PAPER

Triple Prediction from Texts by Using Distributed Representations of Words

Takuma EBISU[†,††a)], ***Nonmember and*** **Ryutaro ICHISE**[†,††], ***Senior Member***

SUMMARY Knowledge graphs have been shown to be useful to many tasks in artificial intelligence. Triples of knowledge graphs are traditionally structured by human editors or extracted from semi-structured information; however, editing is expensive, and semi-structured information is not common. On the other hand, most such information is stored as text. Hence, it is necessary to develop a method that can extract knowledge from texts and then construct or populate a knowledge graph; this has been attempted in various ways. Currently, there are two approaches to constructing a knowledge graph. One is open information extraction (Open IE), and the other is knowledge graph embedding; however, neither is without problems. Stanford Open IE, the current best such system, requires labeled sentences as training data, and knowledge graph embedding systems require numerous triples. Recently, distributed representations of words have become a hot topic in the field of natural language processing, since this approach does not require labeled data for training. These require only plain text, but Mikolov showed that it can perform well with the word analogy task, answering questions such as, "*a* is to *b* as *c* is to _?." This can be considered as a knowledge extraction task from a text for finding the missing entity of a triple. However, the accuracy is not sufficiently high when applied in a straightforward manner to relations in knowledge graphs, since the method uses only one triple as a positive example. In this paper, we analyze why distributed representations perform such tasks well; we also propose a new method for extracting knowledge from texts that requires much less annotated data. Experiments show that the proposed method achieves considerable improvement compared with the baseline; in particular, the improvement in HITS@10 was more than doubled for some relations.

key words: *distributed representations of words, knowledge extraction, knowledge graph completion*

1. Introduction

How we describe knowledge of the real world is an important topic in the field of artificial intelligence. Knowledge graphs are the most common way to do this in a form that can be easily processed by computers. Knowledge graphs such as DBpedia [1] and Freebase [2] are used for many tasks, such as question answering, tagging contents, and knowledge inference. Although knowledge graphs already store a huge amount of information, encompassing millions of entities and billions of facts, they are still far from complete. One of the reasons for this is that the construction of knowledge graphs requires a huge amount of human labor because traditionally knowledge graphs are constructed by human editors or from semi-structured information, such as the infoboxes of Wikipedia, which also require human editors. In the meantime, the information to be stored in knowledge graphs has been increasing. Thus, there is a need for a system that can automatically construct knowledge graphs, and many approaches have been considered. Currently, there are two main approaches to automatically constructing triples: one is called open information extraction (Open IE) [3]–[7], and it extracts a triple from a sentence; the other is called knowledge graph embedding [8]–[16], and it predicts missing triples by embedding the entities and relations of an existing knowledge graph. However, both of these approaches have some problems. Stanford Open IE, the best Open IE system, requires an annotated dataset for training, and there is noise in the output triples, which makes it difficult to match the extracted entities and relations to an existing knowledge graph. Knowledge graph embedding systems require sufficient information about entities stored in the knowledge graph beforehand to predict a new triple.

Recently, distributed representations of words have become a hot topic in natural language processing. Distributed representations are used to represent objects by continuous and low-dimensional vectors. Distributed representations of words are obtained from neural network based language models [17]–[20]. They are trained to fill in a blank in a sentence by considering the surrounding words; unlabeled texts are used as training data. This approach has gained the attention of researchers not because of its ability to predict the correct word but because of its ability to capture semantic or syntactic relations between words; in other words, it can complete word analogy tasks, answering questions like, "*a* is to *b* as *c* is to __?". This can be considered the knowledge extraction task of predicting a missing tail entity, given a head entity and a relation. Yet, if we apply this method straightforwardly to relations contained in a knowledge graph, the accuracy is not sufficiently high, because relations in knowledge graphs are more complicated than those of analogy tasks and this method can use only one triple as a positive example.

In this paper, we consider how distributed representations capture syntactic or semantic relations between words, and we propose a novel method for extracting knowledge from texts as an extension of an analogy task and using multiple positive examples. This approach requires far less annotated data and has various advantages over the existing knowledge graph population methods.

Manuscript received March 30, 2017.
Manuscript revised July 27, 2017.
Manuscript publicized September 12, 2017.

[†]The authors are with SOKENDAI (The Graduate University for Advanced Studies), Tokyo, 101–8430 Japan.
[††]The authors are with National Institute of Informatics, Tokyo, 101–8430 Japan.
a) E-mail: takuma@nii.ac.jp

DOI: 10.1587/transinf.2017EDP7112

The remainder of this paper is organized as follows. In Sect. 2, we discuss the current neural network based language models and the extended analogy task. In Sect. 3, we analyze how distributed representations capture similarities or relations between words. In Sect. 4, we present our proposed method, and in Sect. 5, we present an experimental study in which we compare our method with baseline results. In Sect. 6, we discuss related approaches to constructing triples, and in Sect. 7, we present our conclusions.

2. Preliminary Concepts

Our method uses the distributed representations of words obtained by the skip-gram model [21], which is a well-known neural network based language model. This approach is able to succeed at the analogy task, so we used it to predict triples from text. For the analogy task, only one example is used for predicting the relation, which is problematic for applying it to a knowledge graph because of the low accuracy. Therefore, we extend the analogy task so that it uses multiple examples for predicting a triple; we call this the extended analogy task. Here, we will give a brief description of the skip-gram model, the analogy task, and the extended analogy task.

2.1 The Skip-Gram Model

Neural network based language models are used to learn distributed representations of words, and this requires a large unlabeled corpus. The initial neural network based language model was introduced by Bengio et al. [17]. Following this, Mikolov et al. [18] proposed a simplified model with reduced computational complexity; this is currently the most well-known model and is known as the skip-gram model. We summarize it below.

Setting and Notation. We denote respectively the vocabularies of words and contexts as V_w and V_c. Each $w \in V_w$ and $c \in V_c$ is respectively represented by a vector $\vec{w} \in U = \mathbb{R}^d$ or a vector $\vec{c} \in U$, where d is the dimensionality of the vector space U. The vector $\vec{w}$ is called the distributed representation of w, and it is used for many natural language processing tasks. Let a corpus, an unannotated text, be a sequence of words $w_1, w_2, \ldots, w_n$; then, $V_w = \{w_i \mid i = 1, \ldots, n\}$, and the context of a word w_i consists of the words surrounding it in a (2L+1)-sized window $w_{i-L}, w_{i-L+1}, \ldots, w_{i+L}$. We denote the collection of the observed words and context pairs as D and the number of occurrences of $(w, c) \in D$ as $\#(w, c)$. In addition, $\sum_{c' \in V_c} \#(w, c')$, $\sum_{w' \in V_w} \#(w', c)$, and $\sum_{w' \in V_w} \sum_{c' \in V_c} \#(w', c')$ are respectively denoted as $\#w$, $\#c$, and $\#D$.

Objective Function. In this model, the probability of cooccurrence (w, c) is defined as

$$P(w, c) = \sigma(\vec{w} \cdot \vec{c}) = \frac{1}{1 + e^{-\vec{w} \cdot \vec{c}}}$$

where $\vec{w}$ and $\vec{c}$ are the vectors of the model to be learned. The model is optimized so that $P(w, c)$ increases for each observed pair of a word and a context (w, c), and $P(w', c)$ decreases for randomly sampled words w_N; this is called negative sampling. The objective function for a single (w, c) observation is as follows:

$$\log \sigma(\vec{w} \cdot \vec{c}) - k \cdot \mathbb{E}_{w_N \sim P_N}(\log \sigma(\vec{w_N} \cdot \vec{c})) \quad (1)$$

where k is the number of negative samples, and w_N is the sampled word, drawn according to the distribution P_N (usually $P_N(w) \propto \#(w)^\alpha$, where α is a real constant value). Then, the global objective function F is the summation of (1) over the observed examples (w, c) in the corpus:

$$F = \sum_{w \in V_w} \sum_{c \in V_c} (\log \sigma(\vec{w} \cdot \vec{c}) - k \cdot \mathbb{E}_{w_N \sim P_N}(\log \sigma(\vec{w_N} \cdot \vec{c}))) \quad (2)$$

2.2 Equivalence to Implicit Matrix Factorization

Levy et al. [22] showed that the optimization according to Eq. (2) is equivalent to implicit matrix factorization of a pointwise mutual information matrix. We will make use of this later when we consider distributed representations.

Pointwise Mutual Information: The pointwise mutual information (PMI) of a pair of outcomes x and y that belong to discrete random variables X and Y, respectively, is an information-theoretic association measure that is defined as

$$PMI(x, y) \coloneqq \log \frac{P(x, y)}{P(x)P(y)}$$

We are interested in the distribution of words and contexts. The PMI of a word and a context that is estimated from observed examples in a corpus is as follows:

$$PMI(w, c) = \log \frac{\#(w, c) \cdot \#(D)}{\#(w) \cdot \#(c)}$$

Levy et al. proved analytically that the solution vectors to Eq. (2) are also solutions to the following equation:

$$\vec{w} \cdot \vec{c} = PMI(w, c) - \log k \quad (3)$$

2.3 The Analogy Task

One of the advantages of the distributed representations of words obtained from the skip-gram model is that they can solve the analogy task, which is defined as follows.

The Analogy Task: predict a word $w_{2,2}$ that is in the relation r with a word $w_{1,2}$, given pair of words $(w_{1,1}, w_{2,1})$ in the relation r.

A useful characteristic of distributed representations is that if $(w_{1,1}, w_{2,1})$ and $(w_{1,2}, w_{2,2})$ are in the same relation, then $\vec{w_{2,1}} - \vec{w_{1,1}} \simeq \vec{w_{2,2}} - \vec{w_{1,2}}$. Therefore, $w_{2,2}$ is predicted by searching for a word w whose distributed representation $\vec{w}$ is the closest to $\vec{w_{2,1}} - \vec{w_{1,1}} + \vec{w_{1,2}}$.

2.4 The Extended Analogy Task

In the analogy task, only one triple is used as a positive example for predicting a word. Here, we extend the analogy task to use more examples.

The Extended Analogy Task: predict a word w_2 that is in the relation r with a word w_1, given a set of examples of word pairs $\{(w_{1,i}, w_{2,i}) | i = 1, 2, \ldots, N\}$ that are also in the relation r.

The simple extension of the traditional method for the extended analogy task is to take the average $\vec{r}$ of $\vec{w_{2,i}} - \vec{w_{1,i}}$, but this approach does not provide sufficient improvement.

3. Analysis of How the Skip-Gram Model Works

Distributed representations of words obtained from the skip-gram model have gathered attention because the model has the ability to capture the similarity or relations between words. Similar words are represented by close distributed representations, and if pairs of words $(w_{1,1}, w_{2,1})$ and $(w_{1,2}, w_{2,2})$ represent the same relation, then the formula $\vec{w_{2,1}} - \vec{w_{1,1}} \simeq \vec{w_{2,2}} - \vec{w_{1,2}}$ holds. Thus, the analogy task that consists of the question "$w_{1,1}$ is to $w_{2,1}$ as $w_{1,2}$ is to __?" can be answered by searching for the distributed representation that is closest to $\vec{w_{2,1}} - \vec{w_{1,1}} + \vec{w_{1,2}}$. Although Levy et al. have probed Eq. (3) which relates distributed representation of words with statistic of corpus, they did not explain why the model captures the similarity and relations between words. In this section, we will show that these mechanism can be explained under certain assumptions by using Levy's result Eq. (3). Following this, in the next section, we will propose a new method for the extended analogy task.

3.1 How the Skip-Gram Model Captures the Similarity of Words

Here, we analyze how the skip-gram model captures similarity; that is, why $\vec{w_1}$ and $\vec{w_2}$ are close when words w_1 and w_2 have similar meaning. To do so, we assume the distributional hypothesis [23], and we then show how similarity of the words is captured under this hypothesis.

The Distributional Hypothesis: a word is characterized by the company it keeps.

According to this hypothesis, the words surrounding semantically similar words w_1 and w_2 have similar distributions; that is, if $P(c|w_1) \simeq P(c|w_2)$ for an arbitrary $c \in V_c$, then this implies $PMI(w_1, c) \simeq PMI(w_2, c)$. According to Eq. (3), if $PMI(w_1, c) = \vec{w_1} \cdot \vec{c} - \log k$ and $PMI(w_2, c) = \vec{w_2} \cdot \vec{c} - \log k$, then the equation $\vec{w_1} \cdot \vec{c} \simeq \vec{w_2} \cdot \vec{c}$ holds for arbitrary $c \in V_c$. Hence, the vectors $\vec{w_1}$ and $\vec{w_2}$ are also close.

3.2 How the Skip-Gram Model Performs the Analogy Task

We now consider how the skip-gram model performs the analogy task; that is, we ask why $\vec{w_{2,1}} - \vec{w_{1,1}}$ and $\vec{w_{2,2}} - \vec{w_{1,2}}$ are close when word pairs $(w_{1,1}, w_{2,1})$ and $(w_{1,2}, w_{2,2})$ are in the same relation r. Usually $\{w_{1,1}, w_{1,2}\}$ and $\{w_{2,1}, w_{2,2}\}$ have the same types: respectively, type 1 and type 2. The distributed hypothesis says the meaning of a word is defined by the cooccurrence of each word, and here, to explain the mechanism to capture a relation between words, we will assume two new hypotheses that arar to the distributional hypothesis.

The Type Distributional Hypothesis: There exists $V_t \subset V_c$, and the type of a word is determined by its company in V_t.

The Relational Distributional Hypothesis: For a relation r whose domain is words of type 1 and whose range is words of type 2, there exists $V_r \subset V_c$; if a word of type 1 and a word of type 2 are in the relation r, then the company they keep in V_r is similar.

For example, for the types "human" and "place", pronouns like "his", "him", and "her", or verbs like "walk" or "eat" tend to cooccur with a human name, and the words like "population" and "area" are likely to cooccur with a place name. For the relation "live in", country-specific words such as the names of traditional things or particular places tend to cooccur with the country and the people who live there.

A question of the analogy task consists of two pairs of words in the same relation r: $(w_{1,1}, w_{2,1})$ and $(w_{1,2}, w_{2,2})$, where $w_{2,2}$ is a word to be predicted. Following the type distributional hypothesis, there exists $V_t \subset V_w$ that characterizes the type of words, and $w_{1,1}$ and $w_{1,2}$ have the same type; thus $PMI(w_{1,1}, c) \simeq PMI(w_{1,2}, c)$ for arbitrary $c \in V_t$. The same is true of $w_{2,1}$ and $w_{2,2}$. Following the relational distributional hypothesis, there exists $V_t \subset V_w$ that characterizes the type of words and $V_r \subset V_w$ that characterizes r, and if $w_{1,1}$ and $w_{2,1}$ have the relation r, then $PMI(w_{1,1}, c) \simeq PMI(w_{2,1}, c)$ for arbitrary $c \in V_t$. The same is true of $w_{1,2}$ and $w_{2,2}$. Therefore, the following four equations hold:

$$\vec{w_{1,1}} \cdot \vec{c} \simeq \vec{w_{1,2}} \cdot \vec{c}, \quad \vec{w_{2,1}} \cdot \vec{c} \simeq \vec{w_{2,2}} \cdot \vec{c} \quad (\forall c \in V_t)$$

$$\vec{w_{1,1}} \cdot \vec{c} \simeq \vec{w_{2,1}} \cdot \vec{c}, \quad \vec{w_{1,2}} \cdot \vec{c} \simeq \vec{w_{2,2}} \cdot \vec{c} \quad (\forall c \in V_r)$$

Hence, the following equation holds:

$$(\vec{w_{2,1}} - \vec{w_{1,1}} + \vec{w_{1,2}}) \cdot \vec{c} \simeq \vec{w_{2,2}} \cdot \vec{c} \quad (\forall c \in V_t \cup V_r)$$

Therefore, $\vec{w_{2,1}} - \vec{w_{1,1}} + \vec{w_{1,2}}$ and $\vec{w_{2,2}}$ are similar, and the analogy task can be answered by the distributed representation of words.

Here, we did not consider the words in $V_{rest} = V_c \setminus (V_t \cup V_r)$. We think they adversely affect prediction, and if they are ignored, accuracy will be improved.

4. Proposed Method for The Extended Analogy Task

In this section, we propose a novel method for the extended analogy task using distributed representations of words. In the previous section, we inferred that some words are important for the analogy task under some assumptions, and we

think that the other words adversely affect the performance of the task. After processing text using a neural network based language model, output distributed representations of words have mixed information about cooccurrence with all other words. First, we will show that we can ignore the effect of any given word by using a projection function. After that, we will present a method for selecting appropriate information.

4.1 Reduction of Word Information in Distributed Representations

The distributed representation of a word has complete information about the cooccurrence of that word with other words. However, we inferred that only some of the words are important for the analogy task. Here, we introduce a projection function to reduce the information of cooccurrences for a word.

We denote this projection function from U to U' as $proj_{U'}$, and it is defined as follows:

$$proj_{U'}(\vec{v}) = \sum_{i=1}^{n} (\vec{b_i} \cdot \vec{v})\vec{v} \quad (\vec{v} \in U)$$

where $b_1, b_2, \ldots, b_n$ is a orthonormal basis of U'. If $\vec{c}$ is perpendicular to U', where $c \in V_c$, then $proj_{U'}(\vec{w}) \cdot \vec{c} = 0$ holds for arbitrary $w \in V_w$. According to Eq. (3), the value of $\vec{w} \cdot \vec{c}$ represents the frequency of cooccurrence of w and c, so we can ignore the effect of the cooccurrence with c by using $proj_{U'}$. On the other hand, if $\vec{c}$ is included in U', then $proj_{U'}(\vec{w}) \cdot \vec{c} = \vec{w} \cdot \vec{c}$ holds for arbitrary $w \in V_w$. Hence, by choosing an appropriate U', we can leave the information about the contexts that we need.

4.2 How to Determine a Subspace U' for the Extended Analogy Task

In Sect. 4.1, we showed a way to select information about cooccurring words in distributed representations by using a projection function. In this section, we will show how to find an appropriate U' for the projection when performing the extended analogy task.

We want the distributed representations to hold information about V_t and V_r because we concluded that it was important for the analogy task; this means we want $\vec{c}$ to be in U'. We also inferred that $(\vec{w_{2,1}} - \vec{w_{1,1}}) \cdot \vec{c} \simeq (\vec{w_{2,2}} - \vec{w_{1,2}}) \cdot \vec{c}$ holds for $\forall c \in V_t \cup V_r$, where the word pairs $(w_{1,1}, w_{2,1})$ and $(w_{1,2}, w_{2,2})$ are in the same relation r. Then, all $(w_{2,i} - w_{1,i}) \cdot \vec{c}$ are similar for arbitrary $c \in V_t \cup V_r$, where $\{(w_{1,i}, w_{2,i}) | i = 1, 2, \ldots, N\}$ is a given set. Hence, we determine U' so that $proj_{U'}(\vec{w_{2,i}} - \vec{w_{1,i}})$ are similar for arbitrary $(w_{1,i}, w_{2,i}) \in S_r$. The example of the "capital" relation is presented in Fig. 1. We set the objective function F to determine U' as the variance of $\{proj_{U'}(\vec{w_{2,i}}) - proj_{U'}(\vec{w_{1,i}}) \mid (w_{1,i}, w_{2,i}) \in S_r\}$:

$$F = \sum_{i=1}^{N} \|proj_{U'}(\vec{w_{2,i}}) - proj_{U'}(\vec{w_{1,i}}) - proj_{U'}(\vec{r})\|_2^2 \quad (4)$$

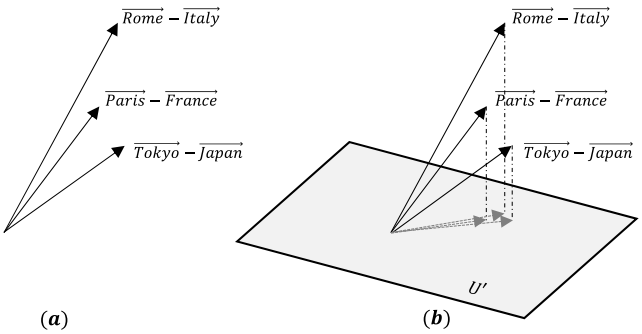

Fig. 1 The differences of word vectors in the relation "capital". The original difference vectors are located apart as shown in (a). The proposed method chooses important elements of vectors by projection to U' and makes them close as shown in (b)

where $\vec{r} = \sum_{i=1}^{N} (\vec{w_{2,i}} - \vec{w_{1,i}})/N$, and d', the difference between d and the dimension of U', is a hyperparameter. For the extended analogy task, w_2 is predicted by finding the word whose projected distributed representation is the closest to $proj_{U'}(\vec{w_1} + \vec{r})$.

4.3 Calculation Technique

We now present a technique for implementing the proposed method. Let $(b_1, \ldots, b_d)$ be an orthonormal basis of U, where $b_{d'+1}, \ldots, b_d$ span U'. Then,

$$\begin{aligned} F &= \sum_{i=1}^{N} \|proj_{U'}(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})\|_2^2 \\ &= \sum_{i=1}^{N} \sum_{j=d'+1}^{d} (b_j^{\mathrm{T}}(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r}))^2 \\ &= \sum_{i=1}^{N} \sum_{j=d'+1}^{d} b_j^{\mathrm{T}}(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})^{\mathrm{T}} b_j \\ &= \sum_{j=d'+1}^{d} b_j^{\mathrm{T}}(\sum_{i=1}^{N}(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})^{\mathrm{T}}) b_j \end{aligned}$$

We note that $\sum_{j=d'+1}^{d}(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})^{\mathrm{T}}$ is a symmetric matrix. Hence, there exists an orthonormal matrix O and a diagonal matrix Λ such that $\sum_{j=d'+1}^{d}(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})^{\mathrm{T}} = O\Lambda O^{\mathrm{T}}$. The vectors of O are eigenvectors of $\sum_{j=d'+1}^{d}(\vec{y_{w,i}} - \vec{w_{1,i}} - \vec{r})(\vec{w_{2,i}} - \vec{w_{1,i}} - \vec{r})^{\mathrm{T}}$, and the diagonal elements of Λ are their eigenvalues. Therefore, we can choose $d - d'$ eigenvectors in descending order of their eigenvalues and take their corresponding eigenvectors as $b_{d'+1}, \ldots, b_d$ to minimize F. These $b_{d'+1}, \ldots, b_d$ determine U'.

5. Experiments

5.1 Experimental Settings

We conduct the extended analogy task. That is, when given

an entity and a relation, we used proposed method to predict the missing entity, or in other words, we predict the tail entity t given $(h, r, *)$. This is an extension of the analogy task, and multiple examples can be used as training data. We conducted experiments following three different approaches.

Exp. 1 We examined the effect of the hyperparameter of proposed method, d', on the accuracy of the extended analogy task by letting its value range from 0 to 300. The experiment was conducted using 10-fold cross-validation for each relation.

Exp. 2 We attempted to determine an appropriate value of d'. We conducted 10-fold cross-validation on the training data, and then used the d' with the highest accuracy on the training data to evaluate the test data.

Exp. 3 Without a knowledge graph, the triples for the training data must be selected by human editors. In such a situation, only a small number of triples will be available for use as training data. We randomly selected 100 and 1000 triples for the training and test data, respectively.

5.1.1 Distributed Representation of Words and Word Pairs for Experiments

Our experiments were conducted on distributed representations made from Wikipedia texts and some of the triples in the FB13 dataset [24].

We used word2vec[†] to obtain the distributed representation of words. The dimension of the representation space was set to 300, and the skip-gram model was used to obtain the distributed representations. The training corpus was Wikipedia in English, which has a total of 1.8 billion words. Prior to processing by word2vec, we processed it twice by word2phrase to form phrases. This meant that there were phrases of up to four words, since word2phrase connects two words that frequently occur together. After processing the corpus by word2vec, we obtained 863,822 unique words. As in the paper by Mikolov [18], we used the normalized distributed representation.

FB13 is a dataset of triples from the people domain of Freebase, and it contains triples concerned with thirteen relations. We used those triples of which the head and tail entities existed in distributed representations. Details of the dataset are shown in Table 1. Triples (h_i, r, t_i) in the dataset are considered as word pairs (h_i, t_i) in the relation r.

5.1.2 Evaluation Protocol

10-fold cross-validation was conducted for each relation in each experiment. In Experiments 1 and 2, for each step, we used 90% of the dataset for training and 10% for tests. In Experiment 3, 100 word pairs were randomly chosen for training data, and 1000 word pairs were used for testing. We

[†]https://code.google.com/archive/p/word2vec/

Table 1 Number of triples of FB13W

Relation	# triples	Relation	# triples
gender	11452	cause of death	3935
nationality	10171	religion	2614
profession	12674	parents	446
place of death	3935	children	435
place of birth	7978	ethnicity	1243
location	7170	spouse	621
institution	1820		

determined U' using the training data as in (4) and given d'. Each test pair (h, t) in the relation was then corrupted by replacing the tail with every word w in the vocabulary, so that we obtained 863,822 corrupted word pairs. We then calculated validity of corrupted triples and ranked them in descending order. The score of the word pairs (h, w) was calculated according to the score function S:

$$S(h, w) = -\|proj_{U'}(\vec{r}) - proj_{U'}(\vec{w} - \vec{h})\|_2^2$$

We then got the rank of the original word pairs, and we calculated the accuracy of the prediction in two ways: the proportion of testing triples whose rank is the top (HITS@1) and whose rank is in the top 10 (HITS@10). A higher HITS@1 or HITS@10 indicates a better performance.

For comparison, we conducted the extended analogy task using other methods. The first method is the one used for the original analogy task. In this method, for each word pair (h, t), a positive example (h', t') is chosen randomly from the training set. The score function for the corrupted word pair (h, w) is as follows:

$$S_{ANA}(h, w) = -\|(\vec{t'} - \vec{h'}) - (\vec{w} - \vec{h})\|_2^2$$

We will refer to this method as ANA.

In the second method, the average of the difference between the distributed representations of the head and tail entities is used to represent the relation. The score function is as follows:

$$S_{AVE}(h, w) = -\|\vec{r} - (\vec{w} - \vec{h})\|_2^2$$

We will refer to this method as AVE. Note that AVE is a special case ($d' = 0$) of the proposed method.

We will refer to the proposed method as PM.

5.2 Experiment 1

To examine the effect of d', the accuracy rate of the extended analogy task was evaluated using the proposed method and letting d' vary from 0 to 300.

Experimental results are shown in Table 2 and labeled PM (Exp. 1); these are the best scores obtained, and d' at that time is indicated. We observed following.

The accuracy of AVE was better than that of ANA for most relations. However, for "cause of death" and "ethnicity", the accuracy was worse. This means that the vectors $(\vec{t} - \vec{h})$ for these relation did not form a single cluster but were divided into multiple clusters. Hence, by taking their

Table 2 Results of experiments 1 and 2

Tasks	HITS@1 (%)					HITS@10 (%)				
Relation	ANA	AVE	PM (Exp. 1)	d'	PM (Exp. 2)	ANA	AVE	PM (Exp. 1)	d'	PM (Exp. 2)
gender	79.1	88.6	92.8	30	92.9	99.2	100	100	0	100
nationality	20.5	46.2	66.2	10	66.2	36.4	70.3	95.8	20	94.4
profession	5.1	9.5	27.4	30	26.6	18.7	29.6	61.4	40	60.1
place of death	1.9	2.2	15.1	40	14.7	8	9.3	35.4	40	33.5
place of birth	0.8	0.7	6.7	40	7.0	3.2	3.5	20.3	60	19.7
location	0.9	0.5	7.0	20	6.6	3.4	2.8	22.0	40	21.7
institution	2.9	5.8	15.7	20	14.4	14.7	25.9	45.2	90	44.6
cause of death	4.4	1.3	11.2	10	10.5	9.4	11.1	56.7	60	54.7
religion	9.9	20.1	34.3	40	34.7	26.9	48.4	79.8	60	76.2
parents	6.0	8.5	9.2	10	8.5	25.8	34.3	39.5	70	38.6
children	7.8	8.7	9.2	20	8.5	23.9	34.2	35.7	30	34.4
ethnicity	23.2	8.8	49.6	40	47.9	25.3	23.9	88.3	160	82.7
spouse	5.5	6.4	72.1	220	71.3	12.4	16.3	89.5	220	88.7

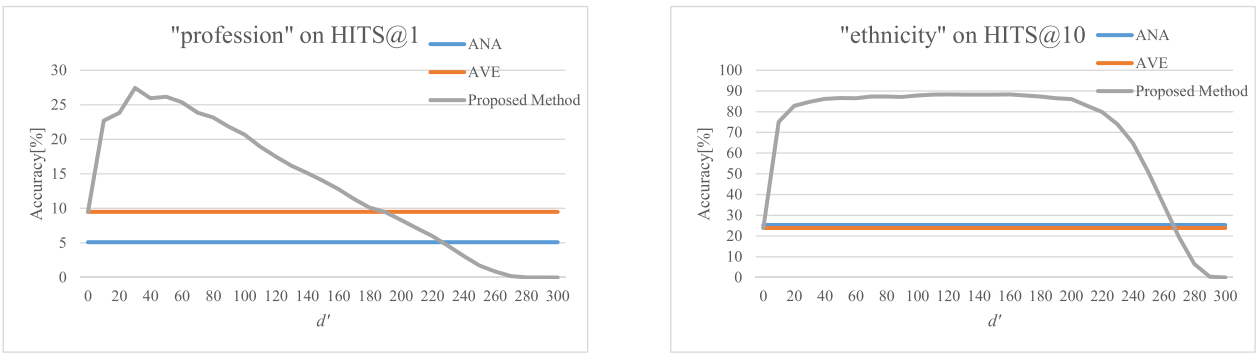

Fig. 2 Graphs for "profession" and "ethnicity"

average, $\vec{r}$ was located far from the other vectors ($\vec{t} - \vec{h}$).

The values of d' with HITS@1 were mostly in the range from 10 to 40, and the values of d' with HITS@10 are mostly in the range from 20 to 90. Hence, the dimensionality of the remaining subspace was greater than 200, and the discarded subspace was small compared with the remaining subspace. Also, note that d' for "spouse" was significantly high, 230 and 210 for HITS@1 and HITS@10, respectively.

The accuracy rate of "spouse" was unexpectedly improved by the proposed method. We think this was because "spouse" is the only symmetrical relation considered. Hence, it is highly possible that reversed triples in the testing data were contained in the training data.

The accuracies for "children" and "parents" were low. We think this is because the range types of these relations are human and the frequency of the occurrence of a specific person is relatively low in the corpus.

The accuracy of our proposed method was higher than each of the baselines. The results of the tasks for "religion", "ethnicity", and "spouse" were especially good. All of them were improved by more than 10% for HITS@1, and for "spouse", the proposed method outperformed the baseline, with an improvement of more than 65% improvement. Predictions for "gender" were the most accurate with the proposed method, at 93.1%. These results show for the proposed method that some of the cooccurrence information is important for predicting related words and the other information adversely affects prediction.

The details of the effect of d' for "profession" and "ethnicity" are shown in Fig. 2. We observed the following.

In both graphs, accuracy increases rapidly as d' increases, and then stabilizes until d' becomes too large, after which it decreases because important components are lost. We show this for only two relations, but the same pattern occurred with the other relations.

As can be seen, accuracy is maintained over a wide range of d'. This shows the robustness of the proposed method.

We conclude the proposed method is better than baselines if the value of d' can be appropriately chosen and to choose the value is expected to be easy because of robustness from these two results.

5.3 Experiment 2

Experiment 1 was performed to evaluate for each preset value of d'. However, when the proposed method is used as the predictor, d' needs to be determined in advance. In this experiment, d' was determined a priori from training data. To do so, for each step of the 10-fold cross-validation, we repeated the 10-fold cross-validation for the training set with a different value of d'; we thus determined the value for d' that resulted in the best accuracy.

Experimental results are shown in Table 2, where the scores of this experiment are labeled PM (Exp. 2). We observed the following.

Only the result for "children" was lower than the baseline for HITS@1, and the result for "parents" equaled that of the baseline. We think this occurred because there were insufficient training data. The triples for "children" and "parents" had the lowest frequencies. The accuracy of HITS@10 was still better than the baseline.

The improvements in HITS@10 were especially large. The accuracy for "cause of death" was improved by 44%, that for "ethnicity" by 59%, and that for "spouse" by 72%.

In most cases, the accuracy of the proposed method was higher than that of the baseline. They were a little lower than the best accuracy rates of the proposed method in Experiment 1, but they were still competitive. These results show that the dimension of the U' can be adequately determined from the training data and the proposed method is the most suitable for the extended analogy task.

5.4 Experiment 3

Experiments 1 and 2 used a large number of triples as the training dataset. This is reasonable for the completion of an existing knowledge graph because these are already available. However, there are difficulties when a new knowledge graph must be prepared by human editors. Thus, we conducted Experiment 3 using a small training set as a way of evaluating the usefulness of the proposed method for the construction of a new knowledge graph. We randomly choose 100 word pairs for the training dataset and 1000 for the test dataset. If there were not enough word pairs, then 100 triples were chosen at random, and the remainder were used for the test.

Experimental results are shown in Table 3. We observed the following.

Prediction of "gender", "nationality", and "profession" were still good, but for "ethnicity" and "spouse" there was little improvement in accuracy. As can be seen, the most appropriate d' for "ethnicity" and "spouse" were higher than the optimal values for the others. Hence we assume that this was because a higher dimensional subspace would be redundant for predicting triples concerned with these relations and would require more triples to determine.

Table 3 Results of experiment 3

Tasks	HITS@1 (%)		HITS@10 (%)	
Relation	AVE	PM	AVE	PM
gender	60.7	63.1	66.7	66.7
nationality	30.6	42.0	45.4	61.5
profession	6.4	18.7	20.5	42.3
place of death	1.5	9.1	6.1	20.9
place of birth	0.4	2.8	2.0	8.0
location	0.2	3.3	1.5	9.8
institution	1.0	2.6	4.6	8.2
cause of death	0.4	3.9	3.8	17.5
religion	5.0	8.0	12.4	20.0
parents	1.1	0.8	4.4	5.0
children	1.2	0.9	4.5	4.9
ethnicity	1.0	6.3	2.1	10.4
spouse	0.1	2.3	2.0	3.6

Most of the results with the proposed method were better than those of the baseline for HITS@1 and all results were better than for HITS@10. These results show that the proposed method is better than the baseline method for constructing a knowledge graph. We conclude the proposed method can adequately determine U' with small amount of training set and predict unknown triples more precisely than the baseline to construct a new knowledge graph.

6. Related Work

There have been many previous studies of knowledge graph completion and knowledge extraction from texts. Currently, there are two main approaches: knowledge graph embedding and open information extraction. Each of these methods and our proposed method requires a different type of data for training and has unique characteristics.

6.1 Knowledge Graph Embedding

Knowledge graph embedding is currently the most active area of research for prediction of links in knowledge graphs. The basic model of knowledge graph embedding is TransE [11]; in this model, entities and relations are represented by vectors, and parameters are learned so that the difference vector of entities matches the vector of the relation between them.

Many variations of TransE have been proposed. TransH [12] uses a projection to a hyperplane for each relation. There might be multiple relations between a given pair of words, so they tried differentiate the relation by leveraging projection. In TransR [13], each relation has an additional matrix, and the entity vectors are transformed before calculating the difference. Relations may have multiple meanings, so in TransG [16], each relation is represented by multiple vectors. These models are used for link prediction or triple classification, similar to what is done in our proposed method.

These models only require a knowledge graph, but it must be sufficiently large, since it is the only source of information. If there is no triple that includes a given entity, then the model will not be able to predict its relations. In our proposed method, we use text, but we use only triples for a single relation. We can predict a triple including a new entity without the need for a priori triples with it.

6.2 Open Information Extraction

OpenIE refers to the extraction of triples from a sentence. REVERB [4], Ollie [5], and Stanford OpenIE [7] are well-known models. This approach is useful for abstraction of a sentence, but output entities and relations are sometimes modified by other words. There are many ways to represent a given entity or relation, so it is necessary to have a system for matching entities and relations with the knowledge graph in order to predict links and classify triples [25]–[27]. Stanford Open IE, the current best such system, is trained

using labeled data: sentences are annotated to indicate what triples can be extracted. Hence, to adapt new relations, additional annotated data must be prepared, and this is not easy.

OpenIE models can extract only triples that are clearly stated in a sentence, but our proposed method can predict those that are stated indirectly. For example, "Musashi Miyamoto", a famous samurai in Japan, and "male" do not appear in any sentence in the corpus, so an OpenIE system cannot predict the triple (Musashi Miyamoto, gender, male). However our proposed method predicted this triple in the experiment using information contained in the surrounding words.

7. Conclusions

In this paper, we analyzed how the skip-gram model captures similarity and relations between words, and we showed that information of cooccurrence about particular words are important for the extended analogy task. We also proposed a new method for the extended analogy task, using distributed word representations that require less labeled data than do traditional methods. Experiments with the extended analogy task gave results that were far better than baseline results and showed that our proposed method has advantages over the traditional methods. Our method can extract information as triples from text, which is the most common form in which information is stored.

References

[1] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, et al., "Dbpedia–a large-scale, multilingual knowledge base extracted from Wikipedia," Semantic Web, vol.6, no.2, pp.167–195, 2015.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp.1247–1250, 2008.

[3] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka, and T.M. Mitchell, "Toward an architecture for never-ending language learning," in Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, pp.1306–1313, 2010.

[4] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp.1535–1545, 2011.

[5] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni, "Open language learning for information extraction," in Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp.523–534, 2012.

[6] F. de Sá Mesquita, J. Schmidek, and D. Barbosa, "Effectiveness and efficiency of open relation extraction," in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp.447–457, ACL, 2013.

[7] G. Angeli, M.J.J. Premkumar, and C.D. Manning, "Leveraging linguistic structure for open domain information extraction," in Proceedings of the 2015 Conference of the Association for Computational Linguistics, pp.344–354, The Association for Computer Linguistics, 2015.

[8] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in Proceedings of the Twenty-Fifth Conference on Artificial Intelligence, AAAI Press, pp.301–306, 2011.

[9] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," in Proceedings of 15th International Conference on Artificial Intelligence and Statistics, pp.127–135, 2012.

[10] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," Machine Learning, vol.94, no.2, pp.233–259, 2014.

[11] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in Proceedings of the 2013 Conference and Workshop on Neural Information Processing Systems, pp.2787–2795, 2013.

[12] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp.1112–1119, AAAI Press, 2014.

[13] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp.2181–2187, 2015.

[14] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp.705–714, The Association for Computational Linguistics, 2015.

[15] S. He, K. Liu, G. Ji, and J. Zhao, "Learning to represent knowledge graphs with Gaussian embedding," in Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp.623–632, 2015.

[16] H. Xiao, M. Huang, Y. Hao, and X. Zhu, "TransG: A generative mixture model for knowledge graph embedding," CoRR, vol.abs/1509.05488, pp.2316–2325, 2015.

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," Journal of Machine Learning Research, vol.3, pp.1137–1155, March 2003.

[18] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Proceedings of the 2013 Conference and Workshop on Neural Information Processing Systems, pp.3111–3119, 2013.

[19] L. Vilnis and A. McCallum, "Word representations via Gaussian embedding," CoRR, vol.abs/1412.6623, 2014.

[20] S.K. Jauhar, C. Dyer, and E. Hovy, "Ontologically grounded multi-sense representation learning for semantic vector space models," in Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.683–693, May–June 2015.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," CoRR, vol.abs/1301.3781, 2013.

[22] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in Proceedings of the 2014 Conference and Workshop on Neural Information Processing Systems, pp.2177–2185, Curran Associates, Inc., 2014.

[23] J.R. Firth, "A synopsis of linguistic theory 1930-55," Studies in Linguistic Analysis, vol.1952-59, pp.1–32, 1957.

[24] R. Socher, D. Chen, C.D. Manning, and A.Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in Proceedings of the 2013 Conference and Workshop on Neural Information Processing Systems, pp.926–934, 2013.

[25] I. Augenstein, S. Padó, and S. Rudolph, "Lodifier: Generating linked data from unstructured text," in Proceedings of the 9th International Conference on The Semantic Web: Research and Applications, pp.210–224, 2012.

[26] P. Exner and P. Nugues, "Entity extraction: From unstructured text to dbpedia rdf triples," in Proceedings of the Web of Linked Entities Workshop in conjuction with the 11th International Semantic Web

Conference, pp.58–69, 2012.
[27] N. Kertkeidkachorn and R. Ichise, "T2kg: An end-to-end system for creating knowledge graph from unstructured text," in Proceedings of AAAI Workshop on Knowledge-based Techniques for Problem Solving and Reasoning, pp.743–749, 2017.

Takuma Ebisu received his B.Sc degree in mathematic from Kyoto University, Kyoto, Japan in 2013. He is currently a Ph.D candidate at SOKENDAI (The Graduate University for Advanced Studies) in Japan. His research interests include knowledge representation, machine learning and natural language processing.

Ryutaro Ichise received his Ph.D. degree in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 2000. From 2001 to 2002, he was a visiting scholar at Stanford University. He is currently an associate professor in the Principles of Informatics Research Division at the National Institute of Informatics in Japan. His research interests include semantic web, machine learning, and data mining.