

Additive Compositionality of Word Vectors

Yeon Seonwoo, Sungjoon Park, Dongkwan Kim, and Alice Oh

School of Computing, KAIST, Republic of Korea

{yeon.seonwoo, sungjoon.park, dongkwan.kim}@kaist.ac.kr,
alice.oh@kaist.edu

Abstract

Additive compositionality of word embedding models has been studied from empirical and theoretical perspectives. Existing research on justifying additive compositionality of existing word embedding models requires a rather strong assumption of uniform word distribution. In this paper, we relax that assumption and propose more realistic conditions for proving additive compositionality, and we develop a novel word and sub-word embedding model that satisfies additive compositionality under those conditions. We then empirically show our model’s improved semantic representation performance on word similarity and noisy sentence similarity.

1 Introduction

Previous word embedding studies have empirically shown linguistic regularities represented as linear translation in the word vector space, but they do not explain these empirical results mathematically (Mikolov et al., 2013b; Pennington et al., 2014; Bojanowski et al., 2017).

Recent studies present theoretical advances to interpret these word embedding models. Levy and Goldberg (2014b) show that the global optimum of SGNS (Skip-Gram with Negative Sampling) is the shifted PMI ($\text{PMI}(i, j) - k$). Arora et al. (2016) propose a generative model to explain PMI-based distributional models and presents a mathematical explanation of the linguistic regularity in Skip-Gram (Mikolov et al., 2013a). Gittens et al. (2017) provide a theoretical justification of the additive compositionality of Skip-Gram and shows that the linguistic regularity of Skip-Gram is explained by additive compositionality of Skip-Gram (Mikolov et al., 2013a). One property of the word vectors that equates to satisfying additive compositionality is the following:

$$\vec{u}_c = \sum_{i=1}^n \vec{u}_{c_i}$$

where word c is the paraphrase word of the set of words $\{c_1, \dots, c_n\}$, and \vec{u} is the vector representation of a word. We explain additive compositionality in more detail in section 3.3.

In this paper, we provide a more sound mathematical explanation of linguistic regularity to overcome the limitations of previous theoretical explanations. For instance, Levy and Goldberg (2014b) do not provide a connection between shifted-PMI and linguistic regularity, and Arora et al. (2016) and Gittens et al. (2017) require strong assumptions in their mathematical explanation about linguistic regularity. Arora et al. (2016) assume isotropy, a uniformly distributed word vector space, and Gittens et al. (2017) assume a uniform word frequency distribution within a corpus, $p(w) = 1/|V|$.

We propose a novel word/sub-word embedding model which we call OLIVE that satisfies exact additive compositionality. The objective function of OLIVE consists of two parts. One is a global co-occurrence term to capture the semantic similarity of words. The other is a regularization term to constrain the size of the inner product of co-occurring word pairs. We show that the global optimum point of OLIVE is the exact PMI matrix under certain condition unlike SGNS whose optimum approximates PMI due to the sampling process in training (Levy and Goldberg, 2014b). The source code and pre-trained word vectors of OLIVE are publicly available¹. By being a more theoretically sound word embedding model OLIVE shows improved empirical performance for semantic representation of word vectors, and by eliminating sampling process in SGNS, OLIVE shows robustness on the size of the vocabulary

¹<https://github.com/yeonsw/OLIVE>

and sentence representation performance on various noisy settings. We evaluate the semantic representation performance with the word similarity task, and we show the robustness of our model by conducting word similarity task on various vocabulary size and sentence similarity task on various noisy settings.

The contributions of our research are as follows:

- We present a novel mathematical explanation of additive compositionality of SGNS.
- We propose a word/sub-word embedding model that theoretically satisfies additive compositionality. We provide the code for this model for reproducibility.
- In addition to theoretical justification, we show the empirical performance of our model and its robustness.

2 Related Work

Learning word co-occurrence distribution is known as an effective method to capture the semantics of words (Baroni and Lenci, 2010; Harris, 1954; Miller and Charles, 1991; Bullinaria and Levy, 2007).

Based on this word co-occurrence distribution, two major types of word embedding research have been conducted. One is to use the local context of words in a corpus to train a neural network (Mikolov et al., 2013a,b; Bojanowski et al., 2017; Xu et al., 2018; Khodak et al., 2018). The other is to use the global statistics (Huang et al., 2012; Pennington et al., 2014). Aside from the general purpose embedding, some approaches are specific to a domain (Shi et al., 2018) or use extra human labeled information (Wang et al., 2018). In this paper, we focus on the general purpose word embedding.

Among those, SGNS (Skip-Gram with Negative Sampling) (Mikolov et al., 2013b) and FastText (Bojanowski et al., 2017) are widely-used neural network based word and sub-word embedding models that use negative sampling (Gutmann and Hyvärinen, 2012; Mnih and Kavukcuoglu, 2013). With the argument that global co-occurrence statistics, overlooked in SGNS and FastText, are important in capturing the word semantics, Pennington et al. (2014) propose GloVe.

There were empirical studies of linguistic regularity in word embedding vector space (Levy and

Goldberg, 2014a; Socher et al., 2012; Botha and Blunsom, 2014; Mikolov et al., 2013c; Socher et al., 2013).

Although Skip-Gram and GloVe seem to capture the linguistic regularity sufficiently, from the theoretical perspective of additive compositionality of word vectors, both models are lacking because they require extra and strong assumptions (Gittens et al., 2017; Arora et al., 2016).

Recently, (Allen and Hospedales, 2019) claim strong assumptions in the previous word embedding studies (Gittens et al., 2017; Arora et al., 2016) and propose theoretical explanation about linguistic regularity in SGNS based on paraphrase definition in (Gittens et al., 2017). Although their theory explains linguistic regularity in SGNS, there are remaining mathematical properties in SGNS: uniqueness of the paraphrase vector, and the meaning of the negative sampling parameter.

In this paper, we recognize the importance of additive compositionality of word vectors which connects word vectors, and their linguistic regularity and we provide a novel mathematical explanation of SGNS’s additive compositionality and uniqueness property in additive compositionality. Further, we propose a novel word/sub-word embedding model that satisfies additive compositionality based on our theory and show its capabilities in capturing the semantics of words.

3 Preliminaries

In this section, we describe three important concepts about word vectors and how they are used in existing word embedding models. First, we describe the PMI matrix which is an approximated global optimum point of SGNS (Skip-Gram with Negative Sampling) (Levy and Goldberg, 2014b) and known to capture the semantic meaning of words in vector space (Arora et al., 2016). Second, we describe sub-sampling, a word frequency balancing method to increase word embedding performance. Third, we describe additive compositionality, the notion that a paraphrase word vector is a vector sum of its context word set (Gittens et al., 2017).

3.1 Objective Function

There are two major types of objective functions used in word embedding models. One is used in SGNS and is based on the PMI; and the other is used in GloVe, based on the joint probability dis-

tribution of word pairs, $p(w_1, w_2)$.

3.1.1 Skip-Gram with Negative Sampling

SGNS iteratively trains word vectors for each co-occurring word pair in the same local context window (Mikolov et al., 2013b). The objective function for each word pair is as follows,

$$L_{SG}(i, j) = \log \sigma(\vec{u}_i^T \vec{v}_j) + \sum_{i=1}^k \mathbb{E}_{n \sim p(w)} [\log \sigma(-\vec{u}_i^T \vec{v}_n)]. \quad (1)$$

By minimizing (1), we get the global optimum point of SGNS as follows, Levy and Goldberg (2014b)

$$\vec{u}_i^T \vec{v}_j = PMI(i, j) - \log k,$$

where k is the number of negative samples.

One problem of SGNS which trains word vectors for each independent context window is that it cannot utilize the global statistics of a corpus (Pennington et al., 2014).

3.1.2 GloVe

To capture the global co-occurrence statistics of word pairs, Pennington et al. (2014) propose GloVe with the following objective function,

$$L_{GloVe} = \sum_{i,j \in D} f(i, j) (\vec{u}_i^T \vec{v}_j + \tilde{b}_i + \tilde{b}_j - \log X_{ij})^2,$$

which learns the word vectors that capture the global co-occurrences of word pairs (X_{ij}). Note that this model does not result in the optimum being the PMI statistic, which is the condition for additive compositionality described in section 4.

3.2 Sub-sampling

Word embedding models such as SGNS, FastText, and GloVe use various balancing methods that reduce the frequencies of very frequent words. These balancing methods are known to improve the semantic structure learned by a word embedding model. GloVe uses a clipping method to their weighting function that has an upper bound on the number of co-occurrences of word pairs. SGNS and FastText use a sub-sampling method that probabilistically discards frequent words during the learning procedure with the discard probability of word i as follows (Mikolov et al., 2013b),

$$p_s(i) = 1 - \left(\sqrt{\frac{s}{p(i)}} + \frac{s}{p(i)} \right). \quad (2)$$

Here, $p(i)$ is word frequency of word i in a corpus. s is a sub-sampling parameter. In this paper, we propose a statistical sub-sampling method that is based on sub-sampling method in SGNS. Our proposed method can be applied to a model that uses global statistics.

3.3 Additive Compositionality of Gittens et al. (2017)

Gittens et al. (2017) provide a mathematical definition of additive compositionality and a theoretical framework to justify the additive compositionality of Skip-Gram. They define additive compositionality by formulating a link between a paraphrase word and its context words, where the definition of paraphrase words is,

$$c = \arg \min_{w \in V} D_{KL}(p(\cdot|C)|p(\cdot|w)). \quad (3)$$

Here, c is a word and $C = \{c_1, \dots, c_n\}$ is a set of words. So, if a word c minimizes (3) for given set of words C , then we say the word c is a paraphrase of the word set, C . If paraphrase vector \vec{u}_c is captured by vector addition of words $\{c_1, \dots, c_n\}$, we call the word vectors, \vec{u} satisfy additive compositionality. Gittens et al. (2017) introduce two conditions of a paraphrase word vector c to be vector sum of a set C , $\vec{u}_c = \sum_{i=1}^n \vec{u}_{c_i}$.

- Given context word c , the probability that word w occurs within the same window can be calculated as follows,

$$p(w|c) = \frac{1}{Z_c} \exp(\vec{u}_w^T \vec{v}_c). \quad (4)$$

- Given the set of words C , the probability distribution of word w can be calculated as follows,

$$p(w|C) = \frac{p(w)^{1-m}}{Z_C} \prod_{i=1}^m p(w|c_i) \quad (5)$$

where Z_c and Z_C are normalization variables ($Z_c = \sum_{w \in V} \exp(\vec{u}_w^T \vec{v}_c)$, $Z_C = \sum_{w \in V} p(w)^{1-m} \prod_{i=1}^m p(w|c_i)$).

We are inspired by this mathematical definition of additive compositionality, and our work is based on their problem definition.

4 Additive Compositionality

To prove additive compositionality of word vectors, i.e., (4) and (5) are satisfied, Gittens et al.

(2017) assume uniform word frequency distribution, $p(w) = 1/|V|$. We know from Zipf’s law that this assumption is not realistic, so we propose to replace (4) with the following such that the word embedding model satisfies additive compositionality without the uniform distribution assumption:

$$p(w|c) = \frac{p(w)}{Z_c} \exp(\vec{u}_w^T \vec{v}_c) \quad (6)$$

where $Z_c = \sum_{w \in V} p(w) \exp(\vec{u}_w^T \vec{v}_c)$.

Theorem 1. *If a word embedding model satisfies (5) and (6), then the embedding vector of a paraphrase word, \vec{u}_c can be represented by the vector sum of its context word set, c_1, \dots, c_m , $\vec{u}_c = \sum_{i=1}^m \vec{u}_{c_i}$.*

Proof. The KL divergence (3) between (5) and (6) is computed as follows,

$$\begin{aligned} D_{\text{KL}} = & - \sum_{w \in V} p(w) \exp(\vec{u}_C^T \vec{v}_w) \{ (\vec{u}_c - \vec{u}_C)^T \vec{v}_w \\ & - \log \left(\sum_{w' \in V} p(w') \exp(\vec{u}_c^T \vec{v}_{w'}) \right) + \log Z \} \frac{1}{Z}. \end{aligned} \quad (7)$$

Here, $Z = Z_C \prod_{i=1}^m z_{c_i}$. $\vec{u}_C = \sum_{i=1}^m \vec{u}_{c_i}$. Differentiating (7) by \vec{u}_c , we get

$$\begin{aligned} \frac{\partial D_{\text{KL}}}{\partial \vec{u}_c} = & - \frac{1}{Z} \sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_C) \vec{v}_w \\ & + \frac{1}{Z} \sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c) \vec{v}_w \quad (8) \\ & \times \frac{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_C)}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)} \end{aligned}$$

Since (7) is convex and (8) becomes 0 when $\vec{u}_c = \vec{u}_C$, a paraphrase word vector can be represented by the vector sum of its context words set, C . We explain the details of the proof about convexity of (7) in the Appendix. \square

From (4) and (6), we can show that SGNS approximately satisfies additive compositionality without the uniform word frequency distribution assumption.

Theorem 2. *SGNS (Skip-Gram with Negative Sampling) approximately satisfies additive compositionality.*

Proof. The approximated global optimum point of SGNS is shifted-PMI (Levy and Goldberg,

Symbol	Description
V	Vocabulary
\mathcal{G}_i	Set of sub-words in word i
i, j	Index of a word in vocabulary
D	Set of co-occurring word pairs
σ	Sigmoid function
\vec{u}_i, \vec{v}_j	Embedding vector of word i, j
\vec{g}_i, \vec{h}_j	Embedding vector of sub-word i, j
k	Regularization parameter
s	Statistic sub-sampling parameter
S_{ij}	Regularization coefficient of word pair (i, j)

Table 1: Summary of the symbols

2014b). We can rewrite the approximated global optimum point as follows,

$$p(i|j) = \frac{p(i)}{k-1} \exp(\vec{u}_i^T \vec{v}_j). \quad (9)$$

Since (9) is the same as (6) and we can prove (5) with Bayes’ theorem, we prove that SGNS approximately satisfies additive compositionality. \square

5 Model

In this section, we describe our word embedding model OLIVE which satisfies additive compositionality described in section 4. We first describe our word level embedding model and its properties, then we expand the model to the sub-word level.

5.1 Word Level Embedding

5.1.1 Loss Function

The loss function in OLIVE consists of two parts: 1) a global co-occurrence term to capture the semantic similarity of co-occurring word pairs, and 2) a regularization term with sigmoid function and a different coefficient value for each word pair. Here, we use the regularization term to control the global optimal point of our model. Our proposed loss function is

$$\begin{aligned} L_{\text{OLIVE}} = & - \sum_{ij \in D} p(i, j) \log \sigma(\vec{u}_i^T \vec{v}_j) + \sum_{ij \in D} S_{ij} \sigma(\vec{u}_i^T \vec{v}_j). \end{aligned} \quad (10)$$

Here, D is the set of co-occurring word pairs, $p(i, j)$ is the probability that the words (i, j) occur in the same context window, \vec{u}, \vec{v} are the word

embedding vectors, and S_{ij} is the regularization coefficient for word pair (i, j) . The notations used in our model are summarized in Table 1. By minimizing this loss function, we get the word embedding vectors \vec{u} and \vec{v} .

5.1.2 Properties

The global optimum of (10) depends on the value S_{ij} . When S_{ij} is $p(i, j) + p(i)p(j)k$, our model has a single local optimum – the global optimum point which is the shifted PMI. Also, OLIVE satisfies additive compositionality when $S_{ij} = p(i, j) + p(i)p(j)k$. We describe the theorem and the proof below.

Theorem 3. *If $S_{ij} = p(i, j) + p(i)p(j)k$ and the dimension of word embedding vector \vec{u}, \vec{v} is sufficiently large to get global optimum point of L_{OLIVE} , then L_{OLIVE} has a unique local optimum with respect to $\vec{u}_i^T \vec{v}_j$, and $\vec{u}_i^T \vec{v}_j$ becomes $PMI(i, j) - \log k$ at the local optimum.*

Proof. Let $r = \vec{u}_i^T \vec{v}_j$. First derivative of (10) is as follows,

$$\begin{aligned} \frac{\partial L_{OLIVE}}{\partial r} \\ = p(i)p(j)k\sigma(-r)\sigma(r)e^{-r} \times (e^r - \frac{p(i, j)}{p(i)p(j)k}) \end{aligned} \quad (11)$$

Since $p(i)p(j)k\sigma(-r)\sigma(r)e^{-r}$ is always positive, $\frac{\partial L_{OLIVE}}{\partial r}$ is positive when r satisfies the following condition:

$$r \in (\log \frac{p(i, j)}{p(i)p(j)k}, \infty). \quad (12)$$

Second derivative of L_{OLIVE} is as follows,

$$\begin{aligned} \frac{\partial L_{OLIVE}}{\partial r^2} = -p(i)p(j)k\sigma(-r)\sigma(r)^2e^{-r} \\ \times (e^r - (2\frac{p(i, j)}{p(i)p(j)k} + 1)) \end{aligned} \quad (13)$$

Since $p(i)p(j)k\sigma(-r)\sigma(r)^2e^{-r}$ is always positive, $\frac{\partial L_{OLIVE}}{\partial r^2}$ is positive when r satisfies following condition:

$$r \in (-\infty, \log(\frac{2p(i, j)}{p(i)p(j)k} + 1)) \quad (14)$$

which leads to the following properties:

- (a) $\frac{\partial L_{OLIVE}}{\partial r^2} > 0$ if $r \in (-\infty, \log(\frac{2p(i, j)}{p(i)p(j)k} + 1))$
- (b) $\frac{\partial L_{OLIVE}}{\partial r} > 0$ if $r \in (\log(\frac{p(i, j)}{p(i)p(j)k}), \infty)$

Since $\log(\frac{2p(i, j)}{p(i)p(j)k} + 1) > \log(\frac{p(i, j)}{p(i)p(j)k})$, L_{OLIVE} has a unique local optimum with respect to $\vec{u}_i^T \vec{v}_j$. By simply finding r that makes (11) 0, we can show that the global optimum is $PMI(i, j) - \log k$. \square

Theorem 4. *If $S_{ij} = p(i, j) + p(i)p(j)k$, then L_{OLIVE} satisfies additive compositionality.*

Proof. From Theorem 1, we can prove Theorem 4 by showing that L_{OLIVE} satisfies (5) and (6).

- (a) We can simply prove (5) with Bayes' theorem.
- (b) From Theorem 3, we can rewrite the global optimum of our model as

$$p(i|j) = \frac{p(i)}{k-1} \exp(\vec{u}_i^T \vec{v}_j). \quad (15)$$

Since (15) is the same as (6), we prove that our model satisfies (6). \square

5.2 Sub-word Level Embedding

We can expand (10) to a sub-word level embedding model without losing the properties in section 5.1.2. Let $\vec{u}_i = \sum_{x \in \mathcal{G}_i} \vec{g}_x / |\mathcal{G}_i|$ and $\vec{v}_j = \sum_{y \in \mathcal{G}_j} \vec{h}_y / |\mathcal{G}_j|$. Then, the expanded sub-word level model can be defined by the loss function

$$\begin{aligned} L_{sub} = \\ - \sum_{ij \in D} p(i, j) \log \sigma((\frac{1}{|\mathcal{G}_i|} \sum_{x \in \mathcal{G}_i} \vec{g}_x)^T (\frac{1}{|\mathcal{G}_j|} \sum_{y \in \mathcal{G}_j} \vec{h}_y)) \\ + \sum_{ij \in D} S_{ij} \sigma((\frac{1}{|\mathcal{G}_i|} \sum_{x \in \mathcal{G}_i} \vec{g}_x)^T (\frac{1}{|\mathcal{G}_j|} \sum_{y \in \mathcal{G}_j} \vec{h}_y)), \end{aligned} \quad (16)$$

where x, y are sub-word indicators in words i, j . \vec{g}, \vec{h} are sub-word embedding vectors, and \mathcal{G}_i is the set of sub-words in word i .

5.3 Statistical Sub-sampling

Similar to SGNS and FastText, we apply sub-sampling to improve word embedding performance, but because SGNS sub-samples words in each iteration of the learning process, we cannot directly apply it to OLIVE which uses the global statistics. Instead we propose a statistical sub-sampling based on the same discard probability

form as (2) but considering for each word whether it needs to be sub-sampled, as follows:

$$p_s(i) = \begin{cases} \sqrt{\frac{s}{p(i)}} + \frac{s}{p(i)} & \text{if } p(i) > s \cdot (\frac{\sqrt{5}+1}{2})^2 \\ 1 & \text{otherwise} \end{cases} \quad (17)$$

Here, s is the sub-sampling parameter. By multiplying the above probability to the frequency of the word, we can get the sub-sampled global statistic of a word. The statistics can be calculated by

$$N_i^s = N_i \cdot p_s(i), \quad N_{i,j}^s = N_{i,j} \cdot p_s(i) \cdot p_s(j) \quad (18)$$

where N_i is the frequency of word i and $N_{i,j}$ is the co-occurrence frequency of word pair (i, j) .

5.4 Updating Rule

In the learning process of (10) and (16), the word vectors are updated by gradient descent, where the gradient of a word vector \vec{u}_i in (10) is

$$\begin{aligned} \frac{\partial L}{\partial \vec{u}_i} = & \sum_{j \in D_i} p(i)p(j)k\vec{v}_j\sigma(-\vec{u}_i^T \vec{v}_j)\sigma(\vec{u}_i^T \vec{v}_j) \\ & - \sum_{j \in D_i} p(i,j)\sigma(-\vec{u}_i^T \vec{v}_j)^2 \vec{v}_j. \end{aligned} \quad (19)$$

In (16), the gradient of the sub-word vector \vec{g}_x and \vec{h}_y can be simply calculated by (19) as

$$\frac{\partial L}{\partial \vec{g}_x} = \sum_{\{i|x \in \mathcal{G}_i\}} \frac{\partial L}{\partial \vec{u}_i}, \quad \frac{\partial L}{\partial \vec{h}_y} = \sum_{\{j|y \in \mathcal{G}_j\}} \frac{\partial L}{\partial \vec{v}_j}. \quad (20)$$

We update the word vectors with the normalized gradient of the word vector. The updating rule is

$$\vec{u}_i := \vec{u}_i - \eta \cdot \frac{\partial L}{\partial \vec{u}_i} \cdot \left| \frac{\partial L}{\partial \vec{u}_i} \right|^{-1} \quad (21)$$

where η is the learning rate.

6 Experiment

In this section, we conduct three experiments to show the empirical effects of theoretical improvement on OLIVE. First, we conduct an experiment on the word similarity task to verify the semantic representation performance of OLIVE. Second, we report the word similarity performance of OLIVE on various vocabulary sizes and noisy sentence representation performance to show the robustness of OLIVE.

6.1 Training Settings

We train our model and baseline models with the Wikipedia English corpus with 4 billion tokens. We preprocess the corpus with Matt Mahoney’s perl script². We use Skip-Gram, FastText, GloVe, and Probabilistic-FastText (Athiwaratkun et al., 2018) as baseline models. For all word and sub-word experiments, we set dimension = 300 and window size = 5. To train our word/sub-word model, we set $s = 10^{-5}$, $k = 50$, the number of iterations to 500, and the initial learning rate, η_0 to 0.5. For every iteration, we decrease the learning rate by the following formula:

$$\eta_t = \eta_0 \times \frac{1}{(t+1)^{0.25}} \quad (22)$$

In Skip-Gram and FastText, we set the number of negative samples to 5, the sub-sampling parameter to 10^{-5} and the initial learning rate to 0.025. In GloVe, we set the parameter x_max to 100 following their paper (Pennington et al., 2014). In Prob-FastText (Athiwaratkun et al., 2018), we set the parameters to the default settings in their code³. In the sub-word experiment, we extract sub-words whose length is in the range $[2, 7]$.

6.2 Word Similarity

We evaluate our word embedding performance with the word similarity task using three word similarity datasets: MTurk-(287, 771) (Radinsky et al., 2011), and SL-999 (Hill et al., 2015). We compare our model with four models: Skip-Gram, FastText, GloVe, and Probabilistic-FastText, a multisense sub-word embedding model (Athiwaratkun et al., 2018).

Table 3 shows the results of the word similarity experiment with words that occur two or more times in the corpus for a vocabulary size of 6.2 million words. Table 2 shows the results for the same experiment but with words that occur five or more times in the corpus for a vocabulary size of 2.8 million words. With both of these tables, we can see that OLIVE outperforms all four comparison models.

6.2.1 Effect of Vocabulary Size

When we compare tables 2 and 3 we see significant performance decreases in FastText, Skip-Gram, and Prob-FastText for the larger vocab-

²<http://mattmahoney.net/dc/textdata>

³<https://github.com/benathi/multisense-prob-fasttext>

	FastText	Skip-Gram	GloVe	Prob-FastText	OLIVE-sub	OLIVE-word
MTurk-287	64.73	66.30	60.62	66.49	65.69	66.54
MTurk-771	63.92	65.32	62.30	65.81	65.43	66.31
SL-999	35.09	36.01	32.53	36.54	34.86	36.18

Table 2: Spearman’s rank correlation coefficient of word similarity task on 2.8×10^6 vocabulary size.

	FastText	Skip-Gram	GloVe	Prob-FastText	OLIVE-sub	OLIVE-word
MTurk-287	62.46	63.38	60.77	63.17	64.33	67.22
MTurk-771	59.60	61.67	61.91	64.56	65.32	66.66
SL-999	33.04	34.20	31.98	33.83	34.99	35.81

Table 3: Spearman’s rank correlation coefficient of word similarity task on 6.2×10^6 vocabulary size.

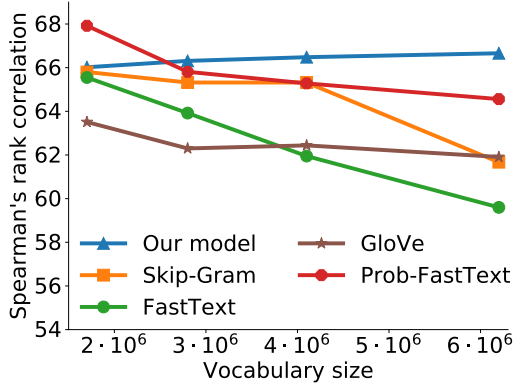


Figure 1: Spearman correlation of our model, Skip-Gram, and FastText on various vocabulary size in MTurk-771 testset.

ulary size. On the other hand, our model and GloVe show consistent performance, and we interpret that as a result of using the global word co-occurrence statistics. To visualize that more clearly, we plot in Figure 1 the word similarity scores for the MTurk-771 dataset on various vocabulary sizes for the different models. This figure clearly shows the robustness of our model with respect to the vocabulary size.

6.3 Noisy Sentence Representation

In sections 4 and 5.1.2, we have shown that our model satisfies additive compositionality and SGNS’s approximated additive compositionality. Theorem 1 implies equality between finding paraphrase vector of context words and finding sentence vector, which makes the sentence vector that minimizes the KL divergence (7) to $\vec{u}_{\text{sentence}} = \sum_{w \in \text{sentence}} \vec{u}_w$. To see robustness on additive compositionality of OLIVE that comes from removing the approximation in SGNS, we measure sentence similarity on various noise settings with

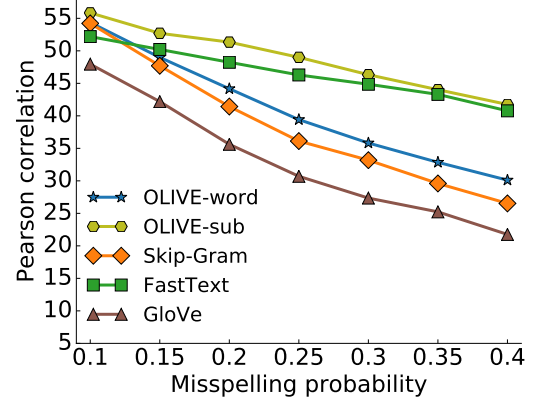


Figure 2: Pearson correlation of OLIVE, SGNS, GloVe, and FastText on SICK-dataset. We misspell words with probability between $[0.1, 0.4]$.

the sentence vectors calculated by the sum of word vectors in the sentence.

We use SICK dataset (Marelli et al., 2014) and all the STS-English dataset (Cer et al., 2017): STS-train, STS-dev, and STS-test. These datasets contain sentence pairs and human-annotated similarity scores for each sentence pair. To make noise in a sentence, we use two types of noise settings: typo and omitted-word. We use misspelling generation method proposed in Piktus et al. (2019) to make typos in a sentence with probability p . Piktus et al. (2019) use query logs of a search engine to build probabilistic distribution of misspelled words of a word, $p(\text{misspelled word}|\text{word})$. In omitted-word setting, we randomly discard words in each sentence with probability p . In both typo/omitted-word settings, we measure sentence similarity on various noise probability settings, $p \in [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4]$.

To compute the similarity of a pair of sentences, we first calculate the sentence vector and take the cosine similarity of the sentence vector pair. Then

		Word Vector Model			Sub-word Vector Model	
		SGNS	GloVe	OLIVE-word	FastText	OLIVE-sub
Typo	STS-Train	55.80/58.45	44.68/44.18	54.95/58.00	52.77/55.58	52.03/ 57.45
	STS-Dev	56.79/54.27	44.59/38.53	58.80/57.82	57.85/54.87	65.05/65.16
	STS-Test	46.21/48.07	34.88/34.65	42.22/45.21	52.30/52.89	54.35/57.10
	SICK	42.15/47.70	39.75/42.19	44.56/48.99	43.58/50.21	45.05/52.70
Omit	STS-Train	52.63/53.53	43.06/41.93	52.40/ 54.49	45.73/45.34	46.44/50.72
	STS-Dev	55.92/49.09	45.00/34.72	57.19/54.98	49.82/44.13	59.37/59.42
	STS-Test	42.57/42.20	31.65/31.23	39.14/40.89	44.83/42.05	47.28/49.35
	SICK	48.22/53.98	44.44/46.51	49.92/55.14	38.63/41.38	42.15/48.48

Table 4: Spearman/Pearson correlation of sentence similarity on typo/omitted-word setting. We misspell/discard words with probability 0.15. The left is Spearman correlation and the right is Pearson correlation.

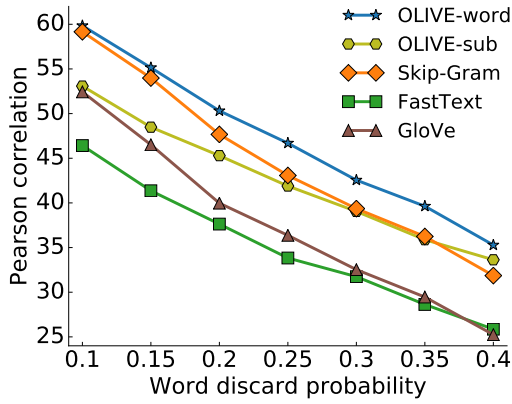


Figure 3: Pearson correlation of OLIVE, SGNS, GloVe, and FastText on SICK-dataset. We discard words with probability between $[0.1, 0.4]$.

we calculate the Spearman and Pearson correlations between the human-annotated similarity and the cosine similarity of the sentence vector pair. We report results on various typo/omitted-word setting in figure 2, 3 and results of typo/omit-word setting on 0.15 noise probability in table 4.

Table 4 show that OLIVE outperforms in each sub-word and word embedding group on both typo/omitted-word settings. When we compare the performance of GloVe, Skip-Gram, and OLIVE in table 4, we get an empirical evidence of approximate and exact additive compositionality in (2) and (4). Since GloVe does not theoretically satisfy additive compositionality, the correlation of GloVe is lower than Skip-Gram. Overall, we establish that our model captures noisy sentence representation better than Skip-Gram, GloVe, and FastText.

Since OLIVE-sub and FastText are character n-gram embedding models, $\vec{u}_i = \frac{1}{|\mathcal{G}_i|} \sum_{x \in \mathcal{G}_i} \vec{g}_x$, misspelling in a word tends to affect vector rep-

resentation of a word insignificantly. In figure 2, we empirically show significant performance difference between sub-word embedding models and word embedding models on typo setting. Also, figure 2 and 3 show that OLIVE outperforms in each sub-word and word embedding group on both typo/omit-word settings with various noise settings.

7 Conclusion

In this paper, we proposed 1) novel theoretical conditions of additive compositional word embedding model and 2) novel word/sub-word embedding model which we call OLIVE that satisfies additive compositionality. The loss function of OLIVE consists of a term for learning semantic similarity and a regularization term. From the loss function, we derived three properties of OLIVE: additive compositionality, uniqueness of local optimum, and shifted-PMI as the global optimum. Through several experiments, we showed OLIVE outperforms other existing embedding models on various word similarity task and showed robustness with respect to the size of the vocabulary. With sentence similarity task on various noisy settings, we showed robustness on additive compositionality of OLIVE.

Acknowledgments

This work was partly supported by NAVER Corp. and Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2017-0-01780, The technology development for event recognition/relational reasoning and

learning knowledge based system for video understanding).

References

- Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings. In *ICML*.
- Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.
- Ben Athiwaratkun, Andrew Wilson, and Anima Anandkumar. 2018. Probabilistic fasttext for multi-sense word embeddings. In *ACL*.
- Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5(1):135–146.
- Jan Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *ICML*.
- John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526.
- Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *SemEval*.
- Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. 2017. Skip-gram-zipf+ uniform= vector additivity. In *ACL*.
- Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(Feb):307–361.
- Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.
- Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *ACL*.
- Mikhail Khodak, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. 2018. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *ACL*.
- Omer Levy and Yoav Goldberg. 2014a. Linguistic regularities in sparse and explicit word representations. In *CoNLL*.
- Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *NIPS*.
- Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *SemEval*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*.
- Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *NAACL-HLT*.
- George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.
- Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *NIPS*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.
- Aleksandra Piktus, Necati Bora Edizel, Piotr Bojanowski, Edouard Grave, Rui Ferreira, and Fabrizio Silvestri. 2019. Misspelling oblivious word embeddings. In *NAACL*.
- Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *WWW*.
- Bei Shi, Zihao Fu, Lidong Bing, and Wai Lam. 2018. Learning domain-sensitive and sentiment-aware word embeddings. In *ACL*.
- Richard Socher, John Bauer, Christopher D Manning, et al. 2013. Parsing with compositional vector grammars. In *ACL*.
- Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP-CoNLL*.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint embedding of words and labels for text classification. In *ACL*.

Yang Xu, Jiawei Liu, Wei Yang, and Liusheng Huang. 2018. Incorporating latent meanings of morphological compositions to enhance word embeddings. In *ACL*.

A Convexity of (7)

The second derivative of D_{KL} is as follows,

$$\begin{aligned} & \frac{\partial^2 D_{KL}}{\partial \vec{u}_c \partial \vec{u}_c^T} \\ &= K \frac{\partial}{\partial \vec{u}_c} \frac{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c) \vec{v}_w}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)} \\ &= K \frac{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c) \vec{v}_w \vec{v}_w^T}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)} \\ &\quad - K \frac{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c) \vec{v}_w}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)} \\ &\quad \times \frac{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c) \vec{v}_w^T}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)}, \end{aligned}$$

where K is $\frac{1}{Z} \sum_{w \in V} p(w) \exp(\vec{v}_w^T (\sum_{i=1}^m \vec{u}_{c_i}))$.

To prove the second derivative of D_{KL} is positive definite matrix, we multiply non-zero vector \vec{x} to the Hessian matrix $(\frac{\partial^2 D_{KL}}{\partial \vec{u}_c \partial \vec{u}_c^T})$.

$$\begin{aligned} & \vec{x}^T \frac{\partial^2 D_{KL}}{\partial \vec{u}_c \partial \vec{u}_c^T} \vec{x} \\ &= K \frac{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c) \vec{x}^T \vec{v}_w \vec{v}_w^T \vec{x}}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)} \\ &\quad - K \frac{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c) \vec{x}^T \vec{v}_w}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)} \\ &\quad \times \frac{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c) \vec{v}_w^T \vec{x}}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)} \\ &= K \sum_{w \in V} r_w q_w^2 - K (\sum_{w \in V} r_w q_w)^2 \end{aligned} \tag{23}$$

Where,

$$r_w = \frac{p(w) \exp(\vec{v}_w^T \vec{u}_c)}{\sum_{w \in V} p(w) \exp(\vec{v}_w^T \vec{u}_c)}, \quad q_w = \vec{x}^T \vec{v}_w.$$

By Jensen's inequality,

$$K \sum_{w \in V} r_w q_w^2 > K (\sum_{w \in V} r_w q_w)^2.$$

Now, we get

$$\vec{x}^T \frac{\partial^2 D_{KL}}{\partial \vec{u}_c \partial \vec{u}_c^T} \vec{x} > 0$$

for all non-zero vector \vec{x} . So, the Hessian of D_{KL} is positive definite matrix which leads to convexity of the D_{KL} .