. Table 5 shows the five closest neighbor phrases in terms of the cosine similarity for the three idiomatic phrases "take toll", "catch eye", and "bear fruit" as well as the two non-idiomatic phrases "make noise" and "buy car". The examples trained with the Wikipedia data are shown for our method and the two baselines, i.e., $\alpha(VO) = 1$ and $\alpha(VO) = 0.5$. As shown in Table 2, the compositionality levels of the first three phrases are low and their non-compositional embeddings are dominantly used to represent their meaning.

One observation with $\alpha(VO) = 1$ is that head words (i.e. verbs) are emphasized in the shown examples except "take toll" and "make noise". As with other embedding-based methods, the compositional embeddings are highly affected by their component words. As a result, the phrases consisting of the same verb and the similar objects are often listed as the closest neighbors. By contrast, our method flexibly allows us to adaptively omit the information about the component words. Therefore, our method puts more weight on capturing the idiomatic aspects of the example phrases by

adaptively using the non-compositional embeddings.

The results of $\alpha(VO) = 0.5$ are similar to those with our proposed method, but we can see some differences. For example, the phrase list for "make noise" of our proposed method captures offensive meanings, whereas that of $\alpha(VO) = 0.5$ is somewhat ambiguous. As another example, the phrase lists for "buy car" show that our method better captures the semantic similarity between the objects than $\alpha(VO) = 0.5$. This is achieved by adaptively assigning a relatively large compositionality score (0.71) to the phrase to use the information about the object "car".

We should note that "make noise" is highly compositional but our method outputs $\alpha(\text{make noise}) = 0.33$, and the phrase list of $\alpha(VO) = 1$ is the most appropriate in this case. Improving the compositionality detection function should thus further improve the learned embeddings.

7 Related Work

Learning embeddings of words and phrases has been widely studied, and the phrase embeddings have proven effective in many language processing tasks, such as machine translation (Cho et al., 2014; Sutskever et al., 2014), sentiment analysis and semantic textual similarity (Tai et al., 2015). Most of the phrase embeddings are constructed by word-level information via various kinds of composition functions like long short-term memory (Hochreiter and Schmidhuber, 1997) recurrent neural networks. Such composition functions should be powerful enough to efficiently encode information about all the words into the phrase embeddings. By simultaneously considering the compositionality of the phrases, our method would be helpful in saving the composition models from having to be powerful enough to perfectly encode the non-compositional phrases. As a first step towards this purpose, in this paper we have shown the effectiveness of our method on the task of learning verb phrase embeddings.

Many studies have focused on detecting the compositionality of a variety of phrases (Lin, 1999), including the ones on verb phrases (Diab and Bhutada, 2009; McCarthy et al., 2003) and compound nouns (Farahmand et al., 2015; Reddy et al., 2011). Compared to statistical feature-based methods (McCarthy et al., 2007; Venkatapathy and Joshi, 2005), recent methods use word and phrase embeddings (Kiela and Clark, 2013; Yazdani et al., 2015). The embedding-based methods assume that word embeddings are given in advance and as a post-processing step, learn or simply employ composition functions to compute phrase embeddings. In other words, there is no distinction between compositional and non-compositional phrases. Yazdani et al. (2015) further proposed to incorporate latent annotations (binary labels) for the compositionality of the phrases. However, binary judgments cannot consider numerical scores of the compositionality. By contrast, our method adaptively weights the compositional and non-compositional embeddings using the compositionality scoring function.

8 Conclusion and Future Work

We have presented a method for adaptively learning compositional and non-compositional phrase embeddings by jointly detecting compositionality levels of phrases. Our method achieves the state of the art on a compositionality detection task of verb-object pairs, and also improves upon the previous state-of-the-art method on a transitive verb disambiguation task. In future work, we will apply our method to other kinds of phrases and tasks.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by CREST, JST.

References

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring Continuous Word Representations for Dependency Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 809–815.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Kenneth Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information and Lexicography. *Computational Linguistics*, 19(2):263–312.

Mona Diab and Pravin Bhutada. 2009. Verb Noun Construction MWE Token Classification. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 17–22.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Meghdad Farahmand, Aaron Smith, and Joakim Nivre. 2015. A Multiword Expression Data Set: Annotating Non-Compositionality and Conventionalization for English Noun Compounds. In *Proceedings of the 11th Workshop on Multiword Expressions*, pages 29–33.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental Support for a Categorical Compositional Distributional Model of Meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2015. Learning Embeddings for Transitive Verb Disambiguation by Implicit Tensor Factorization. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 1–11.

Kazuma Hashimoto, Pontus Stenetorp, Makoto Miwa, and Yoshimasa Tsuruoka. 2014. Jointly Learning Word Representations and Composition Functions Using Predicate-Argument Structures. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1544–1555.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. 2012. A Unified Sentence Space for Categorical Distributional-Compositional Semantics: Theory and Experiments. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 549–558.

Dimitri Kartsaklis, Nal Kalchbrenner, and Mehrnoosh Sadrzadeh. 2014. Resolving Lexical Ambiguity in Tensor Regression Models of Meaning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 212–217.

Douwe Kiela and Stephen Clark. 2013. Detecting Compositionality of Multi-Word Expressions using Nearest Neighbours in Vector Space Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1427–1432.

Geoffrey Leech. 1992. 100 Million Words of English: the British National Corpus. *Language Research*, 28(1):1–13.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308.

Dekang Lin. 1999. Automatic Identification of Non-compositional Phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 317–324.

Diana McCarthy, Bill Keller, and John Carroll. 2003. Detecting a Continuum of Compositionality in Phrasal Verbs. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 73–80.

Diana McCarthy, Sriram Venkatapathy, and Aravind Joshi. 2007. Detecting Compositionality of Verb-Object Combinations using Selectional Preferences. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 369–379.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. 2014. Evaluating Neural Word Representations in Tensor-Based Compositional Settings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 708–719.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based Models of Semantic Composition. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 236–244.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature Forest Models for Probabilistic HPSG Parsing. *Computational Linguistics*, 34(1):35–80, March.

Masayasu Muraoka, Sonse Shimaoka, Kazeto Yamamoto, Yotaro Watanabe, Naoaki Okazaki, and Kentaro Inui. 2014. Finding The Best Model Among Representative Compositional Models. In *Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation*, pages 65–74.

Mark Newton. 2006. *Basic English Syntax with Exercises*. Bölcsész Konzorcium.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Nghia The Pham, Germán Kruszewski, Angeliki Lazaridou, and Marco Baroni. 2015. Jointly optimizing word representations for lexical and sentential tasks with the C-PHRASE model. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 971–981.

Tamara Polajnar, Laura Rimell, and Stephen Clark. 2014. Using Sentence Plausibility to Learn the Semantics of Transitive Verbs. In *Proceedings of Workshop on Learning Semantics at the 2014 Conference on Neural Information Processing Systems*.

Tamara Polajnar, Laura Rimell, and Stephen Clark. 2015. An Exploration of Discourse-Based Sentence Spaces for Compositional Distributional Semantics. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 1–11.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. 2011. An Empirical Study on Compositionality in Compound Nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 210–218.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. 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*, pages 1201–1211.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566.

Tim Van de Cruys. 2014. A Neural Network Approach to Selectional Preference Acquisition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 26–35.

Sriram Venkatapathy and Aravind Joshi. 2005. Measuring the Relative Compositionality of Verb-Noun (V-N) Collocations by Integrating Features. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 899–906.

Majid Yazdani, Meghdad Farahmand, and James Henderson. 2015. Learning Semantic Composition to Detect Non-compositionality of Multiword Expressions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1733–1742.