General Paper

Reflection-based Word Attribute Transfer

Yoichi Ishibashi†, Katsuhito Sudoh†, Koichiro Yoshino† and Satoshi Nakamura†

Word embeddings, which often represent analogic relations such as $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} \approx \overrightarrow{queen}$, can be used to change an attribute of a word, including its gender. To transfer the gender attribute of *king* to obtain *queen* in this analogy, we subtract a difference vector $\overrightarrow{man} - \overrightarrow{woman}$ from *king* based on the knowledge that *king* is male. However, developing such knowledge is significantly costly for words and attributes. In this work, we propose a novel method for word attribute transfer based on reflection mapping without an analogy-based operation. Experimental results show that our proposed method can transfer the word attributes of the given words without changing the words that are invariant with respect to the target attributes.

Key Words: *Word, Word Embedding, Reflection, Word Attribute Transfer*

1 Introduction

Distributed representation (Hinton et al. 1984) is a type of data representation that is trained on objectives to embed similar data samples into closed points in a vector space for capturing their similarities. In recent natural language processing, several studies have used neural networks. The distributed representation is compatible with neural networks because the representation can capture the features of language data and compress them into low-dimensional vectors. Word embedding methods handle word semantics in natural language processing (Mikolov et al. 2013a, 2013b; Pennington et al. 2014; Nickel and Kiela 2017; Vilnis and McCallum 2015). Word embedding models such as skip-gram with negative sampling (SGNS) (Mikolov et al. 2013b) or global vectors for word representation (GloVe) (Pennington et al. 2014) capture analogic relations such as $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} \approx \overrightarrow{queen}$. Previous works (Levy and Goldberg 2014b; Arora et al. 2016; Gittens et al. 2017; Ethayarajh et al. 2019; Allen and Hospedales 2019) offer theoretical explanations based on pointwise mutual information (PMI) (Church and Hanks 1990) for maintaining analogic relations in word vectors.

These relations can be used to transfer a certain attribute of a word, such as changing *king* into *queen* by transferring its gender. This transfer can be applied to perform data augmentation; for example, rewriting *He is a boy* to *She is a girl*. It can be used to generate negative examples

† Graduate School of Science and Technology, Nara Institute of Science and Technology

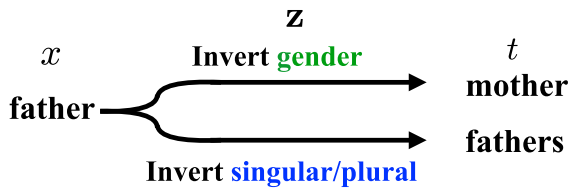


Fig. 1 Examples of word attribute transfer

for natural language inference (Kang et al. 2018). For example, in the Stanford Natural Language Inference (SNLI) corpus (Bowman et al. 2015), negative examples can be generated by transferring a hypothesis sentence from entailment to contradiction. We tackled a novel task that changes a word by transferring certain attributes associated with the word, which is called *word attribute transfer* (Fig. 1).

A naive way for word attribute transfer is to use a difference vector based on analogic relations, such as adding $\overrightarrow{woman} - \overrightarrow{man}$ to $\overrightarrow{king}$ to obtain $\overrightarrow{queen}$. This requires explicit knowledge whether an input word is male or female. We have to add a difference vector to a male word and subtract it from a female word for achieving gender transfer. We also have to avoid changing words that are invariant with respect to gender attributes, such as *is* and *a* in the example above, as they are gender-invariant words. Developing such knowledge is significantly costly for words and attributes in practice. In this paper, we propose a novel framework for word attribute transfer based on *reflection* that does not require explicit knowledge of the given words in its prediction.

The contributions of this work are twofold: (1) We propose a word attribute transfer method that obtains a vector with an inverted binary attribute without explicit knowledge. (2) The proposed method demonstrates more accurate word attribute transfer for words that have target attributes than other baseline methods, while ensuring that the words that do not have target attributes are unchanged.

2 Word Attribute Transfer Task

In this task, we focus on modeling the binary attributes (e.g., male and female[1]). Let x denote a word and let $\mathbf{v}_x \in \mathbb{R}^n$ denote its n-dimensional vector representation. We assume that $\mathbf{v}_x$ is learned in advance using an embedding model, such as a skip-gram. In this task,

[1] Gender-specific words are sometimes considered socially problematic. Here, we use this as an example based on the man-woman relation.

we have two inputs, a word x and vector $\mathbf{z} \in \mathbb{R}^n$, which represents a certain target attribute, and an output word y. y is the word obtained through the transfer of x according to the target attribute specified by $\mathbf{z}$. y should be the same as the reference word t. Note that t is the same as x when x is invariant based on the target attribute. In this paper, $\mathbf{z}$ is an n-dimensional vector embedded from a target attribute ID by using an embedding function of a deep learning framework. For example, given a set of attributes $\boldsymbol{\mathcal{Z}} = \{\text{gender}, \text{antonym}\}$, we assign different random vectors $\mathbf{z}_{\text{gender}}$ for gender and $\mathbf{z}_{\text{antonym}}$ for antonym. Let $\mathcal{A}$ denote a set of triplets $(x, t, \mathbf{z})$, e.g., $(man, woman, \mathbf{z}_{\text{gender}}) \in \mathcal{A}_{\text{gender}}$, and $\mathcal{N}$ denote a set of invariant words for an attribute $\mathbf{z}$, e.g., $(person, \mathbf{z}_{\text{gender}}) \in \mathcal{N}_{\text{gender}}$. This task transfers an input word vector $\mathbf{v}_x$ to an output word vector $\mathbf{v}_y \in \mathbb{R}^n$ by using a transfer function $f_{\mathbf{z}_{\text{attr}}}$ that inverts the attribute $\mathbf{z}_{\text{attr}}$ of $\mathbf{v}_x$. $\mathbf{v}_y$ is expected to be the same as its reference word vector $\mathbf{v}_t \in \mathbb{R}^n$. This is denoted according to the following formula:

$$\mathbf{v}_t \approx \mathbf{v}_y = f_{\mathbf{Z}}(\mathbf{v}_x). \tag{1}$$

The following properties must be satisfied: (1) attribute words $\{x|(x, t, \mathbf{z}) \in \mathcal{A}\}$ are transferred to their counterparts, and (2) invariant words $\{x|(x, \mathbf{z}) \in \mathcal{N}\}$ are not changed (are transferred back into themselves). For instance, with $\mathbf{z}_{\text{gender}}$, for a given input word *man*, the gender attribute transfer $f_{\mathbf{z}_{\text{gender}}}(\mathbf{v}_{man})$ should result in a vector close to $\mathbf{v}_{woman}$. When given another input word *person* as x, the result should be $\mathbf{v}_{person}$.

3 Analogy-based Word Attribute Transfer

Analogy is a general idea that can be used for word attribute transfer. PMI-based word embedding methods, such as SGNS and GloVe, capture analogic relations, as shown in Eq. 2 (Mikolov et al. 2013c; Levy and Goldberg 2014a; Linzen 2016). By rearranging Eq. 2, Eq. 3 is obtained:

$$\mathbf{v}_{queen} \approx \mathbf{v}_{king} - \mathbf{v}_{man} + \mathbf{v}_{woman}, \tag{2}$$

$$\approx \mathbf{v}_{king} - (\mathbf{v}_{man} - \mathbf{v}_{woman}). \tag{3}$$

The analogy-based transfer function is

$$f_{\mathbf{Z}}(\mathbf{v}_x) = \begin{cases} \mathbf{v}_x - \mathbf{d} & \text{if} \quad x \in \mathcal{M}, \\ \mathbf{v}_x + \mathbf{d} & \text{if} \quad x \in \mathcal{F}, \end{cases} \tag{4}$$

where $\mathcal{M}$ is a set of words with a particular target attribute (e.g., male) and $\mathcal{F}$ is a set of words with an inverse attribute (e.g., female). $\mathbf{d}$ is a difference vector, such as $\mathbf{v}_{man} - \mathbf{v}_{woman}$. Eq. 4 indicates that the operation changes depending on whether the input word x belongs to $\mathcal{M}$ or $\mathcal{F}$. However, to transfer the word attribute based on analogy, we require explicit knowledge such as the attribute value ($\mathcal{M}$, $\mathcal{F}$, or others) that is contained by the input word.

4 Reflection-based Word Attribute Transfer

4.1 Ideal Transfer Mapping without Knowledge

What is an ideal transfer function $f_{\mathbf{z}}$ for the word attribute transfer? The following are the ideal natures of a transfer function:

$$\forall (m, w, \mathbf{z}) \in \mathcal{A}, \quad \mathbf{v}_m = f_{\mathbf{z}}(\mathbf{v}_w), \tag{5}$$

$$\forall (m, w, \mathbf{z}) \in \mathcal{A}, \quad \mathbf{v}_w = f_{\mathbf{z}}(\mathbf{v}_m), \tag{6}$$

$$\forall (u, \mathbf{z}) \in \mathcal{N}, \quad \mathbf{v}_u = f_{\mathbf{z}}(\mathbf{v}_u). \tag{7}$$

The function $f_{\mathbf{z}}$ enables a word to be transferred without explicit knowledge because the operation of $f_{\mathbf{z}}$ does not change depending on whether the input word belongs to $\mathcal{M}$ or $\mathcal{F}$. By combining Eqs. 5, 6 and 7, we obtain the following formulas:

$$\forall (m, w, \mathbf{z}) \in \mathcal{A}, \quad \mathbf{v}_m = f_{\mathbf{z}}(\, f_{\mathbf{z}}(\mathbf{v}_m)\,), \tag{8}$$

$$\forall (m, w, \mathbf{z}) \in \mathcal{A}, \quad \mathbf{v}_w = f_{\mathbf{z}}(\, f_{\mathbf{z}}(\mathbf{v}_w)\,), \tag{9}$$

$$\forall (u, \mathbf{z}) \in \mathcal{N}, \quad \mathbf{v}_u = f_{\mathbf{z}}(\, f_{\mathbf{z}}(\mathbf{v}_u)\,). \tag{10}$$

Hence, the ideal transfer function is a mapping that becomes an identity mapping when we apply it twice for any $\mathbf{v}$. Such a mapping is called *involution* in geometry. For example, $f\colon \mathbf{v} \mapsto -\mathbf{v}$ is an example of an involution.

4.2 Reflection

Reflection $\mathrm{Ref}_{\mathbf{a},\mathbf{c}}$ is an ideal function because this mapping is an involution, as shown below:

$$\forall \mathbf{v} \in \mathbb{R}^n, \quad \mathbf{v} = \mathrm{Ref}_{\mathbf{a},\mathbf{c}}(\, \mathrm{Ref}_{\mathbf{a},\mathbf{c}}(\mathbf{v})\,). \tag{11}$$

Reflection reverses the location between two vectors in a Euclidean space through an affine hyperplane called a *mirror*. $\mathbf{a}$ and $\mathbf{c}$ are parameters that determine the mirror. $\mathbf{a} \in \mathbb{R}^n$ is a vector orthogonal to the mirror and $\mathbf{c} \in \mathbb{R}^n$ is a point through which the mirror passes. Reflection is

different from inverse mapping. When m and w are paired words, reflection can transfer $\mathbf{v}_m$ and $\mathbf{v}_w$ between each other with identical reflection mapping as shown in Eqs. 5 and 6; however, an inverse mapping cannot perform this action. Given a vector $\mathbf{v}$ in the Euclidean space $\mathbb{R}^n$, the formula for the reflection in the mirror is given by

$$\mathrm{Ref}_{\mathbf{a},\mathbf{c}}(\mathbf{v}) = \mathbf{v} - 2\frac{(\mathbf{v}-\mathbf{c})\cdot\mathbf{a}}{\mathbf{a}\cdot\mathbf{a}}\mathbf{a}. \tag{12}$$

4.3 Proposed Method: Reflection-based Word Attribute Transfer

Reflection by a Single Mirror We apply reflection to the word attribute transfer process. We learn a mirror (hyperplane) in a pretrained embedding space using training word pairs with binary attribute $\mathbf{z}$ (Fig. 2). Because the mirror is uniquely determined by two parameter vectors, $\mathbf{a}$ and $\mathbf{c}$, we estimate $\mathbf{a}$ and $\mathbf{c}$ from the target attribute $\mathbf{z}$ using fully connected multilayer perceptrons (MLPs):

$$\mathbf{a} = \mathrm{MLP}_{\theta_1}(\mathbf{z}), \tag{13}$$

$$\mathbf{c} = \mathrm{MLP}_{\theta_2}(\mathbf{z}), \tag{14}$$

where θ is a set of trainable parameters of MLP_θ. The transferred vector $\mathbf{v}_y$ is obtained by inverting the attribute $\mathbf{z}$ of $\mathbf{v}_x$ by reflection:

$$\mathbf{v}_y = \mathrm{Ref}_{\mathbf{a},\mathbf{c}}(\mathbf{v}_x). \tag{15}$$

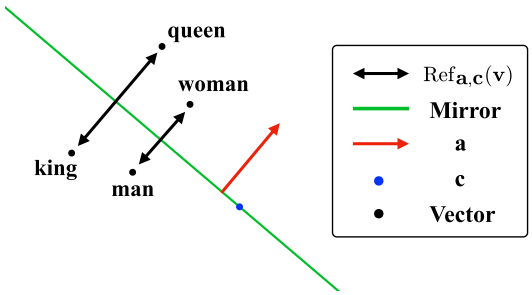


Fig. 2 Reflection-based word attribute transfer with a single mirror

Reflection by Parameterized Mirrors Reflection with a mirror according to Eqs. 13 and 14 assumes a single mirror that only depends on $\mathbf{z}$. The previous discussion assumed pairs that share a stable pair, such as *king* and *queen*.

However, as gender-variant words often do not come in pairs, gender is not sufficiently stable to be modeled by a single mirror. For example, although *actress* is exclusively feminine, *actor* is clearly neutral in several cases. Thus, *actor* is not a masculine counterpart such as *king*. In fact, bias exists in gender words in the embedding space (Zhao et al. 2018; Kaneko and Bollegala 2019). This phenomenon can occur not only with gender attributes but also with other attributes. The assumption of a single mirror forces the mirror to be a hyperplane that goes through the midpoints for all word vector pairs. However, the vector pair *actor-actress*, shown on the right in Fig. 3, cannot be transferred well as the single mirror (the green line) does not satisfy this constraint owing to the bias of the embedding space. To solve this problem, we propose *parameterized mirrors* based on the idea of using different mirrors for different words. We define the mirror parameters $\mathbf{a}$ and $\mathbf{c}$ using the word vector $\mathbf{v}_x$ to be transferred in addition to the attribute vector $\mathbf{z}$:

$$\mathbf{a} = \mathrm{MLP}_{\theta_1}([\mathbf{z}; \mathbf{v}_x]), \tag{16}$$

$$\mathbf{c} = \mathrm{MLP}_{\theta_2}([\mathbf{z}; \mathbf{v}_x]), \tag{17}$$

where $[\cdot;\cdot]$ indicates the vector concatenation in the column. *Parameterized mirrors* are expected to work more flexibly on different words than a single mirror because *parameterized mirrors* dynamically determine similar mirrors for similar words. For instance, as shown in Fig. 3, let us assume that we learned the mirror (the blue line) that transfers $\mathbf{v}_{hero}$ to $\mathbf{v}_{heroine}$ in advance. If the input word vector $\mathbf{v}_{actor}$ resembles $\mathbf{v}_{hero}$, a mirror that resembles the one for $\mathbf{v}_{hero}$ should

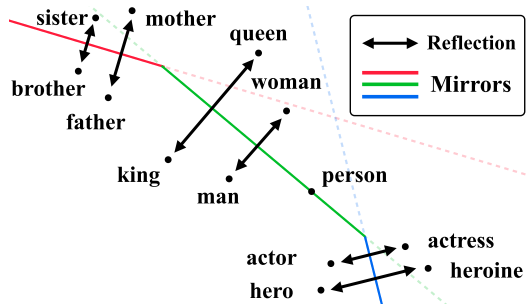


Fig. 3 Reflection using parameterized mirrors

be derived and used for the transfer. Conversely, the reflection works as an identity mapping for a vector on the mirror (e.g., $\mathbf{v}_{person}$ in Fig. 3). That is, the proposed method assumes that invariant word vectors are located on the mirror. Because we used a 300-dimensional embedded space in the experiment, we assume that the invariant word vector exists in a 299-dimensional subspace.

It should be noted that Eq. 11 may not hold for parameterized mirrors. In the reflection with a single mirror, it is true that $\mathbf{v} = \mathrm{Ref}_{\mathbf{a},\mathbf{c}}(\mathrm{Ref}_{\mathbf{a},\mathbf{c}}(\mathbf{v}))$. However, this is not guaranteed with the $\mathbf{v}$-parameterized reflection $\mathrm{Ref}_{\mathbf{a_v},\mathbf{c_v}}(\mathbf{v})$. This is because the mirror parameters $\mathbf{a_v}$ and $\mathbf{c_v}$ depend on an input word vector, as shown in Eqs. 16 and 17. Thus, we exclude this constraint and employ the constraints given by Eqs. 5–7 for our loss function.

Weight Sharing In neural networks, weight sharing can reduce the number of trainable weights and often improve performance (Yang et al. 2018; Lample et al. 2018). The mirror parameters $\mathbf{a}$ and $\mathbf{c}$ can be defined using a shared MLP as follows:

$$\mathbf{o} = \mathrm{MLP}_\theta([\mathbf{z}; \mathbf{v}_x]), \tag{18}$$

$$\mathbf{a} = \mathbf{W}_a\mathbf{o}, \tag{19}$$

$$\mathbf{c} = \mathbf{W}_c\mathbf{o}, \tag{20}$$

where θ indicates the shared weights. $\mathbf{o} \in \mathbb{R}^m$ is an output vector of MLP_θ. $\mathbf{W}_a \in \mathbb{R}^{n\times m}$ and $\mathbf{W}_c \in \mathbb{R}^{n\times m}$ are weight matrices corresponding to $\mathbf{a}$ and $\mathbf{c}$, respectively.

Loss Function The following properties must be satisfied in word attribute transfer: (1) words with attribute $\mathbf{z}$ are transferred and (2) words without it are not transferred. Thus, loss $L(\Theta)$ is defined as:

$$L(\Theta) = \frac{1}{|\mathcal{A}|}\sum_{(x,t,\mathbf{Z})\in\mathcal{A}}(\mathbf{v}_y - \mathbf{v}_t)^2 + \frac{1}{|\mathcal{N}|}\sum_{(x,\mathbf{Z})\in\mathcal{N}}(\mathbf{v}_y - \mathbf{v}_x)^2,, \tag{21}$$

where Θ is a set of trainable parameters ($\Theta = \{\theta\}$ for weight sharing and $\Theta = \{\theta_1, \theta_2\}$ otherwise). The first term draws the target word vector $\mathbf{v}_{t_i}$ closer to the corresponding transferred vector $\mathbf{v}_{y_i}$ and the second term prevents words that are invariant with respect to a target attribute from being moved by the transfer function. $\mathbf{v}_y$ is the output of a reflection (Eq. 15).

5 Experiment

We evaluated the performance of word attribute transfer using data with four different attributes. We used 300-dimensional word2vec[2] and GloVe[3] models as the pretrained word embedding. We used four different datasets of word pairs with four binary attributes: Male-Female (MF), Singular-Plural (SP), Capital-Country (CC), and Antonym (AN) (Table 1). These word pairs were collected from analogy test sets (Mikolov et al. 2013a; Gladkova et al. 2016) and the Internet. Antonyms were obtained from the literature (Nguyen et al. 2017). Their datasets were collected from WordNet (Miller 1995) and Wordnik [4]. The original data by Nguyen et al. (2017) contains synonyms; however, we excluded them and used only the antonyms. We compared the models that train with attributes individually with the models that train with joint attributes. The invariant word dataset $\mathcal{N}$ were constructed by random sampling from WordNet by excluding the attribute-variant words in the corresponding set $\mathcal{A}$. We sampled the invariant words for the invariant portion of the training data by varying their occupancy, i.e., 0, 5, 10, 25, and 50%, to investigate their effects on the tradeoffs between variant and invariant words. We also chose 1,000 invariant words for the test ($|\mathcal{N}_{\text{test}}|$ =1,000).

5.1 Evaluation Metrics

We measured the accuracy and stability performances of the word attribute transfer. The accuracy measures the number of input words in $\mathcal{A}_{\text{test}}$ that were transferred correctly to the corresponding target words. The stability score measures the number of words in $\mathcal{N}_{\text{test}}$ that were not mapped to other words. For example, in the MF transfer, given *man*, the transfer is regarded as correct if *woman* is the closest word to the transferred vector; otherwise, it is incorrect. Given *person*, the transfer is regarded as correct if *person* is the closest word to the transferred vector;

Dataset $\mathcal{A}$	**#Train**	**#Val**	**#Test**	**#Total**
Male-Female (MF)	106	48	48	202
Singular-Plural (SP)	3624	776	776	5176
Capital-Country (CC)	118	50	50	218
Antonym (AN)	5002	642	642	6286

Table 1 Statistics of binary-attribute word datasets

[2] https://code.google.com/archive/p/word2vec/
[3] https://nlp.stanford.edu/projects/glove/
[4] https://www.wordnik.com/

otherwise, it is incorrect. The accuracy and stability scores are calculated using the following formula:

$$\delta(\mathbf{v}_y, t) = \begin{cases} 1 & \text{if} \quad \arg\max_{k \in \mathcal{V}}(\cos(\mathbf{v}_y, \mathbf{v}_k)) = t \\ 0 & \text{otherwise,} \end{cases} \tag{22}$$

$$\text{Accuracy} = \frac{1}{|\mathcal{A}_{\text{test}}|} \sum_{(x,t,\mathbf{Z}) \in \mathcal{A}_{\text{test}}} \delta(\mathbf{v}_y, t), \tag{23}$$

$$\text{Stability} = \frac{1}{|\mathcal{N}_{\text{test}}|} \sum_{(x,\mathbf{Z}) \in \mathcal{N}_{\text{test}}} \delta(\mathbf{v}_y, x), \tag{24}$$

where $\mathcal{V}$ is the vocabulary of the word embedding model and $\cos(\mathbf{v}_y, \mathbf{v}_k)$ is the cosine similarity measure, which is defined as $\cos(\mathbf{v}_y, \mathbf{v}_k) = \frac{\mathbf{v}_y \cdot \mathbf{v}_k}{\|\mathbf{v}_y\| \|\mathbf{v}_k\|}$.

For the accuracy evaluation in the AN transfer, we used a different definition, as presented subsequently, to evaluate the accuracy because there are multiple possible candidates for the transfer in the AN dataset.

$$\delta_{\text{AN}}(\mathbf{v}_y, t) = \begin{cases} 1 & \text{if} \quad \arg\max_{k \in \mathcal{V}}(\cos(\mathbf{v}_y, \mathbf{v}_k)) \in \mathcal{T}, \\ 0 & \text{otherwise,} \end{cases} \tag{25}$$

$$\text{Accuracy}_{\text{AN}} = \frac{1}{|\mathcal{A}_{\text{test}}|} \sum_{(x,\mathcal{T},\mathbf{Z}) \in \mathcal{A}_{\text{test}}} \delta_{\text{AN}}(\mathbf{v}_y, \mathcal{T}), \tag{26}$$

where $\mathcal{T} = \{t_1, t_2, ..., t_3\}$ is a set of target words of the input antonym word x.

5.2 Methods and Configurations

Because these datasets are significantly small, we added 300-dimensional Gaussian noise to every input vector during training to avoid overfitting, i.e., $\mathbf{v}_x + \mathbf{g}_\sigma$, where $\mathbf{g}_\sigma$ is the Gaussian noise and σ is the standard deviation of the Gaussian distribution. The reported results are presented for the best set of hyperparameters evaluated on the validation set for each model after a grid search on the following values: Adam (Kingma and Ba 2015) learning rate $\alpha \in \{0.0001, 0.00015, 0.001, 0.0015\}$ (the other hyperparameters were the same as the original hyperparameters), $\sigma \in \{0.0, 0.05, 0.1, 0.15, 0.2\}$, and MLP inner hidden size $\in \{300, 500, 1500, 3000\}$. Table 2 lists the best hyperparameters of the proposed method. We did not use regularization methods such as dropout (Srivastava et al. 2014) or batch normalization (Ioffe and Szegedy 2015) because they did not show any improvement in our pilot test.

In training, the attribute and invariant word data were combined into one training dataset, where an invariant word $(x, \mathbf{z}) \in \mathcal{N}$ was represented as $(x, x, \mathbf{z}) \in \mathcal{N}$, similar to an attribute

Embedding	Hyperparameters	MF	SP	CC	AN
	Model	REF+PM	REF+PM+SHARE	REF+PM	REF+PM
	Batch size	512	512	512	4096
	Best Iterations	21000	14000	20000	20000
	Noise σ	0.1	0.1	0.1	0.1
word2vec	**Adam α**	0.0001	0.0001	0.0001	0.0001
	Activation Function	ReLU	ReLU	ReLU	ReLU
	Num of MLP layers	3	5	3	5
	Inner hidden size of MLP	300	1500	300	3000
	Size of $\lvert\mathcal{N}_{\text{train}}\rvert$	5%	25%	5%	50%
	Model	REF+PM+SHARE	REF+PM+SHARE	REF+PM	REF+PM
.	**Batch size**	512	512	512	4096
	Best Iterations	48000	20000	24000	30000
	Noise σ	0.1	0.1	0.1	0.1
GloVe	**Adam α**	0.0001	0.0001	0.0001	0.0001
	Activation Function	ReLU	ReLU	ReLU	ReLU
	Num of MLP layers	3	5	3	5
	Inner hidden size of MLP	300	1500	300	3000
	Size of $\lvert\mathcal{N}_{\text{train}}\rvert$	25%	10%	5%	10%

Table 2 Hyperparameters for reflection-based word attribute transfer

word $(x, t, \mathbf{z}) \in \mathcal{A}$. Thus, we could simply implement the loss function (Eq. 21) as follows: $L(\Theta) = \frac{1}{|\mathcal{A}\cap\mathcal{N}|}\sum_{(x,t,\mathbf{z})\in\mathcal{A}\cap\mathcal{N}}(\mathbf{v}_y - \mathbf{v}_t)^2$, where $\mathcal{A}\cap\mathcal{N}$ is a mini-batch of training data.

In our experiment, we compared our proposed method with the following baseline methods:

Ref This is a reflection-based word attribute transfer with a single mirror. We used a fully connected MLP with a rectified linear unit (ReLU) (Glorot et al. 2011) to estimate **a** and **c**.

Ref+PM This is a reflection-based word attribute transfer with *parameterized mirrors*. We used the same architecture of MLP as REF.

Ref+PM+Share This method consists of a reflection-based word attribute transfer with *parameterized mirrors*. We used the MLP with shared weights to estimate **a** and **c** (refer to Weight Sharing in Section 4.3).

MLP This is a fully connected MLP with ReLU: $\mathbf{v}_y = \text{MLP}([\mathbf{v}_x; \mathbf{z}])$. The highest accuracy models are a five-layer MLP with 1500 hidden units for SP, five-layer MLP with 3000 hidden units for AN, and three-layer MLP with 300 hidden units for the other datasets. The optimal configurations were as follows: the learning rate for Adam $\alpha = 0.00015$ for all datasets; moreover, $\sigma = 0.05$ for AN and $\sigma = 0.1$ for the other datasets.

TransE The word attribute transfer task is similar to link prediction in which (x, z, t) is replaced

with (head, label, tail). In link prediction, given a set of triplets (head, label, tail), the knowledge graph embedding model predicts the tail from the head and label. We applied TransE (Bordes et al. 2013), a baseline model for knowledge graph embeddings, to word attribute transfer. We modified the model to input the word vector into the knowledge graph embedding model based on the following equations:

$$\mathbf{h} = \mathbf{W}_{\text{head}}\mathbf{v}_x, \tag{27}$$

$$\mathbf{t} = \mathbf{W}_{\text{tail}}\mathbf{v}_t, \tag{28}$$

where $\mathbf{h} \in \mathbb{R}^k$ is a head vector and $\mathbf{t} \in \mathbb{R}^k$ is a tail vector. $\mathbf{W}_{\text{head}\in\mathbb{R}^{k\times n}}$ and $\mathbf{W}_{\text{tail}\in\mathbb{R}^{k\times n}}$ are weight matrices corresponding to the head and tail, respectively. The label vector $\mathbf{l}$ was embedded in the same way as the original TrasnE based on a set of relations $\{\text{MF}, \text{SP}, \text{CC}, \text{AN}\}$. The optimal configurations were as follows: the latent dimension $k = 200$, learning rate λ for stochastic gradient descent $\lambda = 1.0$, and margin $\gamma = 5.0$. TransE was implemented using an open toolkit for knowledge embedding called OpenKE (Han et al. 2018). In the evaluation, when calculating the accuracy and stability in Eqs. 22 and 25, the score function of TransE was used instead of $\cos(\mathbf{v}_y, \mathbf{v}_k)$.

Diff This method consists of analogy-based word attribute transfer with a difference vector, $\mathbf{d} = \mathbf{v}_m - \mathbf{v}_w$, where m and w are in the training data of $\mathcal{A}$. We chose the $\mathbf{d}$ that achieved the best accuracy in the validation data of $\mathcal{A}$. We determined whether to add or subtract $\mathbf{d}$ to $\mathbf{v}_x$ based on explicit knowledge (Eq. 4). Here, DIFF$^+$ and DIFF$^-$ transfer word attributes using a difference vector regardless of the explicit knowledge. $^+$ and $^-$ add or subtract the difference vector to any input word vector.

MeanDiff This method includes an analogy-based word attribute transfer with a mean difference vector $\bar{\mathbf{d}}$, where $\bar{\mathbf{d}} = \frac{1}{|\mathcal{A}_{\text{train}}|}\sum_{(m_i,w_i,\mathbf{z})\in\mathcal{A}_{\text{train}}}(\mathbf{v}_{m_i} - \mathbf{v}_{w_i})$. We determined whether to add or subtract $\bar{\mathbf{d}}$ to $\mathbf{v}_x$ based on the explicit knowledge (Eq. 4).

5.3 Evaluation of Accuracy and Stability

Table 3 lists the accuracy and stability results. Because AN has a many-to-many relationship, a single difference vector cannot be obtained. Therefore, the analogy methods (DIFF, DIFF^{+-}, MEANDIFF, and MEANDIFF^{+-}) were not applied to AN. Different pretrained word embeddings by GloVe and word2vec provided similar results. REF+PM and REF+PM+SHARE achieved the best accuracy among the methods that did not use explicit attribute knowledge. For example, the accuracy of REF+PM was 74% for CC; however, the accuracy of MLP was 18%. For most

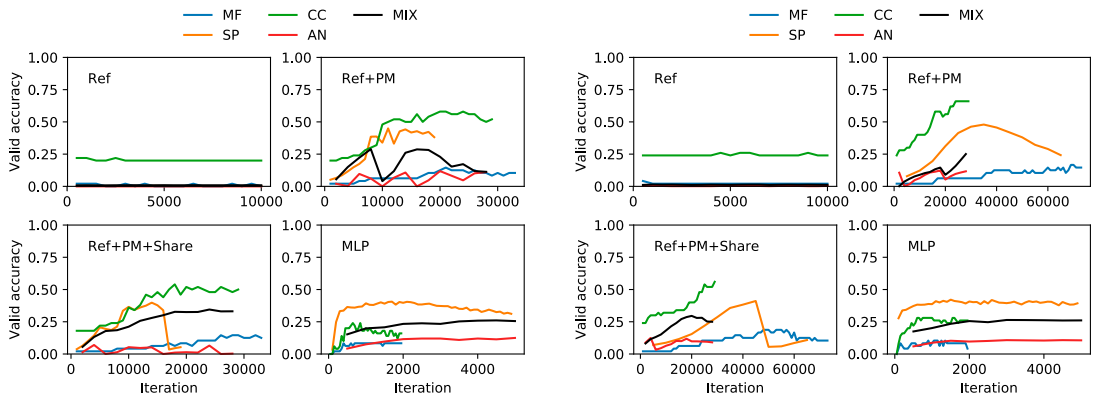
Embedding	Method	Knowledge	Accuracy (%)				Stability (%)			
			MF	**SP**	**CC**	**AN**	**MF**	**SP**	**CC**	**AN**
word2vec	Ref (individual)		22.9	0.5	44.0	0.2	**100.0**	**100.0**	**100.0**	**100.0**
	Ref+Share (individual)		22.9	0.5	42.0	0.2	**100.0**	**100.0**	**100.0**	**100.0**
	Ref+PM (individual)		**41.7**	44.2	**62.0**	11.2	98.5	95.9	**100.0**	75.2
	Ref+PM+Share (individual)		37.5	43.0	60.0	7.2	99.3	98.7	**100.0**	96.2
	MLP (individual)		10.4	40.1	18.0	**12.5**	5.7	95.1	9.2	92.4
	TransE (individual)		0.0	0.2	0.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**
	Ref (joint)		18.8	0.3	42.0	0.2	**100.0**	**100.0**	**100.0**	**100.0**
	Ref+Share (joint)		18.8	0.3	44.0	0.2	**100.0**	**100.0**	**100.0**	**100.0**
	Ref+PM (joint)		25.0	43.4	44.0	13.2	**100.0**	93.7	**100.0**	63.7
	Ref+PM+Share (joint)		18.8	**50.8**	34.0	16.0	**100.0**	98.8	**100.0**	89.0
	MLP (joint)		16.7	38.1	8.0	14.0	95.4	98.6	97.6	97.1
	TransE (joint)		0.0	0.3	0.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**
	Diff $^+$		22.9	3.2	32.0	—	97.1	93.1	89.5	—
	Diff $^-$		22.9	3.1	32.0	—	87.9	98.8	99.4	—
	MeanDiff $^+$		6.3	0.3	22.0	—	**100.0**	**100.0**	99.7	—
	MeanDiff $^-$		8.3	0.3	14.0	—	**100.0**	**100.0**	99.9	—
	Diff	✓	37.5	6.3	64.0	—	—	—	—	—
	MeanDiff	✓	14.6	0.5	36.0	—	—	—	—	—
GloVe	Ref (individual)		10.4	0.5	24.0	0.2	**100.0**	**100.0**	**100.0**	**100.0**
	Ref+Share (individual)		10.4	0.4	24.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**
	Ref+PM (individual)		37.5	45.0	**74.0**	12.3	99.2	99.3	**100.0**	93.3
	Ref+PM+Share (individual)		**39.6**	42.8	72.0	11.5	99.2	99.8	**100.0**	94.7
	MLP (individual)		14.6	41.1	18.0	**14.2**	41.7	97.6	50.3	93.4
	TransE (individual)		0.0	0.2	0.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**
	Ref (joint)		12.5	0.4	24.0	0.2	**100.0**	**100.0**	**100.0**	**100.0**
	Ref+Share (joint)		2.1	0.4	20.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**
	Ref+PM (joint)		12.5	39.8	36.0	9.0	**100.0**	99.7	**100.0**	94.1
	Ref+PM+Share (joint)		12.5	**47.0**	36.0	9.8	**100.0**	97.7	**100.0**	59.0
	MLP (joint)		27.1	35.2	26.0	11.1	98.0	99.7	99.4	97.6
	TransE (joint)		0.0	0.3	0.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**
	Diff $^+$		14.6	4.5	22.0	—	**100.0**	**100.0**	**100.0**	-
	Diff $^-$		12.5	4.3	26.0	—	**100.0**	99.8	99.9	—
	MeanDiff $^+$		0.0	0.3	2.0	—	**100.0**	**100.0**	**100.0**	—
	MeanDiff $^-$		0.0	0.3	4.0	—	**100.0**	**100.0**	**100.0**	—
	Diff	✓	27.1	8.7	48.0	—	—	—	—	-
	MeanDiff	✓	0.0	0.5	6.0	—	—	—	—	—

Table 3 Results of accuracy and stability scores. MF, SP, CC, and AN are datasets. Here, "joint" models are trained with joint attributes and "individual" models are trained with an individual attribute.

attributes, our proposed methods outperformed the analogy-based transfer. Weight sharing did not significantly improve the performance of the proposed methods. The parameterized mirror improved the performance of a reflection-based transfer, although the learning was unstable (Fig. 4). For stability, reflection-based transfers achieved superior stability scores, which exceeded

93% in most cases. We mixed all the attribute datasets and trained models. The best model was the proposed method trained with an individual attribute dataset. In the joint condition, the MLP demonstrated better performance than that in the individual attribute condition with the help of the larger training data. The results show that our proposed methods transfer an input word if it has a target attribute and does not transfer an input word with better scores than the baseline methods, even though the proposed methods do not use knowledge of the input words.

In TransE, the accuracy was almost 0% for all the attributes.[5] However, in the MF, SP, and CC relations, several reference words were within the top three nearest neighbors, as listed in Table 4. This result is similar not only in this task but also when learning with WN18 (Trouillon et al. 2016). This is owing to the nature of TransE. While considering AN, the accuracy in the three nearest neighbors was still low. This can be explained by the poor performance of TransE



(a) word2vec (b) GloVe

Fig. 4 Visualization of validation accuracy

Method	word2vec								GloVe							
	MF		SP		CC		AN		MF		SP		CC		AN	
	@1	@3	@1	@3	@1	@3	@1	@3	@1	@3	@1	@3	@1	@3	@1	@3
TransE (individual)	0.0	75.0	0.2	76.3	0.0	80.0	0.0	20.2	0.0	66.7	0.2	80.0	0.0	76.0	0.0	42.2
TransE (joint)	0.0	72.9	0.3	70.7	0.0	64.0	0.0	20.0	0.0	64.6	0.3	75.6	0.0	72.0	0.0	36.4

Table 4 Accuracy of the top three nearest neighbors of TransE. A "joint" model is trained with joint attributes. An "individual" model is trained with an individual attribute.

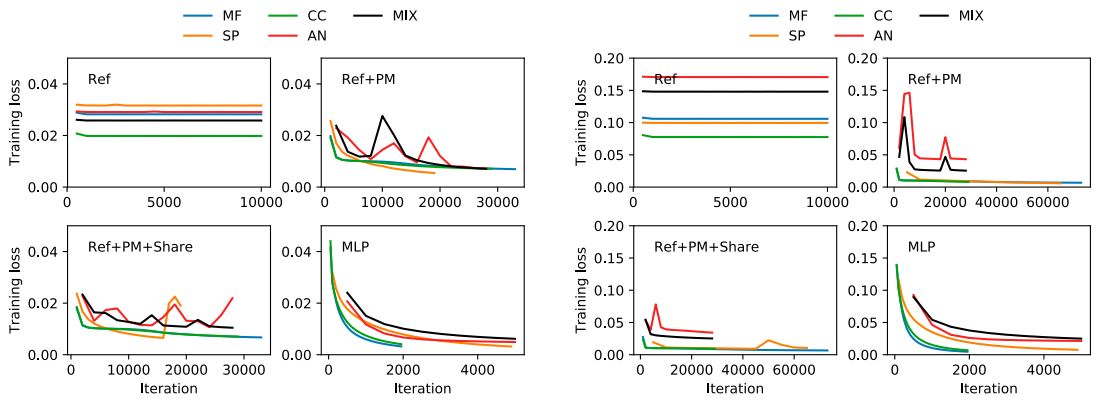
[5] We also experimented with ComplEx (Trouillon et al. 2016), which is a knowledge graph embedding model in a complex space; however, it is not described in this paper because it is not comparable because of its low accuracy.

for one-to-many or many-to-many relationships (Bordes et al. 2013).

In the individual attribute condition, MLP worked poorly, especially in terms of stability for MF and CC, while it showed better transfer accuracy for AN than the proposed method. We reviewed the training curves resulting from the MLP, which are shown in Fig. 4 and 5; however, they showed reasonable convergence. This would be due to the training data size in the individual attribute condition, because MLP stability significantly improves in the joint condition.

We also investigated the tradeoff between transfer accuracy and stability by changing the size of the invariant words and the stability of the learning-based methods by conducting an additional experiment that varied $|\mathcal{N}_{\text{train}}|$. The large size of $\mathcal{N}_{\text{train}}$ is expected to increase the stability; however, it may also decrease the accuracy. The stability scores demonstrated by the MLP did not improve (Table 5) for MF and CC. Conversely, the proposed methods achieved high stability scores with $|\mathcal{N}_{\text{train}}| = 5\%$ and maintained the accuracy. We hypothesized that the high stability was owing to the distance between the word and its mirror. If invariant words are distributed on the mirror, they will not be transferred. We investigated the distance between the input word vector $\mathbf{v}_x$ and its mirror (Fig. 6). The result showed that invariant words were close to the mirror and attribute words were distributed away from it. If the distance between paired words is significantly small, the distance between the word and its mirror is also small. Fig. 7 shows the distribution of the distance between the input $\mathbf{v}_x$ and the target word vector $\mathbf{v}_t$. The distance between paired words for MF and SP is considerably smaller than that for CC and AN.

Although analogy-based methods achieved high stability, their accuracy results were low. In particular, the MeanDiff$^+$ and MeanDiff$^-$ did not change the original vector. We hypothe-



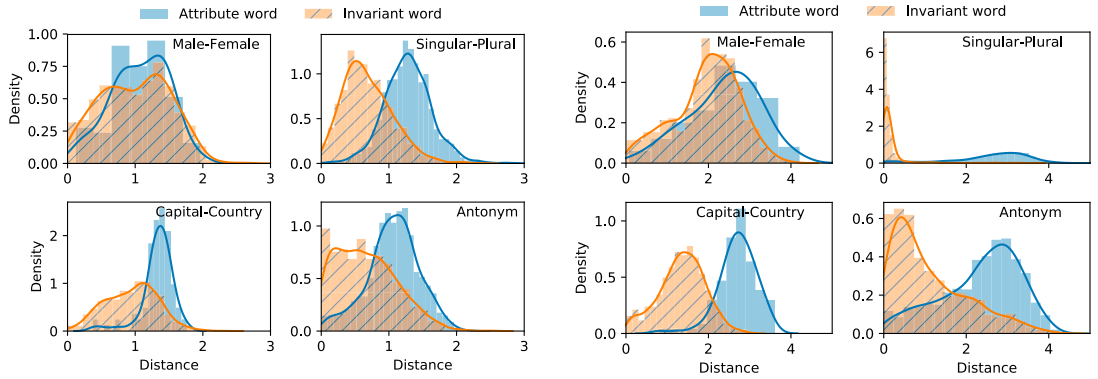
(a) word2vec (b) GloVe

Fig. 5 Visualization of training loss

Embedding		Method	Accuracy (%)					Stability (%)				
			$\lvert\mathcal{N}_{\text{train}}\rvert$					$\lvert\mathcal{N}_{\text{train}}\rvert$				
			0%	5%	10%	25%	50%	0%	5%	10%	25%	50%
word2vec	MF	Ref	18.8	20.8	22.9	22.9	18.8	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
		Ref+PM	35.4	**41.7**	**37.5**	35.4	25.0	86.5	98.5	99.6	99.7	91.8
		Ref+PM+Share	**37.5**	31.2	35.4	**37.5**	**29.2**	78.6	99.4	99.5	99.3	99.8
		MLP	4.2	6.2	8.3	8.3	10.4	0.0	0.0	0.0	0.8	5.7
	SP	Ref	0.4	0.5	0.4	0.3	0.3	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
		Ref+PM	43.3	**46.3**	**44.2**	42.4	**40.3**	53.4	82.4	95.9	99.1	99.5
		Ref+PM+Share	**44.7**	43.6	43.7	**43.0**	38.7	42.3	93.5	94.5	98.7	98.4
		MLP	42.0	41.0	40.1	36.7	36.0	66.8	86.0	95.1	98.1	99.6
	CC	Ref	34.0	34.0	36.0	38.0	44.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
		Ref+PM	**62.0**	**62.0**	54.0	54.0	50.0	90.0	**100.0**	**100.0**	**100.0**	99.8
		Ref+PM+Share	56.0	58.0	**58.0**	**60.0**	**56.0**	86.4	**100.0**	**100.0**	**100.0**	**100.0**
		MLP	10.0	12.0	10.0	18.0	18.0	0.0	0.0	0.0	0.6	9.2
	AN	Ref	0.0	0.2	0.0	0.0	0.2	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
		Ref+PM	12.5	12.9	12.3	11.8	11.2	26.8	26.0	34.3	65.7	75.2
		Ref+PM+Share	13.9	12.8	12.1	12.0	7.2	7.7	20.8	49.7	71.4	96.2
		MLP	**17.0**	**15.4**	**15.1**	**12.5**	**14.2**	1.2	6.2	36.6	92.4	67.2
GloVe	MF	Ref	10.4	4.2	6.2	4.2	2.1	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
		Ref+PM	**37.5**	**39.6**	37.5	37.5	**35.4**	89.3	93.2	95.7	99.2	99.6
		Ref+PM+Share	35.4	31.2	**39.6**	**39.6**	**35.4**	88.7	98.9	97.5	99.2	99.6
		MLP	4.2	12.5	6.2	8.3	14.6	0.0	0.0	0.0	0.3	41.7
	SP	Ref	0.5	0.4	0.5	0.4	0.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
		Ref+PM	**46.3**	**46.6**	**46.4**	44.6	**45.0**	54.1	94.5	97.8	98.9	99.3
		Ref+PM+Share	43.9	43.7	44.8	**45.0**	42.8	52.6	95.5	98.1	99.4	99.8
		MLP	42.7	41.0	41.1	38.9	36.9	70.0	95.2	97.6	99.3	99.8
	CC	Ref	22.0	24.0	24.0	22.0	20.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
		Ref+PM	66.0	**74.0**	**70.0**	70.0	**74.0**	99.9	**100.0**	99.9	**100.0**	99.9
		Ref+PM+Share	**70.0**	70.0	**70.0**	**72.0**	72.0	99.8	99.9	99.9	99.9	**100.0**
		MLP	8.0	6.0	8.0	6.0	18.0	0.0	0.0	0.0	1.3	50.3
	AN	Ref	0.2	0.2	0.0	0.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
		Ref+PM	12.1	13.2	12.3	10.9	10.6	16.0	78.0	93.3	96.1	97.1
		Ref+PM+Share	11.7	12.5	10.7	11.5	8.4	14.0	42.4	73.9	94.7	96.8
		MLP	**16.6**	**14.5**	**16.0**	**14.2**	**11.7**	2.1	53.1	70.5	93.4	97.9

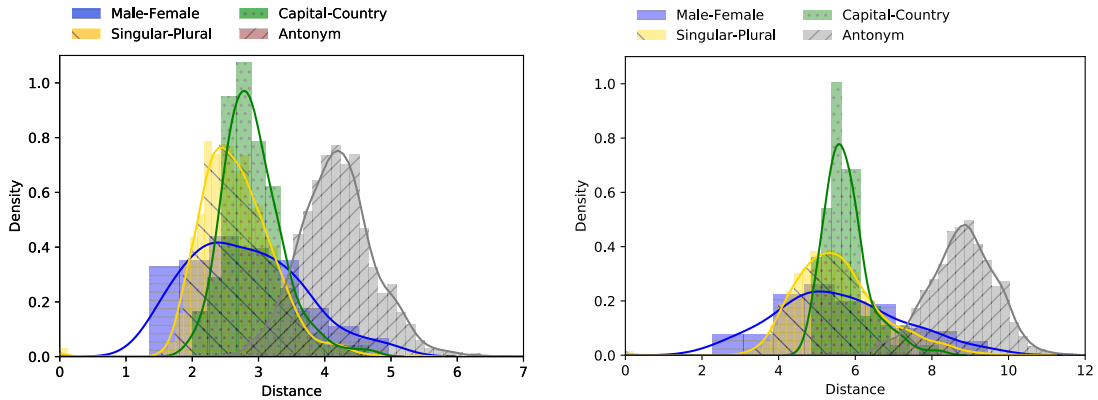
Table 5 Relation between the size of $|\mathcal{N}_{\text{train}}|$ and the stability of methods trained with an individual attribute

sized that the result can be attributed to the significantly small L2 norm of the mean difference vector $\bar{\mathbf{d}}$. Table 6 lists the relationship between the MeanDiff performances and the L2 norm of the mean difference vector. The stability was high because the original vector was almost unchanged even if $\mathbf{d}$ was added or subtracted. Conversely, when the L2 norm of $\bar{\mathbf{d}}$ was large, the accuracy became high as the difference vectors were similar to each other.



(a) word2vec (b) GloVe

Fig. 6 Distribution of distance between the input word vector and its mirror $\frac{|(\mathbf{v}_x - \mathbf{c}) \cdot \mathbf{a}|}{\|\mathbf{a}\|}$ learned by REF+PM. It can be observed that invariant words are close to the mirror and attribute words are distributed away from it.



(a) word2vec (b) GloVe

Fig. 7 Distribution of distance between the input word vector and the target word vector $\|\mathbf{v}_x - \mathbf{v}_t\|$

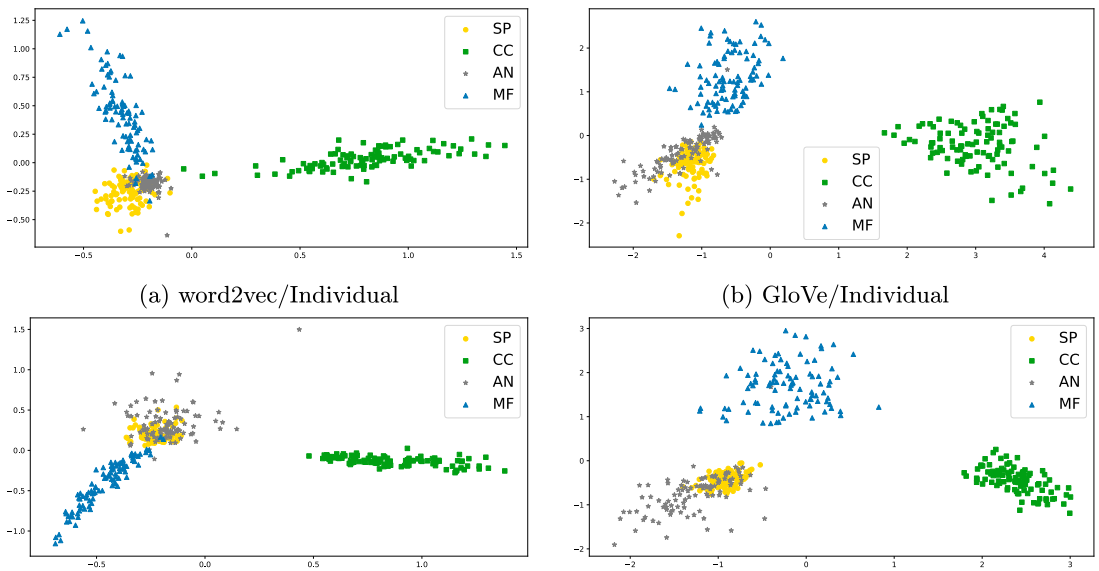
DIFF, DIFF$^+$, and DIFF$^-$ obtained high accuracy for CC and low accuracy for SP. This is due to the use of a fixed difference vector $\mathbf{d}$ in Diff. We investigated the mean cosine similarity between the difference vector $\mathbf{d}$ and other difference vectors, i.e., mean $= \frac{1}{|\mathcal{A}|} \sum_{(x,t,\mathbf{z}) \in \mathcal{A}} \cos(\mathbf{d}_{(x,t)}, \mathbf{d})$, where $\mathbf{d}_{(x,t)}$ is the difference vector of a word pair (x, t) other than $\mathbf{d}$. We found that the mean cosine similarity between SP words was almost 0% in DIFF$^-$, as listed in Table 6. Thus, when we use a single difference vector, several SP words are transferred into inappropriate words.

5.4 Visualization of Parameterized Mirrors

Fig. 8 shows the principal component analysis (PCA) results of the mirror parameter **a** obtained for the test words. We normalized the L2 norm of **a** to 1 ($\frac{\mathbf{a}}{\|\mathbf{a}\|}$). We compared the PCA results with the results of the model trained with the joint attributes and the model trained

Embedding	Attr	MeanDiff$^-$					Diff$^-$				
		Acc	Stb	L2	cos		Acc	Stb	L2	cos	
					mean	var				mean	var
word2vec	MF	8.3	100.0	1.17	0.39	0.05	22.9	97.1	2.91	0.28	0.05
	SP	0.3	100.0	0.75	0.18	0.01	3.1	98.8	3.07	0.06	0.01
	CC	14.0	99.9	1.82	0.61	0.02	32.0	99.4	2.76	0.49	0.02
GloVe	MF	0.0	100.0	2.29	0.38	0.03	14.6	100.0	4.61	0.29	0.04
	SP	0.3	100.0	1.82	0.21	0.02	4.5	100.0	5.68	0.08	0.01
	CC	4.0	100.0	3.56	0.61	0.02	22.0	100.0	6.05	0.44	0.01

Table 6 Analysis of difference vectors. **L2** is the L2 norm of the difference vector that the model used during the inference time (**d** for Diff and $\bar{\mathbf{d}}$ for MeanDiff). **cos** is the distribution of cosine similarities between the difference vector (**d** or $\bar{\mathbf{d}}$) and other difference vectors.



(a) word2vec/Individual (b) GloVe/Individual

(c) word2vec/Joint (d) GloVe/Joint

Fig. 8 Two-dimensional principal component analysis projection of the 300-dimensional mirror parameter **a**. The mirror parameters were estimated by the proposed model (Ref+PM+Share) trained by each attribute.

with an individual attribute. Similar results were obtained for both conditions. Fig. 8 suggests that the mirror parameters of the paired words are similar to each other and that those with an attribute form a cluster; words with the same attribute have similar mirror parameters, i.e., **a**.

5.5 Transfer Example

Table 7 lists the gender transfer results for a tiny example sentence. Here, the attribute transfer was applied to every word in the sentence $X = \{x_1, x_2, ...\}$. Here, because words such as *a* and *.* are not in the vocabulary of word2vec, we omitted them from the inputs. The MLP resulted in several incorrect transfers on gender-invariant words, e.g., *the* became *dear_madam*, *were* became *laundresses*, *boyfriend* became *boyfriend*, and *boy* became *mother*. Analogy-based transfers can transfer only in one direction. DIFF$^+$ could transfer if x is female, e.g., it transferred from *woman* to *man*, but it could not transfer *boy*. Similarly, DIFF$^-$ failed to transfer from female to male. Moreover, as the stability of these methods was low, they resulted in erroneous transfers. For example, in DIFF$^-$, *the* became *Sir*. REF+PM could transfer only gender words without using explicit gender information. Thus, *woman* was transferred to *man* without the knowledge that *woman* is a female word. When the gender-invariant word *married* was input, it was not changed by reflection, without the knowledge that *married* has no gender attribute.

From Table 8, it can be noted that words with different target attributes were transferred by each reflection-based transfer. Thus, when *daughter* was input for a MF transfer, it was transferred to *son* and *daughters* for SP transfer. When *Tokyo* was input for MF, SF, and AN tranfers, it was not transferred; however, it was transferred to *Japan* in the CC transfer. When *stereo* was input for MF, SP, and CC transfers, it was not transferred; however, it was transferred to *monaural* in the AN transfer.

Input words		the woman got married when you were a boy .
Methods	REF	the **man** got married when you were a **boy** .
	REF+PM	the **man** got married when you were a **girl** .
	REF+PM+SHARE	the **man** got married when you were a **girl** .
	DIFF $^+$	Sir **man** got married when you were a **boy** .
	DIFF $^-$	chairwoman **woman** got married chairwoman you were a **girl** .
	MLP	dear_madam **man** dear_madam boyfriend dear_madam lazy_slob laundresses a **mother** .

Table 7 Comparison of gender transfers. Each method transfers words in a sentence one by one.

6 Discussion

In our method, the performance of GloVe was better than that of word2vec. Table 9 lists the scores of the Google analogy test set (Mikolov et al. 2013a) for the different embedding methods, i.e., word2vec and GloVe. From in Table 9, it can be noted that the score of GloVe was higher than that of word2vec. This indicates that the embedding space of GloVe worked better than that of word2vec in this task. The performance of the word attribute transfer using reflection probably depends on the analogic space, because transferring will be easy if the pair of transferred words exists in a similar place in the analogic space.

Input words		Mr. Smith and his daughter want to visit the science museum in Tokyo to see the stereo microphone .
REF+PM **(individual)**	**MF**	**Ms.** Smith and **her son** want to visit the science museum in Tokyo to see the stereo microphone .
	SP	Mr. Smith and his **daughters** want to **visits** the science **museums** in Tokyo to see the stereo **microphones** .
	CC	Mr. Smith and his daughter want to visit the science museum in **Japan** to see the stereo microphone .
	AN	Mr. Smith and his daughter **eliminate** to visit the science museum in Tokyo to **back** the **monaural** microphone .
REF+PM **(joint)**	**MF**	**Ms.** Smith and **her son** want to visit the science museum in Tokyo to see the stereo microphone .
	SP	Mr. Smith and his **daughters dos** to **visits** the science **museums** in Tokyo to **watches** the **cassette_decks microphones** .
	CC	Mr. Smith and his daughter want to visit the science museum in **Japan** to see the stereo microphone .
	AN	Mr. Smith and his daughter want to visit the **zoology** museum in Tokyo to see the **monaural** microphone .

Table 8 Transfer of different attributes with the proposed method (REF+PM)

Embedding method	**Analogy score (%)**
word2vec	74.01
GloVe	**75.13**

Table 9 Comparison of analogy scores

7 Error Analysis

We analyzed the attribute words that could not be transferred accurately by models trained with the individual attributes. We categorized the failed output words into three error cases.

Case1 The output word was the same as the input word ($y = x$).

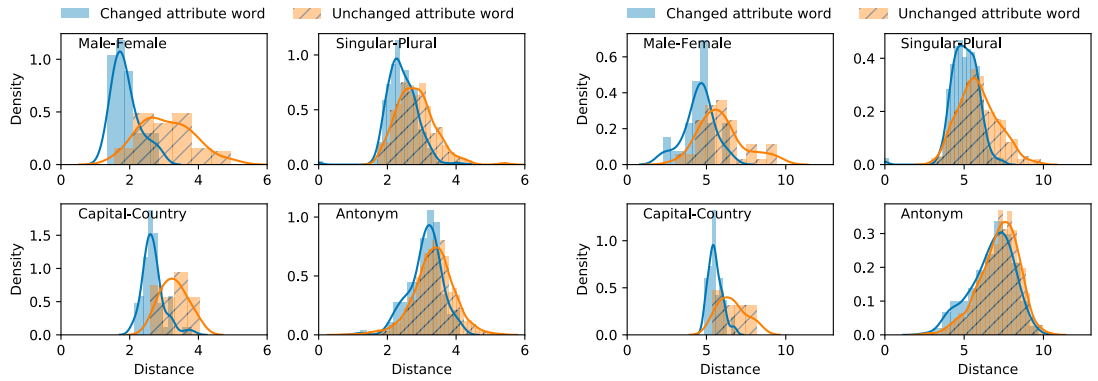
Case2 The attribute of the input word was transferred but was incorrect. For example, in CC transfer, the transfer result was *Beijing* when *Japan* was given.

Case3 Other types of errors.

Table 10 lists the results of the error analysis in word2vec. The results show that most of the failures in the reflection-based transfer was in Case1. We speculated that such unchanged attribute word pairs tended to be close to each other. Fig. 9 shows the difference in the distance between the input word and the target word $\|\mathbf{v}_x - \mathbf{v}_t\|$ in the changed and unchanged attribute word pairs. Contrary to this hypothesis, it was shown that there was no difference in the distance between the changed and unchanged pairs. Table 11 lists examples of Case2 and Case3 in the proposed method. For example, when given *stepbrother* as a gender word, the proposed method (REF+PM) did not output *stepsister* but provided the result of *stepmother*. In MF and CC

		Cases where transfer failed (%)		
		Case1	**Case2**	**Case3**
MF	REF	100.0	0.0	0.0
	REF+PM	86.0	12.0	2.0
	REF+PM+SHARE	88.0	8.0	4.0
	MLP	6.0	70.0	24.0
SP	REF	100.0	0.0	0.0
	REF+PM	64.0	36.0	0.0
	REF+PM+SHARE	62.0	38.0	0.0
	MLP	64.0	34.0	2.0
CC	REF	100.0	0.0	0.0
	REF+PM	37.5	62.5	0.0
	REF+PM+SHARE	55.8	44.2	0.0
	MLP	2.0	98.0	0.0
AN	REF	100.0	0.0	0.0
	REF+PM	42.0	10.0	48.0
	REF+PM+SHARE	94.0	2.0	4.0
	MLP	66.0	10.0	24.0

Table 10 Error analysis results



(a) word2vec (b) GloVe

Fig. 9 Distribution of the distance between the input word vector $\mathbf{v}_x$ and the target word vector $\mathbf{v}_t$ (Comparisons between the changed and unchanged attribute words)

transfers, the maximum failures using MLP were observed in Case2 and Case3 errors, while the proposed methods demonstrated significant failure in the Case1 category. This shows that the reflection-based transfer is more stable in MF and CC transfers than MLP.

8 Related Work

The embedded vectors obtained by SGNS (Mikolov et al. 2013a, 2013b) and GloVe (Pennington et al. 2014) have analogic relations. The theory of analogic relations in word embeddings has been widely discussed (Levy and Goldberg 2014b; Arora et al. 2016; Gittens et al. 2017; Ethayarajh et al. 2019; Allen and Hospedales 2019; Linzen 2016). Levy and Goldberg (2014b) explain that SGNS factorizes a shifted PMI matrix. Allen and Hospedales (2019) and Ethayarajh et al. (2019) argued that they proved the existence of such analogic relations without strong assumptions. In our work, we focused on the analogic relations in a word embedding space and propose a novel framework to obtain a transferred word vector with the target attribute.

Link prediction in knowledge graph embeddings can also be applied to word transfer tasks. In knowledge graph embeddings (Bordes et al. 2013; Trouillon et al. 2016; Nickel et al. 2011), given a set of triplets (head, tail, label), tail is predicted from head and label TransE (Bordes et al. 2013) is a knowledge graph embedding model. TransE embeds entity and relation into an embedding space using a score function. The score function models the analogy-like operation translating from the head entity to the tail entity according to the relation. Trouillon et al. (2016) proposed the

Attribute	Error type	Input (x)	Target (t)	Output (y)
MF	Case2	hens	roosters	oxen
	Case2	stepbrother	stepsister	stepmother
	Case2	emperor	empress	goddess
	Case3	mare	stallion	gelding
SP	Case2	killers	killer	murderer
	Case2	atoll	atolls	islands
	Case2	gulls	gull	heron
	Case2	windmill	windmills	paddlewheels
	Case2	saxophone	saxophones	trombones
	Case2	trails	trail	trailhead
	Case2	spectaculars	spectacular	extravaganza
	Case2	visa	visas	passports
	Case2	neckties	necktie	jacket
	Case2	wagons	wagon	tractor
	Case2	outlook	outlooks	forecasts
CC	Case2	Australia	Canberra	Sydney
	Case2	Canada	Ottawa	Montreal
	Case2	Jamaica	Kingston	Belmopan
	Case2	London	England	Britain
	Case2	Hungary	Budapest	Bucharest
AN	Case2	underbid	overbid	overcharged
	Case2	perfection	imperfection	imperfect
	Case2	rely	suspect, distrust	independent
	Case2	penalty	advantage, reward	acquittal
	Case2	sane	insane, crazy	irrational
	Case3	disinherit	leave, will, bequeath	disinheriting
	Case3	elder	junior	niece
	Case3	unimpressive	impressive	solid
	Case3	unhelpful	helpful	sensible
	Case3	starve	give, feed	encourage
	Case3	extraneous	intrinsic	necessary
	Case3	harmless	harmful	important

Table 11 Examples of Case2 and Case3 errors in the proposed method

knowledge graph embedding model based on complex values for link prediction. Their knowledge graph embedding model is better suited for modeling a variety of binary relations, including symmetric and asymmetric relations. Our task differs from link prediction in that when an

invariant word for attribute $\mathbf{z}$ is entered, the model returns the input x.

The style transfer task presented in previous studies (Jain et al. 2019; Logeswaran et al. 2018) resembles ours. In style transfer, the text style of the input sentences is changed. For instance, Jain et al. (2019) transferred the style from formal to informal sentences. Logeswaran et al. (2018) transferred sentences by controlling attributes such as mood and tense. These style transfer tasks use sentence pairs. Our word attribute transfer task uses word pairs. Style transfer changes sentence styles; however, our task changes the word attributes.

Soricut and Och (2015) studied morphological transformation based on character information. Our work aims for a more general attribute transfer, such as gender transfer and obtaining the antonym, and is not limited to morphological transformation.

9 Conclusion

This research aimed to transfer word binary attributes (e.g., gender) for applications such as data augmentation of a sentence.[6] We can transfer word attributes using the analogy of word vectors; however, this process requires explicit knowledge on whether the input word has the attribute or not. However, this knowledge cannot be developed for various words and attributes in practice. The proposed method uses reflection-based mappings to transfer attribute-variant words into their counterparts while ensuring that attribute-invariant words are unchanged, without using attribute knowledge in the inference time. The experimental results showed that the proposed method outperformed baseline methods in terms of transfer accuracy for attribute-variant words and stability for attribute-invariant words. We speculated that the reason why the proposed method achieved significantly high stability was that invariant words were distributed in the mirrors. We examined the distance between the input word vector and its mirror. The result showed that invariant words were distributed near the mirror and attribute words were distributed away from the mirror.

Acknowledgement

We would like to thank the anonymous reviewers for their insightful comments and suggestions. Part of this work was presented at the 58th Annual Meeting of the Association for Computational Linguistics (ACL2020) Student Research Workshop (SRW) (Ishibashi et al. 2020). This

[6] Our code and datasets are available at: `https://github.com/ahclab/reflection`

work was supported by JST CREST Grant Number JPMJCR1513.

References

Allen, C. and Hospedales, T. M. (2019). "Analogies Explained: Towards Understanding Word Embeddings." In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA*, pp. 223–231.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2016). "A Latent Variable Model Approach to PMI-based Word Embeddings." *Transactions of the Association for Computational Linguistics*, **4**, pp. 385–399.

Bordes, A., Usunier, N., García-Durán, A., Weston, J., and Yakhnenko, O. (2013). "Translating Embeddings for Modeling Multi-relational Data." In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States*, pp. 2787–2795.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). "A Large Annotated Corpus for Learning Natural Language Inference." In Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y. (Eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17–21, 2015*, pp. 632–642. The Association for Computational Linguistics.

Church, K. W. and Hanks, P. (1990). "Word Association Norms, Mutual Information, and Lexicography." *Computational Linguistics*, **16** (1), pp. 22–29.

Ethayarajh, K., Duvenaud, D., and Hirst, G. (2019). "Towards Understanding Linear Word Analogies." In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 3253–3262.

Gittens, A., Achlioptas, D., and Mahoney, M. W. (2017). "Skip-Gram $-Zipf$ + Uniform = Vector Additivity." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30–August 4, Volume 1: Long Papers*, pp. 69–76.

Gladkova, A., Drozd, A., and Matsuoka, S. (2016). "Analogy-based Detection of Morphological and Semantic Relations with Word Embeddings: What Works and What Doesn't." In *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, San Diego California, USA, June 12–17, 2016, pp. 8–15.

Glorot, X., Bordes, A., and Bengio, Y. (2011). "Deep Sparse Rectifier Neural Networks." In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11–13, 2011*, pp. 315–323.

Han, X., Cao, S., Lv, X., Lin, Y., Liu, Z., Sun, M., and Li, J. (2018). "OpenKE: An Open Toolkit for Knowledge Embedding." In Blanco, E. and Lu, W. (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31–November 4, 2018*, pp. 139–144. Association for Computational Linguistics.

Hinton, G. E., McClelland, J. L., Rumelhart, D. E., et al. (1984). *Distributed Representations*. Carnegie-Mellon University Pittsburgh, PA.

Ioffe, S. and Szegedy, C. (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015*, pp. 448–456.

Ishibashi, Y., Sudoh, K., Yoshino, K., and Nakamura, S. (2020). "Reflection-based Word Attribute Transfer." In Rijhwani, S., Liu, J., Wang, Y., and Dror, R. (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2020, Online, July 5–10, 2020*, pp. 51–58. Association for Computational Linguistics.

Jain, P., Mishra, A., Azad, A. P., and Sankaranarayanan, K. (2019). "Unsupervised Controllable Text Formalization." In *The 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, The 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27–February 1, 2019*, pp. 6554–6561.

Kaneko, M. and Bollegala, D. (2019). "Gender-preserving Debiasing for Pre-trained Word Embeddings." In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers*, pp. 1641–1650.

Kang, D., Khot, T., Sabharwal, A., and Hovy, E. H. (2018). "AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples." In Gurevych, I. and Miyao, Y. (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers*, pp. 2418–2428. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2015). "Adam: A Method for Stochastic Optimization." In *3rd*

International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings.

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). "Phrase-Based & Neural Unsupervised Machine Translation." In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31–November 4, 2018*, pp. 5039–5049. Association for Computational Linguistics.

Levy, O. and Goldberg, Y. (2014a). "Linguistic Regularities in Sparse and Explicit Word Representations." In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26–27, 2014*, pp. 171–180.

Levy, O. and Goldberg, Y. (2014b). "Neural Word Embedding as Implicit Matrix Factorization." In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13 2014, Montreal, Quebec, Canada*, pp. 2177–2185.

Linzen, T. (2016). "Issues in Evaluating Semantic Spaces Using Word Analogies." In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP, RepEval@ACL 2016, Berlin, Germany, August 2016*, pp. 13–18.

Logeswaran, L., Lee, H., and Bengio, S. (2018). "Content Preserving Text Generation with Attribute Controls." In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3–8 December 2018, Montréal, Canada*, pp. 5108–5118.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). "Efficient Estimation of Word Representations in Vector Space." In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). "Distributed Representations of Words and Phrases and their Compositionality." In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States.*, pp. 3111–3119.

Mikolov, T., tau Yih, W., and Zweig, G. (2013c). "Linguistic Regularities in Continuous Space Word Representations." In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9–14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pp. 746–751.

Miller, G. A. (1995). "WordNet: A Lexical Database for English." *Communications of the ACM*,

38 (11), pp. 39–41.

Nguyen, K. A., im Walde, S. S., and Vu, N. T. (2017). "Distinguishing Antonyms and Synonyms in a Pattern-based Neural Network." In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3–7, 2017, Volume 1: Long Papers*, pp. 76–85.

Nickel, M. and Kiela, D. (2017). "Poincaré Embeddings for Learning Hierarchical Representations." In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA*, pp. 6338–6347.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). "A Three-Way Model for Collective Learning on Multi-Relational Data." In Getoor, L. and Scheffer, T. (Eds.), *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28–July 2, 2011*, pp. 809–816. Omnipress.

Pennington, J., Socher, R., and Manning, C. D. (2014). "Glove: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543.

Soricut, R. and Och, F. J. (2015). "Unsupervised Morphology Induction Using Word Embeddings." In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31–June 5, 2015*, pp. 1627–1637.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal fo Machine Learning Research*, **15** (1), pp. 1929–1958.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). "Complex Embeddings for Simple Link Prediction." In Balcan, M.-F. and Weinberger, K. Q. (Eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016*, Vol. 48 of *JMLR Workshop and Conference Proceedings*, pp. 2071–2080. JMLR.org.

Vilnis, L. and McCallum, A. (2015). "Word Representations via Gaussian Embedding." In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Yang, Z., Chen, W., Wang, F., and Xu, B. (2018). "Unsupervised Neural Machine Translation with Weight Sharing." In Gurevych, I. and Miyao, Y. (Eds.), *Proceedings of the 56th Annual*

Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers, pp. 46–55. Association for Computational Linguistics.

Zhao, J., Zhou, Y., Li, Z., Wang, W., and Chang, K.-W. (2018). "Learning Gender-Neutral Word Embeddings." In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31–November 4, 2018*, pp. 4847–4853.

Yoichi Ishibashi: He is a Ph.D. student at the Nara Institute of Science and Technology. He received his bachelor's degree in computer science from Kyoto Sangyo University in 2018 and his master's degree in engineering from Nara Institute of Science and Technology in 2020.

Katsuhito Sudoh: He is an associate professor at the Nara Institute of Science and Technology. He received his bachelor's degree in engineering in 2000 and his master's and Ph.D. degrees in informatics in 2002 and 2015, respectively, from Kyoto University. He worked with the NTT Communication Science Laboratories from 2002 to 2017. He currently works on machine translation and natural language processing. He is a member of ACL, ANLP, IPSJ, and ASJ.

Koichiro Yoshino: He received his B.A. degree in 2009 from Keio University, M.S. degree in informatics in 2011, and Ph.D. degree in informatics in 2014 from Kyoto University. From 2014 to 2015, he was a research fellow (PD) of Japan Society for Promotion of Science. From 2015, he has worked as an assistant professor in Nara Institute of Science and Technology (NAIST). He is currently working on areas of spoken and natural language processing, especially on spoken dialogue systems. Dr. Koichiro Yoshino received the JSAI SIG Research Award in 2013 and ANLP outstanding paper award in 2018. He is a member of IEEE, SIGDIAL, ACL, IPSJ, RSJ, and ANLP.

Satoshi Nakamura: He is a professor at the Nara Institute of Science and Technology, Team Leader of the Tourism Information Analytics Team, AIP Center, RIKEN, and Honorar professor at the Karlsruhe Institute of Technology, Germany. He received his B.S. from the Kyoto Institute of Technology in 1981 and a Ph.D. from the Kyoto University in 1992. He was the Director of ATR Spoken Language Communication Research Laboratories in the period 2000–2008

and Vice President of ATR in the period 2007–2008. He was the Director General of Keihanna Research Laboratories and Executive Director of Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology in 2009–2010. He is currently the director of the Augmented Human Communication laboratory and a full professor at the Data Science Center and Information Science Division, Graduate School of Science and Technology, Nara Institute of Science and Technology. He was an Elected Board Member of the International Speech Communication Association, ISCA, in the period June 2011-2019, and IEEE Signal Processing Magazine Editorial Board member in 2012-2015, IEEE SPS Speech and Language Technical Committee Member in 2013-2015. He is an ATR Fellow, IPSJ Fellow, IEEE Fellow, and ISCA Fellow.

(Received July 30, 2020)
(Revised November 20, 2020)
(Accepted December 26, 2020)