Engineering Applications of Artificial Intelligence 50 (2016) 256–264

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

# Modeling semantic compositionality of relational patterns

Sho Takase*, Naoaki Okazaki, Kentaro Inui

*Graduate School of Information Sciences, Tohoku University, 6-6-05 Aramaki Aza Aoba, Aobaku, Sendai, Miyagi, Japan*

ARTICLE INFO

*Article history:*
Received 25 August 2015
Received in revised form
11 December 2015
Accepted 12 January 2016
Available online 16 February 2016

*Keywords:*
Knowledge acquisition
Natural language processing
Relation extraction
Recursive neural network
Word embedding
Semantic compositionality

ABSTRACT

Vector representation is a common approach for expressing the meaning of a relational pattern. Most previous work obtained a vector of a relational pattern based on the distribution of its context words (e.g., arguments of the relational pattern), regarding the pattern as a single 'word'. However, this approach suffers from the data sparseness problem, because relational patterns are productive, i.e., produced by combinations of words. To address this problem, we propose a novel method for computing the meaning of a relational pattern based on the semantic compositionality of constituent words. We extend the Skip-gram model (Mikolov et al., 2013) to handle semantic compositions of relational patterns using recursive neural networks. The experimental results show the superiority of the proposed method for modeling the meanings of relational patterns, and demonstrate the contribution of this work to the task of relation extraction.

## 1. Introduction

Relation extraction is the task of extracting semantic relations between entities from corpora. This task is crucial for a number of NLP applications such as question answering and recognizing textual entailment. In this task, it is essential to identify the meaning of a *relational pattern* (a linguistic pattern connecting entities). Based on the distributional hypothesis (Harris, 1954), most previous studies construct a co-occurrence matrix between relational patterns (e.g., "*X* cause *Y*") and entity pairs (e.g., "*X*: smoking, *Y*: cancer"), and then they recognize relational patterns sharing the same meaning regarding the co-occurrence distribution as a semantic vector (Mohamed et al., 2011; Min et al., 2012; Nakashole et al., 2012). For example, we can find that the patterns "*X* cause *Y*" and "*X* increase the risk of *Y*" have the similar meaning because the patterns share many entity pairs (e.g., "*X*: smoking, *Y*: cancer"). Using semantic vectors, we can map a relational pattern such as "*X* cause *Y*" into a predefined semantic relation such as CAUSALITY only if we can compute the similarity between the semantic vector of the relational pattern and the prototype vector for the relation. In addition, we can discover relation types by clustering relational patterns based on semantic vectors.

However, this approach suffers from the data sparseness problem due to regarding a pattern as a 'word'. Fig. 1 shows the frequency and rank of relational patterns appearing in the ukWaC corpus (Baroni et al., 2009). The graph confirms that the distribution of occurrences of relational patterns follows Zipf's law. Here, we identify two critical problems. First, the quality of a semantic vector of a relational pattern may vary, because the frequency of occurrence of a relational pattern varies drastically. For example, the pattern "*X* cause *Y*" can obtain sufficiently many co-occurrence statistics (appearing more than $10^5$ times), while the pattern "*X* cause an increase in *Y*" cannot (appearing less than $10^2$ times). Second, we cannot compute semantic vectors of out-of-vocabulary patterns. We often discard less frequently occurring relational patterns, say, occurring fewer than $10^2$ times, even though we have no way of computing semantic vectors for the discarded or unseen patterns.

A natural approach to these problems is to compute the meaning of a relational pattern based on semantic compositionality, e.g., computing the vector for "*X* increase the risk of *Y*" from the constituent words (e.g, 'increase' and 'risk'). This treatment can be expected to improve the quality of semantic vectors, incorporating information of the constituent words into the semantic vectors of relational patterns. For example, we can infer that the relational pattern "*X* increase the risk of *Y*" has a meaning similar to that of "*X* increase the danger of *Y*" only if we know that the word 'risk' is similar to 'danger'.

Recently, there has been much progress in the methods for learning continuous vector representations of words (Bengio et al., 2003; Collobert and Weston, 2008; Mikolov et al., 2013). Among these methods, the Skip-gram model (Mikolov et al., 2013) received a fair amount of attention from the NLP community, because the model exhibits the additive compositionality

* Corresponding author.
*E-mail addresses:* takase@ecei.tohoku.ac.jp (S. Takase), okazaki@ecei.tohoku.ac.jp (N. Okazaki), inui@ecei.tohoku.ac.jp (K. Inui).

http://dx.doi.org/10.1016/j.engappai.2016.01.027

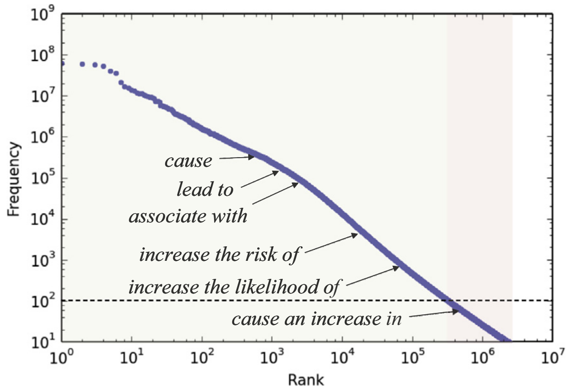


**Fig. 1.** The frequency of relational patterns in ukWaC.

exemplified by the famous example, $\boldsymbol{v}_{\text{king}} - \boldsymbol{v}_{\text{man}} + \boldsymbol{v}_{\text{woman}} \approx \boldsymbol{v}_{\text{queen}}$. Although we found a number of positive reports regarding additive compositionality, a linear combination of vectors is inadequate in some cases. For example, “*X* prevent the growth of *Y*” is dissimilar to “*X* grow *Y*” because ‘prevent’ negates the meaning of ‘grow’, but additive compositionality cannot handle the transformation. On the other hand, since “*X* have access to *Y*” has almost the same meaning as “*X* access *Y*”, we should not add the meaning of ‘have’ to that of ‘access’. For handling the verbs changing or inheriting the meaning, it is appropriate to apply a matrix because a matrix can transform (or inherit) a vector. In fact, Socher et al. (2012) proposed the recursive neural network (RNN) method that can handle a word changing the meaning by using matrices, but the method requires a certain amount of labeled data.

In this paper, we propose a novel method for modeling semantic vectors of relational patterns based on compositionality. More specifically, in addition to additive compositionality, we model the verbs that change or inherit the meaning by using RNN. We extend the Skip-gram model so that it can learn parameters for RNNs and semantic vectors of words from unlabeled data. In addition, we introduce $l_1$-regularization for training parameters of RNN to obtain a simpler model for semantic composition.

We conduct four kinds of experiments on the existing datasets, pattern similarity, relation extraction, and word similarity. The experimental results show that the proposed method can successfully model semantic compositions of relational patterns, outperforming strong baselines such as additive composition. The experiments also demonstrate the contribution of this work to the task of relation extraction. We confirm that the proposed method improves not only the quality of vectors for relational patterns but also that for words.

## 2. Proposed method

The proposed method bases on the Skip-gram model and RNN. Therefore, we first review the Skip-gram model in Section 2.1 and RNN in Section 2.2 followed by the proposed method.

### 2.1. Skip-gram model

Let $\mathcal{D}$ denote a corpus consisting of a sequence of words $w_1, w_2, \ldots, w_T$, and $V$ the set of words occurring in the corpus. The Skip-gram model minimizes the objective function,

$$J = -\sum_{w \in \mathcal{D}} \sum_{c \in C_w} \log\ p(c|w). \tag{1}$$

Here, $C_w$ is the set of context words for word $w$. $C_w = \{w_{-h}, \ldots, w_{-1}, w_{+1}, \ldots, w_{+h}\}$ ($h$ is a parameter that adjusts the width of contexts), where $w_{-p}$ and $w_{+p}$ represent the word appearing $p$ words before and after, respectively, the centered word $w$. The conditional probability $p(c|w)$ for predicting context word $c$ from word $w$, is formalized by a log-bilinear model,

$$p(c|w) = \frac{\exp(\boldsymbol{v}_w \cdot \tilde{\boldsymbol{v}}_c)}{\sum_{c' \in V} \exp(\boldsymbol{v}_w \cdot \tilde{\boldsymbol{v}}_{c'})}. \tag{2}$$

Here, $\boldsymbol{v}_w \in \mathbb{R}^d$ is the vector for word $w$, and $\tilde{\boldsymbol{v}}_c \in \mathbb{R}^d$ is the vector for context $c$. Training the log-bilinear model yields two kinds of vectors $\boldsymbol{v}$ and $\tilde{\boldsymbol{v}}$, but we use only $\boldsymbol{v}$ as semantic vectors of words (word vectors). Because computing the denominator in Eq. (2), the sum of the dot products for all the words in the corpus, is intractable, Mikolov et al. (2013) proposed the negative sampling method based on noise contrastive estimation (Gutmann and Hyvärinen, 2012). The negative sampling method trains logistic regression models to be able to discriminate an observed context word $c$ from $k$ noise samples (pseudo-negative words $z$).

$$\log\ p(c|w) \approx \log \sigma(\boldsymbol{v}_w \cdot \tilde{\boldsymbol{v}}_c) + k\, \mathbb{E}_{z \sim P_n}\left[\log \sigma(-\boldsymbol{v}_w \cdot \tilde{\boldsymbol{v}}_z)\right] \tag{3}$$

Here, $P_n$ is the probability distribution for sampling noise words. In this study, we used the probability distribution of unigrams raised to the 3/4 power (Mikolov et al., 2013).

### 2.2. Recursive neural network (RNN)

Recursive neural network computes the semantic vectors of phrases based on compositionality (Socher et al., 2011b). Using a weight matrix $M \in \mathbb{R}^{d \times 2d}$ and an activation function $g$ (e.g., tanh), RNN computes the semantic vector of the phrase consisting of two words $w_a$ and $w_b$,

$$g\left(M \begin{bmatrix} \boldsymbol{v}_{w_a} \\ \boldsymbol{v}_{w_b} \end{bmatrix}\right). \tag{4}$$

The vector computed by Eq. (4) is expected to represent the meaning of the phrase based on semantic compositionality. Socher et al. (2011b) apply this function recursively inside a binarized parse tree, and compose the semantic vectors of phrases and sentences. Although the study modeled only one compositional function with a single matrix $M$, Socher et al. (2012) extended RNN to matrix-vector RNN (MV-RNN) in order to configure a compositional function for each word, assigning a word with both a vector and a matrix.

### 2.3. Semantic composition for relational patterns

We extend the Skip-gram model to enable it to take into account the semantic composition for relational patterns. We provide an overview of the proposed method using the example in Fig. 2. Here, we have a sequence of lemmatized words “yeast help reduce the serious risk of infection”. As explained in Section 1, it is inefficient to regard the relational pattern “*X* help reduce the serious risk of *Y*” as a single ‘word’ (upper). Instead, we compute the semantic vector from the constituent words of the relational pattern, e.g., ‘help’, ‘reduce’, ‘serious’, and ‘risk’. Simultaneously, we would like to handle cases in which words have a major influence on changing the meaning of the entire phrase.

Inspired by Socher et al. (2012), we represent the words inheriting or changing the meaning with matrices in RNN. In this paper, we assume that verbs appearing frequently in relational patterns may inherit or change the meaning computed by other constituent words. We call these verbs *transformational verbs*.[1] In the example in Fig. 2, we may think that ‘reduce’ changes the

[1] Transformational verbs are similar to *light verbs* and *catenative verbs*, but it is hard to give a formal definition.

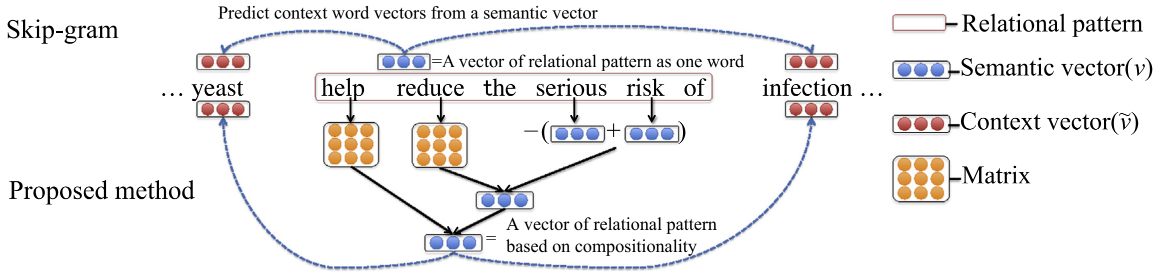

Skip-gram
Predict context word vectors from a semantic vector
=A vector of relational pattern as one word
… yeast
help reduce the serious risk of
infection …
−( + )
Proposed method
A vector of relational pattern based on compositionality
=
Relational pattern
Semantic vector(v)
Context vector(ṽ)
Matrix

**Fig. 2.** Overview of the proposed method. The original Skip-gram model is illustrated on the upper level.

meaning of ‘risk’ and ‘help’ inherits the meaning of “reduce the serious risk of”; and the change and inheritance are represented by matrices.

To compute the semantic vector for the relational pattern “*X* help reduce the serious risk of *Y*”, the proposed method first computes the semantic vector for “the serious risk of”. In this study, we assume that additive compositionality for words except for transformational verbs. For this reason, the proposed method obtains the semantic vector for “the serious risk of” by computing the mean of the semantic vectors of ‘serious’ and ‘risk’. Next, the proposed method multiplies the semantic vector for “the serious risk of” and the matrix for ‘reduce’, and then multiplies the computed vector and the matrix for ‘help’. To learn the parameters in matrices and vectors, we incorporate the RNN framework into the Skip-gram model. This is not only because the Skip-gram model achieved successes in training high-quality word vectors from a large corpus, but also because the online training algorithm (word-by-word) is suitable for incorporating matrix-vector compositions used in RNN.

Meanwhile, giving a formal definition of transformational verbs is arguable. In this study, we make three assumptions for identifying transformational verbs:

1. Verbs can behave as transformational verbs.
2. Whether a verb is a transformational verb or not is determined by the statistics of its occurrences in relational patterns.
3. Other words, e.g, nouns, adjectives, adverbs, and verbs not qualified to be transformational verbs express meanings of their own. We call these words *content words*.

Although these assumptions are rather provisional, we would like to explore the possibility of semantic compositionality for relational patterns.

We assume that the relational pattern $P$ is composed of transformational verbs $p_1, \dots, p_n$, followed by content words $p_{n+1}, \dots, p_m$. For example, the lemmatized relational pattern “help reduce the serious risk of” is composed of the transformational verbs ‘help’ and ‘reduce’ as well as ‘the’, ‘serious’, ‘risk’, ‘of’, as shown in Fig. 2. Removing non-content words such as determiners and prepositions, we obtain content words ‘serious’ and ‘risk’. Accordingly, the relational pattern “help reduce the serious risk of” is represented by $P = (p_1, p_2, p_3, p_4) = (\text{help}, \text{reduce}, \text{serious}, \text{risk})$. The total number of words in $P$ is $m = 4$, and the boundary between the transformational verbs and content words is $n = 2$. Although formal definitions of relational pattern, transformational verbs, and content words are open questions, we mine them from the corpus (refer to Section 3.1).

As previously mentioned, in this study, we assume that additive compositionality for content words in a relational pattern. That is to say, the meaning of content words $p_{n+1}, \dots, p_m$ is computed from the mean of the semantic vectors corresponding to the content words,

$$\frac{\boldsymbol{v}_{p_{(n+1)}} + \boldsymbol{v}_{p_{(n+2)}} + \dots \boldsymbol{v}_{p_m}}{m - n}. \tag{5}$$

In contrast, we assume that each transformational verb $p_i$ inherits or transforms a given semantic vector using the mapping function, $f_{p_i} : \mathbb{R}^d \rightarrow \mathbb{R}^d$. Hence, the semantic vector $\boldsymbol{v}_P$ for the relational pattern $P$ is computed as,

$$\boldsymbol{v}_P = f_{p_1}\left(f_{p_2}\left(\dots f_{p_n}\left(\tanh\left(\frac{\boldsymbol{v}_{p_{(n+1)}} + \boldsymbol{v}_{p_{(n+2)}} + \dots \boldsymbol{v}_{p_m}}{m - n}\right)\right)\right)\right). \tag{6}$$

We design the mapping function $f_{p_i}$ using RNN (Socher et al., 2011b). More specifically, the mapping function for the transformational verb $p_i$ is modeled using a matrix $W_i \in \mathbb{R}^{d \times d}$ and an activation function.

$$f_{p_i}(\boldsymbol{v}) = \tanh(W_{p_i}\boldsymbol{v}) \tag{7}$$

In short, the proposed method computes the meaning of content words of a relational pattern as the vector mean, and inherits/transforms the meaning using matrix-vector products of RNN.

### 2.4. *Training*

The proposed method is identical to the Skip-gram model when a context window involves no relational pattern. In other words, we train $\boldsymbol{v}_w$ and $\tilde{\boldsymbol{v}}_c$ in the same manner as the original Skip-gram model with negative sampling. We summarize the differences from the original Skip-gram model:

1. We treat a relational pattern as a ‘word’, but its semantic vector is computed using Formula 6. We update the vectors for content words and matrices for transformational verbs to enable the composed vector of the relational pattern $P$ to predict context words $c$ well.
2. In addition to word vectors $\boldsymbol{v}$ and $\tilde{\boldsymbol{v}}$, we train semantic matrices $W$ for the transformational verbs. We use backpropagation for updating vectors and matrices.
3. We do not use Formula 6 for computing a context vector of a relational pattern. In other words, when the negative sampling picks a relational pattern for a centered word, we use a context vector $\tilde{\boldsymbol{v}}$ assigned for the pattern.
4. We apply the activation function tanh even for word vectors $\boldsymbol{v}$. This keeps the value range of semantic vectors consistent between composed vectors and word vectors. Each dimension of a semantic vector of a relational pattern is bound to the range of $(-1, 1)$, because Formula 6 uses tanh as an activation function.

Meanwhile, some transformational verbs (e.g., light verbs) may not contribute to meanings. For example, the word ‘take’ in the pattern “take care of” does not have a strong influence on the meaning of the pattern. Thus, we explore the use of $l_1$-regularization to encourage diagonal matrices. We modify the objective function (Eq. (1)) into:

$$J' = -\sum_{w \in \mathcal{D}} \sum_{c \in C_w} \log\ p(c|w) + \lambda \sum_{W \in \mathbb{W}} r(W). \tag{8}$$

Here, $\mathbb{W}$ represents the set of all matrices for the transformational verbs. The function $r(W)$ computes the $l_1$-norm from off-diagonal elements of $W$,

$$r(W) = \sum_{i \neq j} |W_{i,j}|. \tag{9}$$

## 3. Experiments

### 3.1. Corpora and training settings

We used ukWaC[2] as the corpus for training the semantic vectors and matrices. This corpus includes the text of Web pages crawled from the .uk domain, and contains 2 billion words. This corpus also includes parts-of-speech tags and lemmas annotated by the TreeTagger.[3] In our experiment, we lowercased words and used the lemmas except for past participle forms of verbs (we used their surface forms).[4] Furthermore, tokens consisting of a single character (e.g., 'a' and 'b'), determiners (e.g., 'the'), interrogative words (e.g., 'what'), and prepositions were removed as stop words.

We applied Reverb (Fader et al., 2011) to the ukWaC corpus to extract relational pattern candidates. To remove unuseful relational patterns, we followed the filtering rules that are compatible with the ones used in the publicly available extraction result[5]: the confidence score of a pattern must be no less than 0.9 at least once in the corpus, a relational pattern must not contain a temporal expression (e.g., 'yesterday' and 'tonight'), and the frequency of occurrence of a pattern must be no less than 5. Additionally, throughout the experiments, we removed relational patterns that appear in the evaluation data in order to examine the performance of the proposed method in composing semantic vectors of unseen relational patterns. After the above preprocessing, we obtained 55,885 relational patterns.

Verbs appearing in five or more kinds of relational patterns were identified as transformational verbs. We removed the verb 'be' in this experiment. Using the criterion, we identified 697 verbs as transformational verbs in relational patterns. If a relational pattern consists only of transformational verbs, we regarded the last word as a content word. While there may be some room for consideration regarding the definition of transformational verbs and content words, we used these criteria in this experiment.

When training the proposed method, we removed words and relational patterns appearing less than 10 times. As a result, we obtained approximately 0.7 million words (including relational patterns) as targets for training semantic vectors. When a transformational verb appears outside of a relational pattern (e.g., 'reduce'), we update a vector for the word in the same way as for an ordinary word.

For comparing with the existing methods, we used the same hyper-parameters as the ones presented in the papers (Socher et al., 2012; Mikolov et al., 2013). We set the number of dimensions $d = 50$, following Socher et al. (2012). For the width of context window $h$, number of negative samples $k$, and subsampling parameter in the Skip-gram, we used the same hyper-parameters as in Mikolov et al. (2013): $h = 5$, $k = 5$, and subsampling with $10^{-5}$. We initialize word vectors $\boldsymbol{v}$ and context vectors $\tilde{\boldsymbol{v}}$ using the result from the original Skip-gram model (Mikolov et al., 2013). Elements in semantic matrices $W$ are initialized with random values sampled from a Gaussian distribution with mean 0 and variance 0.1. We learn parameters ($\boldsymbol{v}$, $\tilde{\boldsymbol{v}}$, and $W$) by the backpropagation with the stochastic gradient descent (SGD) method. We control the learning rate $\alpha$ for an instance by using the formula implemented in word2vec[6]:

$$\alpha = \alpha_0 * \left(1 - \frac{\text{the number of processed sentences}}{\text{the number of total sentences} + 1}\right). \tag{10}$$

In Eq. (10), $\alpha_0$ represents the initial learning rate (0.025 in this experiments). Eq. (10) decreases the learning rate steadily according to the number of processed sentences.

### 3.2. Evaluation datasets

We conducted three experiments, pattern similarity, relation extraction, and word similarity.

*Pattern similarity*: We would like to examine whether our proposed method can successfully compose semantic vectors of relational patterns. The performance of a method can be measured by the correlation between similarity judgments of humans for relational patterns and the similarities of the corresponding semantic vectors computed by the method. However, unfortunately, no existing dataset provides similarity judgements between relational patterns. Instead, we adapted the dataset developed for semantic inferences between relational patterns (Zeichner et al., 2012).[7] Using relational patterns extracted by Reverb, this dataset labels whether a pair of relational patterns (e.g., '*X* prevent *Y*' and "*X* reduce the risk of *Y*") is meaningful[8] or not. A meaningful pair is annotated with a label indicating whether the pair has an inference relation (entailment). The dataset consists of 6567 pairs overall.

After discarding pairs labeled meaningless and cases where the set of arguments is reversed between paired patterns such as "*X* contain embedded *Y*" and "*Y* be embedded within *X*", we extracted 5409 pairs for evaluation. The evaluation dataset includes 2447 pairs with inference relation (similar), 2962 pairs without inference relation (dissimilar). This dataset includes only binary decisions (similar or dissimilar) for relational patterns, whereas similarity values computed by a method range in [0, 1.0]. Thus, we regard pattern pairs having similarity values greater than a threshold as 'similar', and the rest as 'dissimilar'. In this way, we can measure the precision and recall of a method for detecting similar relational patterns with the given threshold. By changing the threshold from 0.0 to 1.0, we can draw a precision-recall curve for each method.

*Relation extraction*: To examine the contribution of this work to the relation extraction task, we used the SemEval-2010 Task 8 dataset (Hendrickx et al., 2010). The task is to identify the relationship of a given entity pair. The dataset consists of 10,717 relation instances (8000 training and 2717 test instances), each of which is annotated with a relation label. The data set has 19 candidate relation labels, nine directed relationships (e.g., Cause-Effect) and one undirected relationship Other. For example, the entity pair 'burst' and 'pressure' in the sentence "The burst has

[2] http://wacky.sslmit.unibo.it/doku.php?id=corpora

[3] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

[4] We use past particle forms to express the passive/active voice.

[5] http://reverb.cs.washington.edu/

[6] https://code.google.com/p/word2vec/

[7] http://u.cs.biu.ac.il/~nlp/resources/downloads/annotation-of-rule-applications/

[8] When an annotator judges a pair, the slots of relational patterns are filled with the same subject and object. If the annotator can easily understand the both of expressions, the pair is meaningful. Take the pair "*X* belong to *Y*" and "*X* be property of *Y*" as an example. If the pair is filled with "Such people" and "the left", it is unrealistic to understand the meaning of "Such people be property of the left". In this case, the pair is annotated with meaningless.

been caused by water hammer pressure" is labeled as Cause-Effect $(e_1, e_2)$.

*Word similarity*: We also evaluated the word vectors to verify that the proposed method does not degrade the quality of word vectors. We used a variety of word similarity datasets: WordSim-353 (Finkelstein et al., 2001), MC (Miller and Charles, 1991), RG (Rubenstein and Goodenough, 1965), and SCWS (Huang et al., 2012). For each dataset, the numbers of word pairs are 353, 30, 65, and 2003. For evaluation, we used all word pairs included in the datasets after lowercasing and lemmatizing similarly to the training procedure. We calculate Spearman's rank correlation coefficients between human judgments and cosine similarity values of the semantic vectors computed by each method.

### 3.3. Results

*Pattern similarity*: Fig. 3 shows precision-recall curves of the proposed method and baseline methods on the pattern-similarity task. The red locus shows the performance of the proposed method. In this figure, we set the parameter for $l_1$-regularization to $10^6$, because the parameter achieved the best performance (Table 1). The blue locus corresponds to the Skip-gram model, in which relational patterns are regarded as single words. This treatment is identical to the procedure for training phrase vectors in Mikolov et al. (2013). The green locus shows the performance of additive compositions of word vectors trained by the Skip-gram model. In this method, we trained word vectors as usual (without considering relational patterns), and computed vectors of relational patterns as the mean of vectors of constituent words. This treatment is the popular and strong baseline method to compute a phrase vector from its constituent words (Muraoka et al., 2014). The light blue locus reports the performance when we fix matrices for transformational verbs as identity matrices: this corresponds to ignoring transformational verbs in a relational pattern. The yellow and purple loci correspond to assigning every verb (yellow) and every word (purple), respectively, with a matrix (rather than a vector). In these settings, we use a vector representation for a content word that are located at the end of relational patterns. In other words, these settings are more flexible than the recommended setting (red) for composing semantic vectors, having more free parameters to train the models.

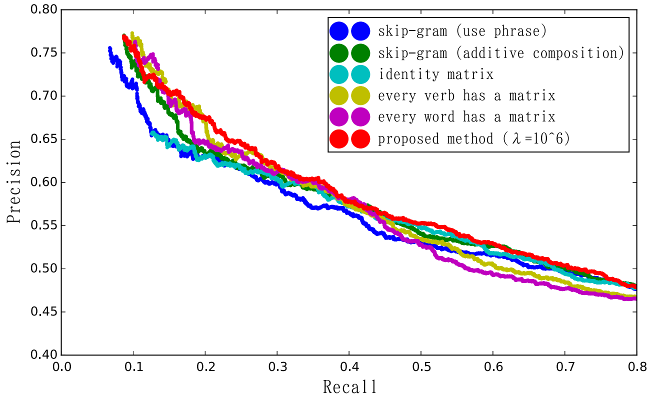


**Fig. 3.** Precision-recall curve of each method on the pattern-similarity task.

**Table 1**
Area under the curve (AUC) of each method on the pattern-similarity task. This table also reports the sparsity (the ratio of zero-elements) of matrices with different parameters for $l_1$-regularization. If a method outperforms the Skip-gram (additive) with 95% statistical significance ($p < 0.05$), we put [a] on the value of AUC.

| Method | AUC | Sparsity |
|---|---|---|
| Skip-gram (phrase) | 0.557 | – |
| Skip-gram (additive) | 0.568 | – |
| Identity matrix | 0.552 | – |
| Every verb has a matrix | 0.566 | – |
| Every word has a matrix | 0.561 | – |
| Proposed ($\lambda = 0$) | 0.570 | 0.0% |
| Proposed ($\lambda = 1$) | 0.570 | 0.0% |
| Proposed ($\lambda = 10$) | 0.570 | 0.7% |
| Proposed ($\lambda = 10^2$) | 0.573 | 14.4% |
| Proposed ($\lambda = 10^3$) | 0.574[a] | 54.4% |
| Proposed ($\lambda = 10^4$) | 0.575[a] | 88.2% |
| **Proposed**($\lambda = 10^5$) | **0.576**[a] | 96.6% |
| **Proposed**($\lambda = 10^6$) | **0.576**[a] | 97.8% |
| Proposed ($\lambda = 10^7$) | 0.575[a] | 98.0% |

Fig. 3 shows that the proposed method performed better than all baseline methods. It is noteworthy that the proposed method performed better than the Skip-gram model with additive compositionality in green, which has been regarded as a strong baseline for semantic composition. This result indicates that representing transformational verbs with matrices in RNN is more suitable than additive composition for computing the semantic vectors of relational patterns.

The proposed method outperformed the setting with transformational verbs ignored (light blue). This result indicates that the treatment for transformational verbs is important for composing semantic vectors of relational patterns. In fact, the proposed method successfully computes semantic vectors of relational patterns with inhibitory verbs: for example, the proposed method could predict the similarity between "prevent the growth of" and 'inhibit' while the use of identity matrices cannot.

The comparison among the proposed method (red), representing every verb with a matrix (yellow), and representing every word with a matrix (purple) demonstrates the effectiveness of identifying transformational verbs in advance. This result suggests that transformational verbs (verbs appearing frequently in relational patterns) can inherit/change the meaning and that it is important to incorporate their behaviors in composing semantic vectors of relational patterns.

Training semantic vectors of relational patterns by regarding a relation pattern as a single word (blue) performed worse than most of other methods in this experiment. This suggests the difficulty in learning vector representations of relational patterns only with the distributional hypothesis. The incorporation of the distributional hypothesis with the semantic compositionality is the key to success in modeling semantic vectors of relational patterns.

Table 1 shows the area under the curve (AUC) of each method appearing in Fig. 3. In addition to AUC values, we report the sparsity (the percentage of zero elements in matrices), changing the parameter for $l_1$-regularization from 0 to $10^7$ in powers of ten.[9] Again, we can reconfirm from this table that the proposed method outperforms the baseline methods. Moreover, the methods with $\lambda = 10^3, 10^4, 10^5, 10^6$, and $10^7$ outperform the strong baseline, Skip-gram (additive) with 95% statistical significance ($p < 0.05$) measured by paired bootstrap resampling (Koehn, 2004). The proposed method obtained the best performance (0.576) with over 95% sparsity ($\lambda = 10^5$ and $10^6$). The results indicate that the use of $l_1$-regularization for off-diagonal elements of matrices improves the performance even though the obtained model becomes compact. However, the model with $\lambda = 10^7$ was too sparse to achieve the best performance.

[9] We stopped increasing $\lambda$ to $10^7$ because the AUC decreased when we changed $\lambda$ from $10^6$ to $10^7$.

**Table 2**
Comparison of using the proposed method with previously published results.

| Method | Features | F1 |
|---|---|---|
| SVM | Basic features | 76.0 |
| (Use semantic vectors obtained by the proposed method) | Basic features, semantic vectors | 79.0 |
| | Basic features, WordNet, NE | 79.9 |
| | Basic features, semantic vectors, WordNet, NE | 82.1 |
| SVM (Best in SemEval 2010) (Rink and Harabagiu, 2010) | POS, prefixes, morphological, WordNet, dependency parse, Levin classed, ProBank, FrameNet, NomLex-Plus, Google n-gram, paraphrases, TextRunner | 82.2 |
| RNN | – | 74.8 |
| (Socher et al., 2011b) | WordNet, NE | 77.6 |
| MV-RNN | - | 79.1 |
| (Socher et al., 2012) | WordNet, NE | 82.4 |
| CNN (Zeng et al., 2014) | WordNet | 82.7 |
| FCM | - | 80.6 |
| (Yu et al., 2014) | Dependency parse, NE | 83.0 |
| CR-CNN (dos Santos et al., 2015) | – | 84.1 |
| RelEmb | - | 82.8 |
| (Hashimoto et al., 2015) | Dependency parse, WordNet, NE | 83.5 |
| depLCNN+NS | – | 84.0 |
| (Xu et al., 2015) | WordNet | 85.6 |

Table 1 also shows that representing every verb with a matrix or representing every word with a matrix performed worse than Skip-gram (additive). This also suggests that it is essential to distinguish transformational verbs from content words. In addition, representing content words with matrices gave too much flexibility for this task.

*Relation extraction*: Table 2 shows the performance of each method on the relation extraction task. The top 4 rows represent the results of the baseline method and improvements by using semantic vectors computed by the proposed method. To predict whether a given entity pair has a specific relation, we built one-versus-one classifiers modeled by SVM with radial basis function (RBF) kernel. We defined basic features for the classifiers: parts-of-speech tags, surface forms, and lemmas of words appearing between an entity pair, and lemmas of the words in the entity pair. In addition, we included the value of each dimension of the semantic vectors of a relational pattern and entity pairs as features in order to examine the effect of the semantic vectors obtained by the proposed method with ($\lambda = 10^6$). Moreover, we employed named entity information and WordNet super sense classes predicted by a super sense tagger (Ciaramita and Altun, 2006). We used libsvm[10] for training SVM models. For hyper-parameters, we determined $C = 8.0$ and $\gamma = 0.03125$ based on 5-fold cross-validation.

Table 2 shows that the use of semantic vectors of the proposed method boosted the performance from 76.0 to 79.0 F1 scores. Moreover, even with the external knowledge (WordNet super sense), semantic vectors computed by the proposed method improved the performance from 79.9 to 82.1. This demonstrates the usefulness of the semantic vectors computed by the proposed method for the task of relation extraction.

For comparison, Table 2 includes the performance reported in the previous work. Table 2 shows that using our semantic vectors exhibited performance closed to the best method in the SemEval-2010 task 8 competition (Rink and Harabagiu, 2010). The proposed method outperformed RNN (Socher et al., 2011b) by a large margin. Moreover, the proposed method achieved a comparable performance with MV-RNN (Socher et al., 2012).

**Table 3**
Spearman's rank correlation coefficients on the word similarity tasks.

| Method | WS353 | MC | RG | SCWS |
|---|---|---|---|---|
| Baseline (Skip-gram without relational patterns) | 63.0 | 69.5 | 74.2 | 60.3 |
| Proposed ($\lambda = 10^6$) | **68.4** | **73.7** | **75.4** | **61.5** |

However, the best result obtained by the proposed method was lower than that of the state-of-the-art methods. These methods specially train vector representations for words such that it predicts predefined relation labels in the SemEval-2010 Task 8 dataset. In other words, they fine-tuned vector representations for this task. In fact, Yu et al. (2014) reported that fine-tuning improved the performance. In addition, Hashimoto et al. (2015) indicated that they achieved a better performance when they refined vector representations for initialization. Therefore, we may obtain a further improvement if we tune matrices and vectors in our model specialized for the SemEval-2010 Task 8. In contrast, our focus is to model semantic composition of relational patterns in a generic and unsupervised fashion. We will explore the possibility of fine-tuning in future work.

*Word similarity*: Table 3 reports the results for word similarity on four different datasets. In this task, we compared the proposed method with the Skip-gram model ignoring relational patterns. Table 3 shows that the proposed method yielded the better performance than the Skip-gram model without relational patterns. In other words, the result indicates that our approach also improved the quality of the semantic vectors of words.

### 3.4. Visualizing the matrices

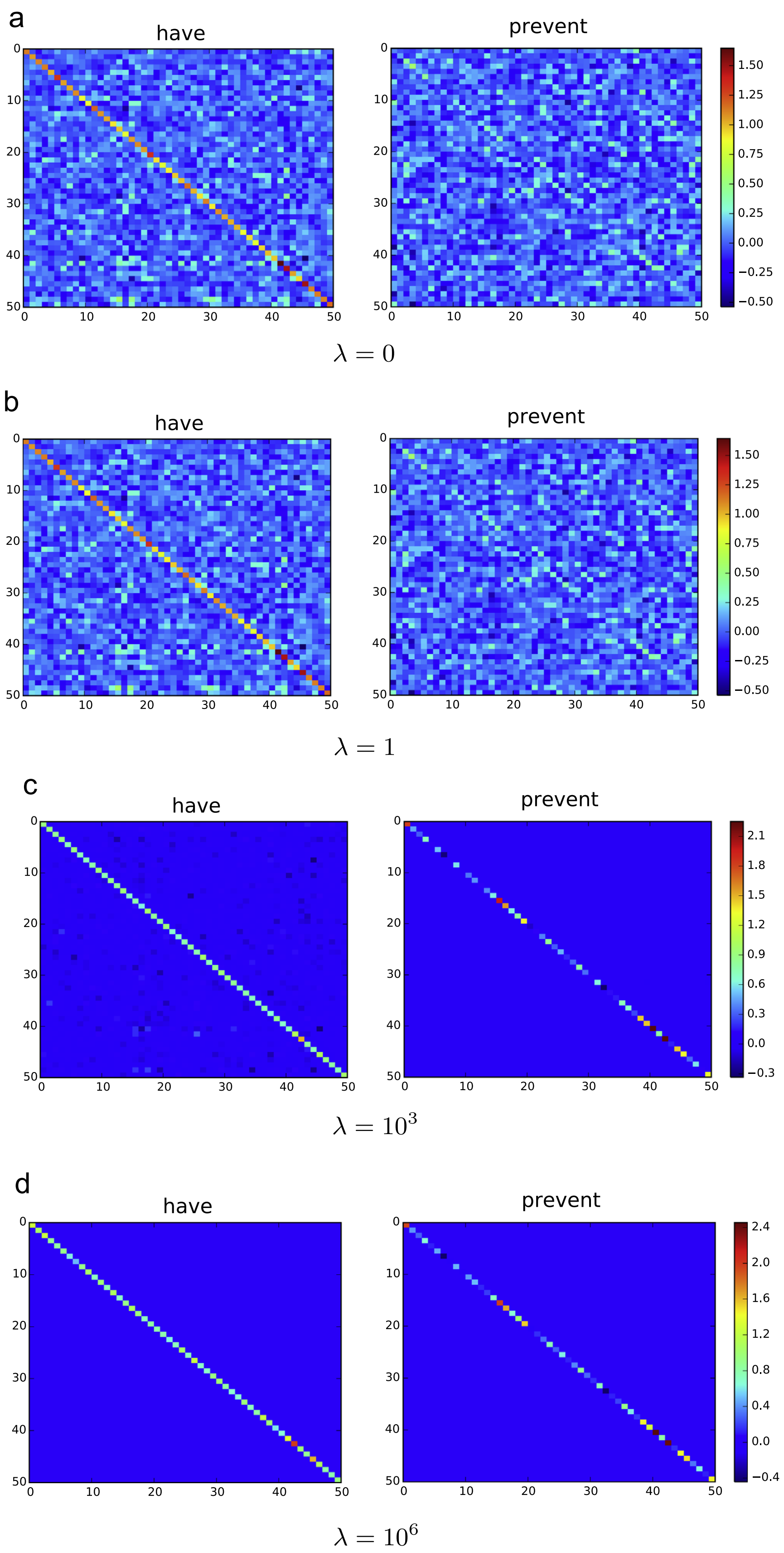
Fig. 4 shows a visualization of matrices for the words ‘have’ and ‘prevent’ learned by the proposed method with different parameters for $l_1$-regularization ($\lambda = 0, 1, 10^3$, and $10^6$). The values of the diagonal elements in the matrix for ‘have’ are high while the off-diagonal elements are close to zero. In other words, the matrix for ‘have’ is close to the identity matrix, implying that the word ‘have’ inherits the meaning from content words. The proposed method learned this behavior because a number of relational patterns (e.g., “have access to” and “have an impact on”) include the word ‘have’, but their contexts are similar to those for content words (e.g., ‘access’ and ‘impact’). We could observe the similar tendency for verbs such as ‘make’ and ‘take’.

In contrast, the matrix for ‘prevent’ is entirely different from that for ‘have’. With the small $l_1$-regularization parameters ($\lambda = 0$ and 1), the matrix for ‘prevent’ does not have an obvious tendency. With the large $l_1$-regularization parameters ($\lambda = 10^3$ and $10^6$), the matrix is close to the diagonal matrix. However, the matrix is different from the identity matrix: each diagonal element have a non-uniform value. This is probably because the word ‘prevent’ tends to negate the meaning of content words, as in “prevent the growth of”. Thus, the proposed method found a matrix so that it does not pass the meaning of the content word (e.g., ‘growth’) directly to that of the whole.

## 4. Related work

*Relation extraction*: A number of previous studies extracted semantic relations between entities using linguistic patterns (Pantel and Pennacchiotti, 2006; Rosenfeld and Feldman, 2007; Carlson et al., 2010; Min et al., 2012; Nakashole et al., 2012). These studies mostly explored methods for obtaining relation instances

[10] https://www.csie.ntu.edu.tw/~cjlin/libsvm/



$\lambda = 0$

$\lambda = 1$

$\lambda = 10^3$

$\lambda = 10^6$

**Fig. 4.** Examples of the matrices learned using the proposed method ($\lambda = 0, 1, 10^3$, and $10^6$).

with high-precision e.g., using pointwise mutual information between entity pairs and patterns (Pantel and Pennacchiotti, 2006), checking types of arguments of relational patterns (Rosenfeld and Feldman, 2007), and extracting entities and relation instances simultaneously (Carlson et al., 2010). On the other hand, Min et al. (2012) improved recall by incorporating various knowledge sources into the extracting algorithm. However, these approaches suffer from the data sparseness problem described in Section 1.

Nakashole et al. (2012) presented *PATTY*, a large resource for relational patterns. PATTY has an automatic method for inducing rules for generalizing relational patterns with part-of-speech tags, wildcards, and argument types. For example, PATTY can generalize the relational pattern "*singer* sings her *song*" into "*singer* sings [prp] *song*", where [prp] represents a pronoun. This approach could reduce the data sparseness problem to some extent, but could not model the compositionality of relational patterns, e.g., similarity between words in two relational patterns.

*Semantic composition*: Mitchell and Lapata (2010) demonstrated the ability of computing the meaning of a phrase from constituent words. They explored various functions for composing phrase vectors, e.g., additive and multiplicative compositions. Mikolov et al. (2013) proposed the Skip-gram model, which was inspired by neural language models (Bengio et al., 2003; Collobert and Weston, 2008). The Skip-gram model exhibits additive compositionality. Levy and Goldberg (2014a) and Levy and Goldberg (2014b) provided theoretical analyses of additive compositionality of the Skip-gram model with negative sampling. Pennington et al. (2014) demonstrated that semantic composition could be modeled also by a co-occurrence matrix between words and their context words. Although these studies achieved good performance in additive compositionality, they cannot model the case in which a word such as 'prevent' or 'inhibit' changes the meaning of an entire phrase, e.g., "prevent the growth of." Baroni and Zamparelli (2010) suggested representing modifiers with matrices rather than with vectors.

MV-RNN (Socher et al., 2012), which is the extension method of RNN (Socher et al., 2011b), can handle a word changing the meaning but they requires supervision data for specific tasks (e.g., sentiment analysis). In addition, those authors did not determine whether the vector representation of internal nodes of a tree really exhibits the meanings of the phrases. Muraoka et al. (2014) proposed a method that reduces MV-RNN parameters. The method uses a single matrix for composing a phrase with the same part-of-speech pattern (e.g., adj–noun). However, they did not evaluate the method for composing a phrase vector from three or more words. Socher et al. (2011a) proposed a method to learn word vectors and a matrix from an unlabeled corpus using an autoencoder but this approach uses only a single matrix for vector composition. In other words, the method cannot take modification of each word into account. Hashimoto et al. (2014) proposed a method for training weights for linear combinations of word vectors. Although their method jointly learns the vector representation and weighting factors of words from an unlabeled corpus, they cannot model changing aspects of words without the capability of linear transformations.

## 5. Conclusion

In this paper, we proposed a novel method for computing the meanings of relational patterns based on semantic compositionality. We extended the Skip-gram model to incorporate semantic compositions modeled by RNNs. In addition, we introduced $l_1$-regularization to obtain a simpler model. The experimental results showed that the proposed method can successfully model semantic compositions of relational patterns, outperforming strong baselines such as additive compositionality. The experiments also demonstrated the contribution of this work to the task of relation extraction. We confirmed that the proposed method could improve not only the quality of vectors for relational patterns but also that for words.

In this study, we defined transformational verbs heuristically. Even though this study could demonstrate superiority in handling transformational verbs, we need to explore a better approach for determining whether a word should have a vector or matrix.

## Acknowledgments

This work was partially supported by Grant-in-Aid for JSPS Fellows Grant no. 26.5820, JSPS KAKENHI Grant number 15H05318, and JST, CREST.

## References

Baroni, M., Zamparelli, R., 2010. 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 (EMNLP 2010), pp. 1183–1193.

Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E., 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Lang. Resour. Eval. 43, 209–226.

Bengio, Y., Ducharme, R., Vincent, P., Janvin, C., 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E.R.H., Mitchell, T.M., 2010. Toward an architecture for never-ending language learning. In: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010), pp. 1306–1313.

Ciaramita, M., Altun, Y., 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pp. 594–602.

Collobert, R., Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning (ICML 2008), pp. 160–167.

dos Santos, C., Xiang, B., Zhou, B., 2015. Classifying relations by ranking with convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015), pp. 626–634.

Fader, A., Soderland, S., Etzioni, O., 2011. Identifying relations for open information extraction. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011), pp. 1535–1545.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E., 2001. Placing search in context: the concept revisited. In: Proceedings of the 10th International Conference on World Wide Web (WWW 2001), pp. 406–414.

Gutmann, M.U., Hyvärinen, A., 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. J. Mach. Learn. Res. 13, 307–361.

Harris, Z., 1954. Distributional structure. Word 10, 146–162.

Hashimoto, K., Stenetorp, P., Miwa, M., Tsuruoka, Y., 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 2014), pp. 1544–1555.

Hashimoto, K., Stenetorp, P., Miwa, M., Tsuruoka, Y., 2015. Task-oriented learning of word embeddings for semantic relation classification. In: Proceedings of the Nineteenth Conference on Computational Natural Language Learning (CoNLL 2015), pp. 268–278.

Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S., 2010. Semeval-2010 task 8: multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010), pp. 33–38.

Huang, E.H., Socher, R., Manning, C.D., Ng, A.Y., 2012. Improving word representations via global context and multiple word prototypes. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012), pp. 873–882.

Koehn, P., 2004. Statistical significance tests for machine translation evaluation. In: Lin, D., Wu, D. (Eds.), Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), pp. 388–395.

Levy, O., Goldberg, Y., 2014a. Linguistic regularities in sparse and explicit word representations. In: Proceedings of the 18th Conference on Computational Natural Language Learning (CoNLL 2014), pp. 171–180.

Levy, O., Goldberg, Y., 2014b. Neural word embedding as implicit matrix factorization. In: Proceedings of the 27th Advances in Neural Information Processing Systems (NIPS 2014), pp. 2177–2185.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th Advances in Neural Information Processing Systems (NIPS 2013), pp. 3111–3119.

Miller, G.A., Charles, W.G., 1991. Contextual correlates of semantic similarity. Lang. Cogn. Process. 6, 1–28.

Min, B., Shi, S., Grishman, R., Lin, C.Y., 2012. Ensemble semantics for large-scale unsupervised relation extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP 2012), pp. 1027–1037.

Mitchell, J., Lapata, M., 2010. Composition in distributional models of semantics. Cogn. Sci. 34, 1388–1439.

Mohamed, T., Hruschka, E., Mitchell, T., 2011. Discovering relations between noun categories. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011), pp. 1447–1455.

Muraoka, M., Shimaoka, S., Yamamoto, K., Watanabe, Y., Okazaki, N., Inui, K., 2014. Finding the best model among representative compositional models. In: Proceedings of the 28th Pacific Asia Conference on Language, Information, and Computation (PACLIC 2014), pp. 65–74.

Nakashole, N., Weikum, G., Suchanek, F., 2012. Patty: a taxonomy of relational patterns with semantic types. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP 2012), pp. 1135–1145.

Pantel, P., Pennacchiotti, M., 2006. Espresso: leveraging generic patterns for automatically harvesting semantic relations. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006), pp. 113–120.

Pennington, J., Socher, R., Manning, C., 2014. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (ENMLP 2014), pp. 1532–1543.

Rink, B., Harabagiu, S., 2010. Utd: classifying semantic relations by combining lexical and semantic resources. In: Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010), pp. 256–259.

Rosenfeld, B., Feldman, R., 2007. Using corpus statistics on entities to improve semi-supervised relation extraction from the web. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007), pp. 600–607.

Rubenstein, H., Goodenough, J.B., 1965. Contextual correlates of synonymy. Commun. ACM 8, 627–633.

Socher, R., Huang, E.H., Pennington, J., Ng, A.Y., Manning, C.D., 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: Proceedings of the 24th Advances in Neural Information Processing Systems (NIPS 2011), pp. 801–809.

Socher, R., Lin, C.C.Y., Ng, A.Y., Manning, C.D., 2011b. Parsing natural scenes and natural language with recursive neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 129–136.

Socher, R., Huval, B., Manning, C.D., Ng, A.Y., 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 (EMNLP 2012), pp. 1201–1211.

Xu, K., Feng, Y., Huang, S., Zhao, D., 2015. Semantic relation classification via convolutional neural networks with simple negative sampling. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pp. 536–540.

Yu, M., Gormley, M.R., Dredze, M., 2014. Factor-based compositional embedding models. In: Workshop on Learning Semantics at the 2014 Conference on Neural Information Processing Systems (NIPS 2014).

Zeichner, N., Berant, J., Dagan, I., 2012. Crowdsourcing inference-rule evaluation. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers (ACL 2012), vol. 2, pp. 156–160.

Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J., 2014. Relation classification via convolutional deep neural network. In: Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014), pp. 2335–2344.