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

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

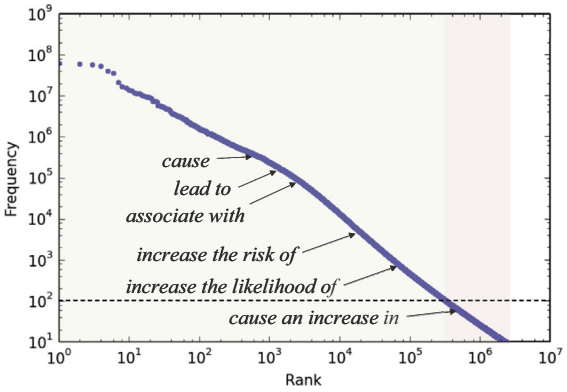

Fig. 1. The frequency of relational patterns in ukWaC.

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.