Why does PairDiff *work*? – A Mathematical Analysis of Bilinear Relational Compositional Operators for Analogy Detection

Huda Hakami
The University of Liverpool
Liverpool, UK
h.a.hakami@liv.ac.uk

Kohei Hayashi
Preferred Networks
Tokyo, Japan
hayashi.kohei@gmail.com

Danushka Bollegala
The University of Liverpool
Liverpool, UK
danushka.bollegala@liv.ac.uk

Abstract

Representing the semantic relations that exist between two given words (or entities) is an important first step in a wide-range of NLP applications such as analogical reasoning, knowledge base completion and relational information retrieval. A simple, yet surprisingly accurate method for representing a relation between two words is to compute the vector offset (PairDiff) between their corresponding word embeddings. Despite the empirical success, it remains unclear as to whether PairDiff is the best operator for obtaining a relational representation from word embeddings. We conduct a theoretical analysis of generalised bilinear operators that can be used to measure the ℓ_2 relational distance between two word-pairs. We show that, if the word embeddings are standardised and uncorrelated, such an operator will be independent of bilinear terms, and can be simplified to a linear form, where PairDiff is a special case. For numerous word embedding types, we empirically verify the uncorrelation assumption, demonstrating the general applicability of our theoretical result. Moreover, we experimentally discover PairDiff from the bilinear relational compositional operator on several benchmark analogy datasets.

1 Introduction

Different types of semantic relations exist between words such as HYPERNYMY between *ostrich* and *bird*, or ANTONYMY between *hot* and *cold*. If we consider entities[1], we can observe even a richer diversity of relations such as FOUNDER-OF between *Bill Gates* and *Microsoft*, or CAPITAL-OF between *Tokyo* and *Japan*. Identifying the relations between words and entities is important for various Natural Language Processing (NLP) tasks such as automatic knowledge base completion (Socher et al., 2013), analogical reasoning (Turney and Littman, 2005; Bollegala et al., 2009) and relational information retrieval (Duc et al., 2010). For example, to solve a word analogy problem of the form "*a* is to *b* as *c* is to *?*", the relationship between the two words in the pair (a, b) must be correctly identified in order to find candidates d that have similar relations with c. For example, given the query "*Bill Gates* is to *Microsoft* as *Steve Jobs* is to *?*", a relational search engine must retrieve *Apple Inc.* because the FOUNDER-OF relation exists between the first and the second entity pairs.

Two main approaches for creating relation embeddings can be identified in the literature. In the first approach, from given corpora or knowledge bases, word and relation embeddings are *jointly* learnt such that some objective is optimised (Guo et al., 2016; Yang et al., 2015; Nickel et al., 2016; Bordes et al., 2013; Rocktäschel et al., 2016; Minervini et al., 2017; Trouillon et al., 2016). In this approach, word and relation embeddings are considered to be *independent* parameters that must be learnt by the embedding method. For example, TransE (Bordes et al., 2013) learns the word and relation embeddings such that we can accurately predict relations (links) in a given knowledge base using the learnt word and relation embeddings. Because relations are learnt independently from the words, we refer to methods that are based on this approach as *independent* relational embedding methods.

[1] We interchangeably use the terms *word* and *entity* to represent both unigrams as well as a multi-word expressions including named entities.

Proceedings of the 27th International Conference on Computational Linguistics, pages 2493–2504
Santa Fe, New Mexico, USA, August 20-26, 2018.

A second approach for creating relational embeddings is to apply some operator on two word embeddings to *compose* the embedding for the relation that exits between those two words, if any. In contrast to the first approach, we do not have to learn relational embeddings and hence this can be considered as an unsupervised setting, where the compositional operator is predefined. A popular operator for composing a relational embedding from two word embeddings is PairDiff, which is the vector difference (offset) of the word embeddings (Mikolov et al., 2013b; Levy and Goldberg, 2014; Vylomova et al., 2016; Bollegala et al., 2015b; Blacoe and Lapata, 2012). Specifically, given two words a and b represented by their word embeddings respectively $\boldsymbol{a}$ and $\boldsymbol{b}$, the relation between a and b is given by $\boldsymbol{a} - \boldsymbol{b}$ under the PairDiff operator. Mikolov et al. (2013b) showed that PairDiff can accurately solve analogy equations such as $\overrightarrow{\mathbf{king}} - \overrightarrow{\mathbf{man}} + \overrightarrow{\mathbf{woman}} = \overrightarrow{\mathbf{queen}}$, where we have used the top arrows to denote the embeddings of the corresponding words. Bollegala et al. (2015a) showed that PairDiff can be used as a proxy for learning better word embeddings and Vylomova et al. (2016) conducted an extensive empirical comparison of PairDiff using a dataset containing 16 different relation types. Besides PairDiff, concatenation (Hakami and Bollegala, 2017; Yin and Schütze, 2016), circular correlation and convolution (Nickel et al., 2016) have been used in prior work for representing the relations between words. Because the relation embedding is composed using word embeddings instead of learning as a separate parameter, we refer to methods that are based on this approach as *compositional* relational embedding methods. Note that in this approach it is implicitly assumed that there exist only a single relation between two words.

In this paper, we focus on the operators that are used in compositional relational embedding methods. If we assume that the words and relations are represented by vectors embedded in some common space, then the operator we are seeking must be able to produce a vector representing the relation between two words, given their word embeddings as the only input. Although there have been different proposals for computing relational embeddings from word embeddings, it remains unclear as to what is the best operator for this task. The space of operators that can be used to compose relational embeddings is open and vast. A space of particular interest from a computational point-of-view is the bilinear operators that can be parametrised using tensors and matrices. Specifically, we consider operators that consider pairwise interactions between two word embeddings (second-order terms) and contributions from individual word embeddings towards their relational embedding (first-order terms). The optimality of a relational compositional operator can be evaluated, for example, using the expected relational distance/similarity such as ℓ_2 between analogous (positive) vs. nonanalogous (negative) word-pairs.

If we assume that word embeddings are standardised, uncorrelated and word-pairs are i.i.d, then we prove in §3 that bilinear relational compositional operators are independent of bilinear pairwise interactions between the two input word embeddings. Moreover, under regularised settings (§3.1), the bilinear operator further simplifies to a linear combination of the input embeddings, and the expected loss over positive and negative instances becomes zero. In §4.1, we empirically validate the uncorrelation assumption for different pre-trained word embeddings such as the Continuous Bag-of-Words Model (CBOW) (Mikolov et al., 2013a), Skip-Gram with negative sampling (SG) (Mikolov et al., 2013a), Global Vectors (GloVe) (Pennington et al., 2014), word embeddings created using Latent Semantic Analysis (LSA) (Deerwester et al., 1990), Sparse Coding (HSC) (Faruqui et al., 2015; Yogatama et al., 2015), and Latent Dirichlet Allocation (LDA) (Blei et al., 2003). This empirical evidence implies that our theoretical analysis is applicable to relational representations composed from a wide-range of word embedding learning methods. Moreover, our experimental results show that a bilinear operator reaches its optimal performance in two different word-analogy benchmark datasets, when it satisfies the requirements of the PairDiff operator. We hope that our theoretical analysis will expand the understanding of relational embedding methods, and inspire future research on accurate relational embedding methods using word embeddings as the input.

2 Related Work

As already mentioned in §1, methods for representing a relation between two words can be broadly categorised into two groups depending on whether the relational embeddings are learnt *independently* of the word embeddings, or they are *composed* from the word embeddings, in which case the relational

embeddings fully depend on the input word embeddings. Next, we briefly overview the different methods that fall under each category. For a detailed survey of relation embedding methods see Nickel et al. (2015).

Given a knowledge base where an entity h is linked to an entity t by a relation r, the TransE model (Bordes et al., 2013) scores the tuple (h, t, r) by the ℓ_1 or ℓ_2 norm of the vector $(\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t})$. Nickel et al. (2011) proposed RESCAL, which uses $\boldsymbol{h}^\top \mathbf{M}_r \boldsymbol{t}$ as the scoring function, where $\mathbf{M}_r$ is a matrix embedding of the relation r. Similar to RESCAL, Neural Tensor Network (Socher et al., 2013) also models a relation by a matrix. However, compared to vector embeddings of relations, matrix embeddings increase the number of parameters to be estimated, resulting in an increase in computational time/space and likely to overfit. To overcome these limitations, DistMult (Yang et al., 2015) models relations by vectors and use elementwise multilinear dot product $\boldsymbol{r} \odot \boldsymbol{h} \odot \boldsymbol{t}$. Unfortunately, DistMult cannot capture directionality of a relation. Complex Embeddings (Trouillon et al., 2016) overcome this limitation of DistMult by using complex embeddings and defining the score to be the real part of $\boldsymbol{r} \odot \boldsymbol{h} \odot \bar{\boldsymbol{t}}$, where $\bar{\boldsymbol{t}}$ denotes the complex conjugate of $\boldsymbol{t}$.

The observation made by Mikolov et al. (2013b) that the relation between two words can be represented by the difference between their word embeddings sparked a renewed interest in methods that compose relational embeddings using word embeddings. Word analogy datasets such as Google dataset (Mikolov et al., 2013b), SemEval 2012 Task2 dataset (Jurgens et al., 2012), BATS (Drozd et al., 2016) etc. have established as benchmarks for evaluating word embedding learning methods.

Different methods have been proposed to measure the similarity between the relations that exist between two given word pairs such as CosMult, CosAdd and PairDiff (Levy and Goldberg, 2014; Bollegala et al., 2015a). Vylomova et al. (2016) studied as to what extent the vectors generated using simple PairDiff encode different relation types. Under supervised classification settings, they conclude that PairDiff can cover a wide range of semantic relation types. Holographic embeddings proposed by Nickel et al. (2016) use circular convolution to mix the embeddings of two words to create an embedding for the relation that exist between those words. It can be showed that circular correlation is indeed an elementwise product in the Fourier space and is mathematically equivalent to complex embeddings (Hayashi and Shinbo, 2017).

Although PairDiff operator has been widely used in prior work for computing relation embeddings from word embeddings, to the best of our knowledge, no theoretical analysis has been conducted so far explaining why and under what conditions PairDiff is optimal, which is the focus of this paper.

3 Bilinear Relation Representations

Let us consider the problem of representing the semantic relation $\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t})$ between two given words h and t. We assume that h and t are already represented in some d-dimensional space respectively by their word embeddings $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^d$. The relation between two words can be represented using different linear algebraic structures. Two popular alternatives are vectors (Nickel et al., 2016; Bordes et al., 2013; Minervini et al., 2017; Trouillon et al., 2016) and matrices (Socher et al., 2013; Bollegala et al., 2015b). Vector representations are preferred over matrix representations because of the smaller number of parameters to be learnt (Nickel et al., 2015).

Let us assume that the relation r is represented by a vector $\boldsymbol{r} \in \mathbb{R}^\delta$ in some δ-dimensional space. Therefore, we can write $\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t})$ as a function that takes two vectors (corresponding to the embeddings of the two words) as the input and returns a single vector (representing the relation between the two words) as given in (1).

$$\boldsymbol{r}: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^\delta \tag{1}$$

Having both words and relations represented in the same $\delta = d$ dimensional space is useful for performing linear algebraic operations using those representations in that space. For example, in TransE (Bordes et al., 2013), the strength of a relation r that exists between two words h and t is computed as the $\ell_{1,2}$ norm of the vector $(\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t})$ using the word and relation embeddings. Such direct comparisons between word and relation embeddings would not be possible if words and relations were not embedded in the

same vector space. If $\delta < d$, we can first project word embeddings to a lower δ-dimensional space using some dimensionality reduction method such as SVD, whereas if $\delta > d$ we can learn higher δ-dimensional overcomplete word representations (Faruqui et al., 2015) from the original d-dimensional word embeddings. Therefore, we will limit our theoretical analysis to the $\delta = d$ case for ease of description.

Different functions can be used as $\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t})$ that satisfy the domain and range requirements specified by (1). If we limit ourselves to bilinear functions, the most general functional form is given by (2).

$$\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t}) = \boldsymbol{h}^\top \underline{\mathbf{A}} \boldsymbol{t} + \mathbf{P}\boldsymbol{h} + \mathbf{Q}\boldsymbol{t} \tag{2}$$

Here, $\underline{\mathbf{A}} \in \mathbb{R}^{d \times d \times d}$ is a 3-way tensor in which each slice is a $d \times d$ real matrix. Let us denote the k-th slice of $\underline{\mathbf{A}}$ by $\mathbf{A}^{(k)}$ and its (i, j) element by $A_{ij}^{(k)}$. The first term in (2) corresponds to the pairwise interactions between $\boldsymbol{h}$ and $\boldsymbol{t}$. $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{d \times d}$ are the nonsingular[2] projection matrices involving first-order contributions respectively of $\boldsymbol{h}$ and $\boldsymbol{t}$ towards $\boldsymbol{r}$.

Let us consider the problem of learning the simplest bilinear functional form according to (2) from a given dataset of analogous word-pairs $\mathcal{D}_+ = \{((h, t), (h', t'))\}$. Specifically, we would like to learn the parameters $\underline{\mathbf{A}}$, $\mathbf{P}$ and $\mathbf{Q}$ such that some distance (loss) between analogous word-pairs is minimised. As a concrete example of a distance function, let us consider the popularly used Euclidean distance[3] (ℓ_2 loss) for two word pairs given by (3).

$$J((h, t), (h', t')) = \left|\left|\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t}) - \boldsymbol{r}(\boldsymbol{h'}, \boldsymbol{t'})\right|\right|_2^2 \tag{3}$$

If we were provided only analogous word-pairs (i.e. positive examples), then this task could be trivially achieved by setting all parameters to zero. However, such a trivial solution would not generalise to unseen test data. Therefore, in addition to $\mathcal{D}_+$ we would require a set of non-analogous word-pairs $\mathcal{D}_-$ as negative examples. Such negative examples are often generated in prior work by randomly corrupting positive relational tuples (Nickel et al., 2016; Bordes et al., 2013; Trouillon et al., 2016) or by training an adversarial generator (Minervini et al., 2017).

The total loss J over both positive and negative training data can be written as follows:

$$J = \sum_{((h,t),(h',t')) \in \mathcal{D}_+} \left|\left|\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t}) - \boldsymbol{r}(\boldsymbol{h'}, \boldsymbol{t'})\right|\right|_2^2 - \sum_{((h,t),(h',t')) \in \mathcal{D}_-} \left|\left|\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t}) - \boldsymbol{r}(\boldsymbol{h'}, \boldsymbol{t'})\right|\right|_2^2 \tag{4}$$

Assuming that the training word-pairs are randomly sampled from $\mathcal{D}_+$ and $\mathcal{D}_-$ according to two distributions respectively p_+ and p_-, we can compute the total expected loss, $\mathbb{E}_p[J]$, as follows:

$$\mathbb{E}_p[J] = \mathbb{E}_{p_+}\left[\left|\left|\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t}) - \boldsymbol{r}(\boldsymbol{h'}, \boldsymbol{t'})\right|\right|_2^2\right] - \mathbb{E}_{p_-}\left[\left|\left|\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t}) - \boldsymbol{r}(\boldsymbol{h'}, \boldsymbol{t'})\right|\right|_2^2\right] \tag{5}$$

We make the following assumptions to further analyse the properties of relational embeddings.

Uncorrelation: The correlation between any two distinct dimensions of a word embedding is zero. One might think that the uncorrelation of word embedding dimensions to be a strong assumption, but we later show its validity empirically in §4.1 for a wide range of word embeddings.

Standardisation: Word embeddings are standardised to zero mean and unit variance. This is a linear transformation in the word embedding space and does not affect the topology of the embedding space. In particular, translating word embeddings such that they have a zero mean has shown to improve performance in similarity tasks Mu et al. (2018).

Relational Independence Word pairs in the training data are assumed to be i.i.d. For example, whether a particular semantic relation r exists between h and t, is assumed to be independent of any other relation r' that exists between h' and t' in a different pair. For example, (*ostrich*, is-a-large, *bird*) is independent from (*Trump*, president-of, *USA*).

[2] If the projection matrix is nonsingular, then the inverse projection exists, which preserves the dimensionality of the embedding space.

[3] For ℓ_2 normalised vectors, their Euclidean distance is a monotonously decreasing function of their cosine similarity.

For relation representations given by (2), 1 holds:

Theorem 1. *Consider the bilinear relational embedding defined by 2 computed using uncorrelated word embeddings. If the word embeddings are standardised, then the expected loss given by 5 over a relationally independent set of word pairs is independent of* $\underline{\mathbf{A}}$.

Proof. Let us consider the bilinear term in (2), because i and $j(\neq i)$ dimensions of word embeddings are uncorrelated by the assumption (i.e. $\text{corr}(u_i, u_j) = 0$), from the definition of correlation we have,

$$\text{corr}(u_i, u_j) = \mathbb{E}[u_i u_j] - \mathbb{E}[u_i]\mathbb{E}[u_j] = 0 \tag{6}$$

$$\mathbb{E}[u_i u_j] = \mathbb{E}[u_i]\mathbb{E}[u_j]. \tag{7}$$

Moreover, from the standardisation assumption we have, $\mathbb{E}[u_i] = 0, \ \ \forall_{i=1\ldots n}$. From (7) it follows that:

$$\mathbb{E}[u_i u_j] = 0 \tag{8}$$

for $i \neq j$ dimensions.

We will next show that (5) is independent of $\underline{\mathbf{A}}$. For this purpose, let us consider the $\mathbb{E}_{p_+}$ term first and write the k-th dimension of $\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t})$ using $\mathbf{A}^{(k)}$, $\mathbf{P}$ and $\mathbf{Q}$ as follows:

$$\sum_{i,j} \left(A^{(k)}_{ij} h_i t_j \right) + \sum_n P_{kn} h_n + \sum_n Q_{kn} t_n \tag{9}$$

Plugging (9) in (5) and computing the loss over all positive training instances we get,

$$\mathbb{E}_{p_+}[\sum_k (\sum_{i,j} \left(A^{(k)}_{ij} (h_i t_j - h'_i t'_j) \right) + \sum_n P_{kn}(h_n - h'_n) + \sum_n Q_{kn}(t_n - t'_n))^2] \tag{10}$$

Terms that involve only elements in $\mathbf{A}^{(k)}$ take the form:

$$\begin{aligned} &\sum_{i,j} \sum_{l,m} \mathbb{E}_{p_+} \left[A^{(k)}_{ij} A^{(k)}_{lm} (h_i t_j - h'_i t'_j)(h_l t_m - h'_l t'_m) \right] \\ &= \sum_{i,j} \sum_{l,m} A^{(k)}_{ij} A^{(k)}_{lm} (\mathbb{E}_{p_+}[h_i t_j h_l t_m] - \mathbb{E}_{p_+}[h_i t_j h'_l t'_m] - \mathbb{E}_{p_+}[h'_i t'_j h_l t_m] + \mathbb{E}_{p_+}[h'_i t'_j h'_l t'_m]) \end{aligned} \tag{11}$$

Lets first analyse the cases where $i \neq j$ and $l \neq m$. Because of the relational independence assumption, the second and the third expectations in (11) can be written as follows: $\mathbb{E}_{p_+}[h_i t_j]\mathbb{E}_{p_+}[h'_l t'_m]$ and $\mathbb{E}_{p_+}[h'_i t'_j]\mathbb{E}_{p_+}[h_l t_m]$, respectively. Each of the these expectations contains the product of different dimensionalities in two different words. Expected correlation of different dimensions in the same word is zero from (8). Therefore, such cross-correlations are likely to be small for different word pairs, which are nonsynomyous. On the other hand, first and fourth expectations in (11) involve the same pair of words. For example, we could write the first expectation as follows:

$$\mathbb{E}_{p_+}[h_i t_j h_l t_m] = \mathbb{E}_{p_+}[(h_i h_l)(t_j t_m) = \mathbb{E}_{p_+}[H_{il} T_{jm}]$$

where $H = h_i h_l$ and $T_{jm} = t_j t_m$. If we think of $\boldsymbol{H}$ and $\boldsymbol{T}$ as d^2-dimensional word embeddings, $\mathbb{E}_{p_+}[H_{il} T_{jm}]$ represents the expectation over two distinct dimensions of $\boldsymbol{H}$ and $\boldsymbol{T}$ for $il \neq jm$. Therefore, from the same logic as above, this expectation is approximately zero.[4]

For $i = j = l = m$ case we have,

$$A^{(k)2}_{ij} (\mathbb{E}_{p_+}[h_i^2 t_i^2] - 2\mathbb{E}_{p_+}[h_i t_i h'_i t'_i] + \mathbb{E}_{p_+}[h'^2_i t'^2_i]) \tag{12}$$

[4]Note that il could be equal to jm even when $i \neq j$ and $l \neq m$. However, such cases will be a rare minority. Nevertheless, it is an approximation and not an exact zero.

Because we are considering word-pairs (h, t) for which a relation r is known to hold, from the definition of the correlation between the same dimension in different words we have:

$$\text{corr}(h_i^2, t_i^2) = \mathbb{E}[h_i^2 t_i^2] - \mathbb{E}[h_i^2]\mathbb{E}[t_i^2] = 1$$
$$\mathbb{E}[h_i^2 t_i^2] = \mathbb{E}[h_i^2]\mathbb{E}[t_i^2] + 1$$

Because $\mathbb{E}[h_i^2] = \mathbb{E}[t_i^2] = 1$ from the standardisation, we get:

$$\mathbb{E}[h_i^2 t_i^2] = 2 \tag{13}$$

Lets analyse the second term in (12). From the relational independence and because the word embeddings are assumed to be standardised to unit variance, we obtain the follows:

$$2\mathbb{E}_{p_+}[h_i t_i h_i' t_i'] = 2\mathbb{E}_{p_+}[h_i t_i]\mathbb{E}_{p_+}[h_i' t_i'] = 2. \tag{14}$$

According to (13) and (14), (12) evaluates to $2{A_{ij}^{(k)}}^2$. We will then get the same term from the negative expectations and they would cancel out as $2{A_{ij}^{(k)}}^2$ is independent of the training dataset.

Next, lets consider the $A_{ij}^{(k)} P_{kn}$ terms in the expansion of (10) given by,

$$2\sum_{i,j}\sum_{n} A_{ij}^{(k)} P_{kn}(h_i t_j - h_i' t_j')(h_n - h_n'). \tag{15}$$

Taking the expectation of (15) w.r.t. p_+ we get,

$$2\sum_{i,j}\sum_{n} A_{ij}^{(k)} P_{kn}(\mathbb{E}_{p_+}[h_i t_j h_n] - \mathbb{E}_{p_+}[h_i t_j h_n'] - \mathbb{E}_{p_+}[h_i' t_j' h_n] + \mathbb{E}_{p_+}[h_i' t_j' h_n']). \tag{16}$$

Likewise, from the uncorrelation assumption and following the same logic as above it follows that all the expectations in (16) are approximately zero. A similar argument can be used to show that terms that involve $A_{ij}^{(k)} Q_{kn}$ disappear from (10). Therefore, $\underline{\mathbf{A}}$ does not play any part in the expected loss over positive examples. Similarly, we can show that $\underline{\mathbf{A}}$ is independent of the expected loss over negative examples. Therefore, from (5) we see that the expected loss over the entire training dataset is independent of $\underline{\mathbf{A}}$. □

3.1 Regularised ℓ_2 loss

As a special case, if we attempt to minimise the expected loss under some regularisation on $\underline{\mathbf{A}}$ such as the Frobenius norm regularisation, then this can be achieved by sending $\underline{\mathbf{A}}$ to zero tensor because according to 1 2 is independent from $\underline{\mathbf{A}}$.

With $\underline{\mathbf{A}} = \underline{\mathbf{0}}$, the relation between h and t can be simplified to:

$$\boldsymbol{r}(\boldsymbol{h}, \boldsymbol{t}) = \mathbf{P}\boldsymbol{h} + \mathbf{Q}\boldsymbol{t} \tag{17}$$

Then the expected loss over the positive instances is given by (18).

$$\begin{aligned}
&\mathbb{E}_{p_+}[\left|\left|\mathbf{P}(\boldsymbol{h} - \boldsymbol{h'}) + \mathbf{Q}(\boldsymbol{t} - \boldsymbol{t'})\right|\right|_2^2] \\
&= \mathbb{E}_{p_+}[(\boldsymbol{h} - \boldsymbol{h'})^\top \mathbf{P}^\top \mathbf{P}(\boldsymbol{h} - \boldsymbol{h'})] + \mathbb{E}_{p_+}[(\boldsymbol{h} - \boldsymbol{h'})^\top \mathbf{P}^\top \mathbf{Q}(\boldsymbol{t} - \boldsymbol{t'})] + \\
&\mathbb{E}_{p_+}[(\boldsymbol{t} - \boldsymbol{t'})^\top \mathbf{Q}^\top \mathbf{P}(\boldsymbol{h} - \boldsymbol{h'})] + \mathbb{E}_{p_+}[(\boldsymbol{t} - \boldsymbol{t'})^\top \mathbf{Q}^\top \mathbf{Q}(\boldsymbol{t} - \boldsymbol{t'})]
\end{aligned} \tag{18}$$

The second expectation term in the right hand side of (18) can be computed as follows:

$$\begin{aligned}
&\mathbb{E}_{p_+}[(\boldsymbol{h} - \boldsymbol{h'})^\top \mathbf{P}^\top \mathbf{Q}(\boldsymbol{t} - \boldsymbol{t'})] \\
&= \sum_{i,j} (\mathbf{P}^\top \mathbf{Q})_{ij} \mathbb{E}_{p_+}[(h_i - h_i')(t_j - t_j')] \\
&= \sum_{i,j} (\mathbf{P}^\top \mathbf{Q})_{ij} \left(\mathbb{E}_{p_+}[h_i t_j] - \mathbb{E}_{p_+}[h_i t_j'] - \mathbb{E}_{p_+}[h_i' t_j] + \mathbb{E}_{p_+}[h_i' t_j']\right)
\end{aligned} \tag{19}$$

When $i \neq j$, each of the four expectations in the RHS of (21) are zero from the uncorrelation assumption. When $i = j$, each term will be equal to one from the standardisation assumption (unit variance) and cancel each other out. A similar argument can be used to show that the third expectation term in the RHS of (18) vanishes.

Now lets consider the first expectation term in the RHS of (18), which can be computed as follows:

$$\begin{aligned}
&\mathbb{E}_{p_+}[(\boldsymbol{h}-\boldsymbol{h'})^\top \mathbf{P}^\top \mathbf{P}(\boldsymbol{h}-\boldsymbol{h'})] \\
&= \sum_{i,j} (\mathbf{P}^\top \mathbf{P})_{ij} \mathbb{E}_{p_+}[(h_i - h'_i)(h_j - h'_j)] \\
&= \sum_{i,j} (\mathbf{P}^\top \mathbf{P})_{ij} (\mathbb{E}_{p_+}[h_i h_j] - \mathbb{E}_{p_+}[h_i h'_j] - \mathbb{E}_{p_+}[h'_i h_j] + \mathbb{E}_{p_+}[h'_i h'_j])
\end{aligned} \tag{20}$$

When $i \neq j$, it follows from the uncorrelation assumption that each of the four expectation terms in the RHS of (20) will be zero. For $i = j$ case we have,

$$\begin{aligned}
&\sum_{i,j} (\mathbf{P}^\top \mathbf{P})_{ii} \left(\mathbb{E}_{p_+}[h_i^2] - 2\mathbb{E}_{p_+}[h_i h'_i] + \mathbb{E}_{p_+}[h_i'^2] \right) \\
&= 2 \sum_{i,j} (\mathbf{P}^\top \mathbf{P})_{ii}
\end{aligned} \tag{21}$$

Note that from the relational independence between h and h' we have $\mathbb{E}_{p_+}[h_i h'_i] = \mathbb{E}_{p_+}[h_i]\mathbb{E}_{p_+}[h'_i]$. From the standardisation (zero mean) assumption this term is zero. On the other hand $\mathbb{E}_{p_+}[h_i^2] = \mathbb{E}_{p_+}[h_i'^2] = 1$ from the standardisation (unit variance) assumption, which gives the result in (21).

Similarly, the fourth expectation term in the RHS of (18) evaluates to $2\sum_{i,j} (\mathbf{Q}^\top \mathbf{Q})_{ii}$, which shows that (18) evaluates to $2\sum_{i,j} \left((\mathbf{P}^\top \mathbf{P})_{ii} + (\mathbf{Q}^\top \mathbf{Q})_{ii} \right)$. Note that this is independent of the positive instances and will be equal to the expected loss over negative instances, which gives $\mathbb{E}_p[J] = 0$ for the relational embedding given by (17).

It is interesting to note that PairDiff is a special case of (17), where $\mathbf{P} = \mathbf{I}$ and $\mathbf{Q} = -\mathbf{I}$. In the general case where word embeddings are nonstandardised to unit variance, we can set $\mathbf{P}$ to be the diagonal matrix where $\mathbf{P}_{ii} = 1/\sigma_i$, where σ_i is the variance of the i-th dimension of the word embedding space, to enforce standardisation. Considering that $\mathbf{P}, \mathbf{Q}$ are parameters of the relational embedding, this is analogous to *batch normalisation* (Ioffe and Szegedy, 2015), where the appropriate parameters for the normalisation are learnt during training.

4 Experimental Results

4.1 Cross-dimensional Correlations

A key assumption in our theoretical analysis is the uncorrelations between different dimensions in word embeddings. Here, we empirically verify the uncorrelation assumption for different input word embeddings. For this purpose, we create SG, CBOW and GloVe embeddings from the ukWaC corpus[5]. We use a context window of 5 tokens and select words that occur at least 6 times in the corpus. We use the publicly available implementations for those methods by the original authors and set the parameters to the recommended values in (Levy et al., 2015) to create 50-dimensional word embeddings. As a representative of counting-based word embeddings, we create a word co-occurrence matrix weighted by the positive pointwise mutual information (PPMI) and apply singular value decomposition (SVD) to obtain 50-dimensional embeddings, which we refer to as the Latent Semantic Analysis (LSA) embeddings.

We use Latent Dirichlet Allocation (LDA) (Blei et al., 2003) to create a topic model, and represent each word by its distribution over the set of topics. Ideally, each topic will capture some semantic category and the topic distribution provides a semantic representation for a word. We use gensim[6] to extract 50 topics from a 2017 January dump of English Wikipedia. In contrast to the above-mentioned word embeddings,

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

[6] https://radimrehurek.com/gensim/wiki.html

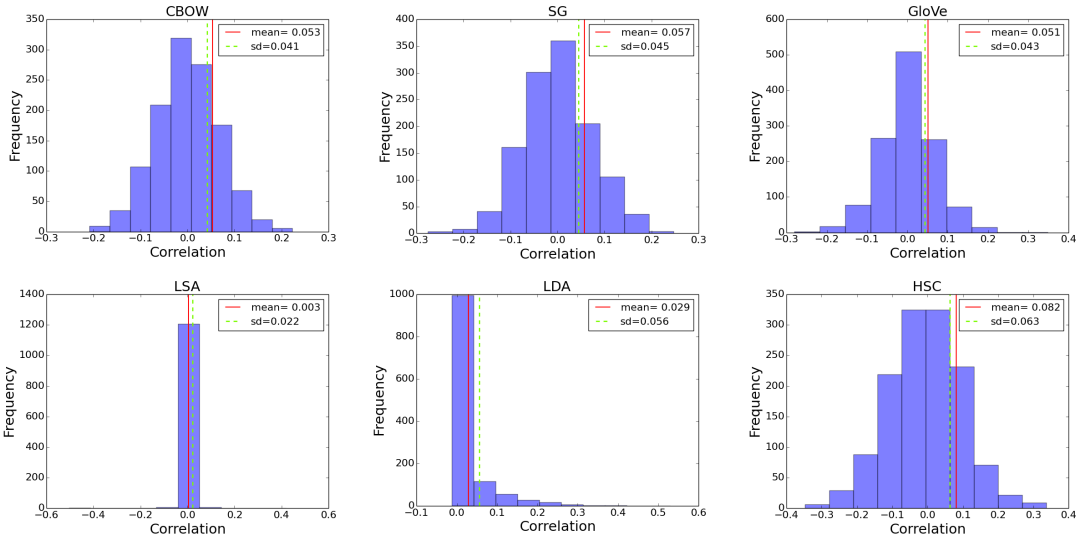

Figure 1: Cross-dimensional correlations for six word embeddings

which are dense and flat structured, we used Hierarchical Sparse Coding[7] (HSC) (Yogatama et al., 2015) to produce sparse and hierarchical word embeddings.

Given a word embedding matrix $\mathbf{W} \in \mathbb{R}^{m \times d}$, where each row correspond to the d-dimensional embedding of a word in a vocabulary containing m words, we compute a correlation matrix $\mathbf{C} \in \mathbb{R}^{d \times d}$, where the (i, j) element, C_{ij}, denotes the Pearson correlation coefficient between the i-th and j-th dimensions in the word embeddings over the m words. By construction $C_{ii} = 1$ and the histograms of the cross-dimensional correlations ($i \neq j$) are shown in Figure 1 for 50 dimensional word embeddings obtained from the six methods described above. The mean of the absolute pairwise correlations for each embedding type and the standard deviation (sd) are indicated in the figure.

From Figure 1, irrespective of the word embedding learning method used, we see that cross-dimensional correlations are distributed in a narrow range with an almost zero mean. This result empirically validates the uncorrelation assumption we used in our theoretical analysis. Moreover, this result indicates that Theorem 1 can be applied to a wide-range of existing word embeddings.

4.2 Learning Relation Representations

Our theoretical analysis in §3 claims that the performance of the bilinear relational embedding is independent of the tensor operator $\underline{\mathbf{A}}$. To empirically verify this claim, we conduct the following experiment. For this purpose, we use the BATS dataset (Gladkova et al., 2016) that contains of 40 semantic and syntactic relation types[8], and generate positive examples by pairing word-pairs that have the same relation types. Approximately each relation type has 1,225 word-pairs, which enables us to generate a total of 48k positive training instances (analogous word-pairs) of the form $((h, t), (h', t'))$. For each pair (h, t) related by a relation r, we randomly select pairs (h', t') with a different relation type r', according to the ℓ_2 distance between the two pairs to create negative (nonanalogous) instances.[9] We collectively refer both positive and negative training instances as the *training* dataset.

Using the $d = 50$ dimensional word embeddings from CBOW, SG, GloVe, LSA, LDA, and HSC methods created in §4.1, we learn relational embeddings according to (2) by minimising the ℓ_2 loss, (4). To avoid overfitting, we perform ℓ_2 regularisation on $\underline{\mathbf{A}}$, $\mathbf{P}$ and $\mathbf{Q}$ are regularised to diagonal matrices $p\mathbf{I}$ and $q\mathbf{I}$, for $p, q \in \mathbb{R}$. We initialise all parameters by uniformly sampling from $[-1, +1]$ and use AdaGrad (Duchi et al., 2011) with initial learning rate set to 0.01.

Figure 2 shows the Frobenius norm of the tensor $\underline{\mathbf{A}}$ (on the left vertical axis) and the values of p

[7] http://www.cs.cmu.edu/~ark/dyogatam/wordvecs/

[8] http://vsm.blackbird.pw/bats

[9] 10 negative instances are generated from each word-pair in our experiments.

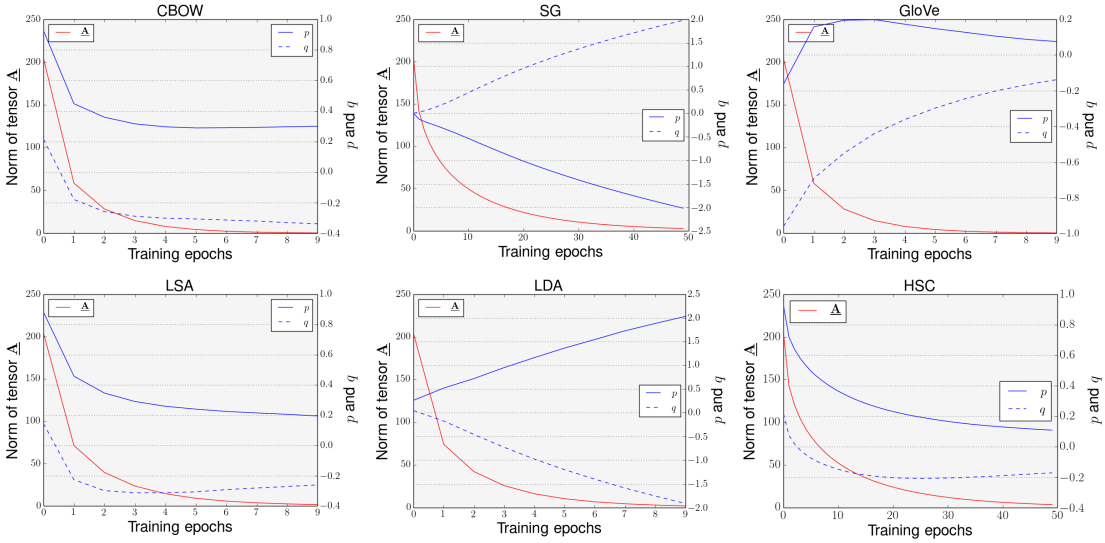

Figure 2: The learnt model parameters for different word embeddings of 50 dimensions.

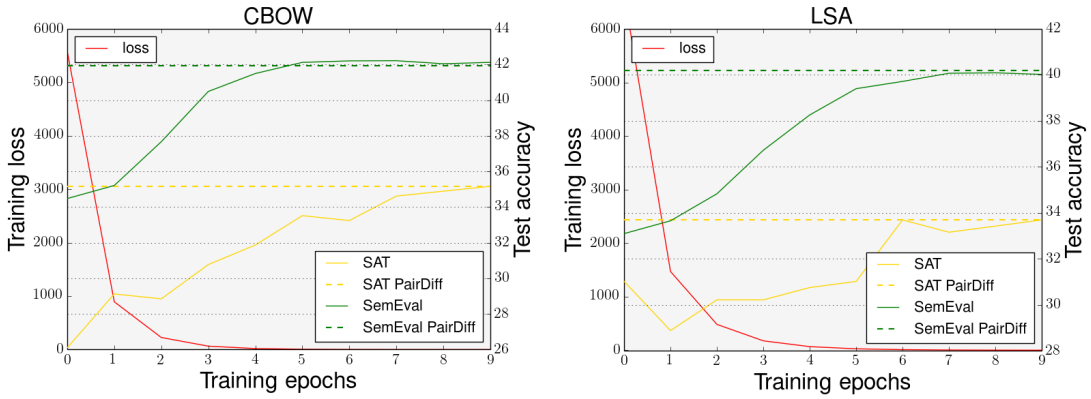

Figure 3: The training loss and test performance on SAT and SemEval benchmarks for relational embeddings.

and q (on the right vertical axis) for the six word embeddings. In all cases, we see that as the training progresses, $\underline{\mathbf{A}}$ goes to zero as predicted by Theorem 1 under regularisation. Moreover, we see that approximately $p \approx -q = c$ is reached for some $c \in \mathbb{R}$ in all cases, which implies that $\mathbf{P} \approx -\mathbf{Q} = c\mathbf{I}$, which is the PairDiff operator. Among the six input word embeddings compared in Figure 1, HSC has the highest mean correlation (0.082), which implies that its dimensions are correlated more than in the other word embeddings. This is to be expected by design because a hierarchical structure is imposed on the dimensions of the word embedding during training. However, HSC embeddings also satisfy the $\underline{\mathbf{A}} \approx \underline{\mathbf{0}}$ and $p \approx -q = c$ requirements, as expected by the PairDiff. This result shows that the claim of Theorem 1 is empirically true even when the uncorrelation assumption is mildly violated.

4.3 Generalisation Performance on Analogy Detection

So far we have seen that the bilinear relational representation given by (2) does indeed converge to the form predicted by our theoretical analysis for different types of word embeddings. However, it remains unclear whether the parameters learnt from the training instances generated from the BATS dataset accurately generalise to other benchmark datasets for analogy detection. To emphasize, our focus here is not to outperform relational representation methods proposed in previous works, but rather to empirically show that the learnt operator converges to the popular PairDiff for the analogy detection task.

To measure the generalisation capability of the learnt relational embeddings from BATS, we measure their performance on two other benchmark datasets: SAT Turney and Bigham (2003) and SemEval 2012-Task2[10]. Note that we *do not* retrain $\underline{\mathbf{A}}$, $\mathbf{P}$ and $\mathbf{Q}$ in (2) on SAT nor SemEval, but simply to use their values learnt from BATS because the purpose here to evaluate the generalisation of the learnt operator.

In SAT analogical questions, given a stem word-pair (a, b) with five candidate word-pairs (c, d), the task is to select the word-pair that is relationally similar to the the stem word-pair. The relational similarity between two word-pairs (a, b) and (c, d) is computed by the cosine similarity between the corresponding relational embeddings $\boldsymbol{r}(\boldsymbol{a}, \boldsymbol{b})$ and $\boldsymbol{r}(\boldsymbol{c}, \boldsymbol{d})$. The candidate word-pair that has the highest relational similarity with the stem word-pair is selected as the correct answer to a word analogy question. The reported accuracy is the ratio of the correctly answered questions to the total number of questions. On the other hand, SemEval dataset has 79 semantic relations, with each relation having ca. 41 word-pairs and four prototypical examples. The task is to assign a score for each word pair which is the average of the relational similarity between the given word-pair and prototypical word-pairs in a relation. Maximum difference scaling (MaxDiff) is used as the evaluation measure in this task.

Figure 3 shows the performance of the relational embeddings composed from 50-dimensional CBOW and LSA embeddings[11]. For CBOW, the level of performance reported by PairDiff on SAT and SemEval datasets are respectively 35.16% and 41.94%, and are shown by horizontal dashed lines. From Figure 3, we see that the training loss gradually decreases with the number of training epochs and the performance of the relational embeddings on SAT and SemEval datasets reach that of the PairDiff operator. This result indicates that the relational embeddings learnt not only converge to PairDiff operator on training data but also generalise to unseen relation types in SAT and SemEval test datasets.

5 Conclusion

This paper theoretically analyses the bilinear operator for representing relations between words using their embeddings. We showed that, if the word embeddings are standardised and uncorrelated, then the expected ℓ_2 distance between analogous and non-analogous word-pairs is independent of bilinear terms, and the relation embedding further simplifies to the popular PairDiff operator under regularised settings. Among diverse methods for calculating word embeddings, we empirically show the uncorrelation in word embedding dimensions, which is one of the prerequisites for simplifying the bilinear operator to a linear one. Empirically, we supports the theoretical analysis by showing that when optimising a general bilinear formulation on a labeled word pair relational dataset, the solution converges to the simple linear form, and more specifically to the simple PairDiff formulation.

In this work, we model relations as vectors and we measure relational strength using Euclidean distance. We are aware that there are many other relation representation methods and relational strength measurement methods besides what we have considered in the paper. Similar analysis can be conducted in follow-up work for different types of relation representations and strength measures. For instance, an interesting future research direction of this work is to extend the theoretical analysis to nonlinear relation composition operators, such as for nonlinear neural networks.

Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proc. of EMNLP*. pages 546–556.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

[10] `https://sites.google.com/site/semeval2012task2/`

[11] Similar trends were observed for all six word embedding types.

Danushka Bollegala, Takanori Maehara, and Ken ichi Kawarabayashi. 2015a. Embedding semantic relationas into word representations. In *Proc. of IJCAI*. pages 1222 – 1228.

Danushka Bollegala, Takanori Maehara, Yuichi Yoshida, and Ken ichi Kawarabayashi. 2015b. Learning word representations from relational graphs. In *Proc. of AAAI*. pages 2146 – 2152.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2009. A relational model of semantic similarity between words using automatically extracted lexical pattern clusters from the web. In *Proc. of EMNLP*. pages 803–812.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhenko. 2013. Translating embeddings for modeling multi-relational data. In *Proc. of NIPS*. pages 2787–2795.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *American Society for Information Science* 41(6):391–407.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proc. of COLING*. pages 3519–3530.

Nguyen Tuan Duc, Danushka Bollegala, and Mitsuru Ishizuka. 2010. Using relational similarity between word pairs for latent relational search on the web. In *Proc. of WI*. pages 196–199.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12:2121–2159.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015. Sparse overcomplete word vector representations. In *Proc. of ACL*. pages 1491–1500.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proc. of SRW@ HLT-NAACL*. pages 8–15.

Shu Guo, Quan Wang, Lihong Wang, Bin Wang, and Li Guo. 2016. Jointly embedding knowledge graphs and logical rules. In *Proc. of EMNLP*. pages 192–202.

Huda Hakami and Danushka Bollegala. 2017. Compositional approaches for representing relations between words: A comparative study. *Knowledge-Based Systems* 136:172–182.

Katsuhiko Hayashi and Masashi Shinbo. 2017. On the equivalence of holograpic and complex embeddings for link prediction. In *Proc. of ACL*. pages 554–559.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of Machine Learning Research*. pages 448–456.

David A. Jurgens, Saif Mohammad, Peter D. Turney, and Keith J. Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proc. of *SEM*. pages 356 – 364.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proc. of CoNLL*. pages 171–180.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of Association for Computational Linguistics* 3:211–225.

Tomas Mikolov, Kai Chen, and Jeffrey Dean. 2013a. Efficient estimation of word representation in vector space. In *Proc. of ICLR*. pages 1–12.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proc. of HLT-NAACL*. pages 746–751.

Pasquale Minervini, Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. 2017. Adversarial sets for regularising neural link predictors. In *Proc. of UAI*.

J. Mu, S. Bhat, and P. Viswanath. 2018. All-but-the-Top: Simple and Effective Postprocessing for Word Representations. In *Proc. of ICLR*.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. In *Proc. of the IEEE*. 1, pages 11–33.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proc. of AAAI*. pages 1955–1961.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proc. of ICML*. pages 809–816.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*. pages 1532–1543.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *Proc. of ICLR*.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proc. of NIPS*. pages 926–934.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proc. of ICML*. pages 2071–2080.

Peter D Turney and Jeffrey Bigham. 2003. Combining independent modules to solve multiple-choice synonym and analogy. In *Proc. of RANLP*. pages 482–489.

Peter D Turney and Michael L Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning* 60(1):251–278.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relational learning. In *Proc. of ACL*. pages 1671–1682.

Bishan Yang, Wen tau Yih, Xiadong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proc. of ICLR*.

Wenpeng Yin and Hinrich Schütze. 2016. Learning meta-embeddings by using ensembles of embedding sets. In *Proc. of ACL*. pages 1351–1360.

Dani Yogatama, Manaal Faruqui, Chris Dyer, and Noah A. Smith. 2015. Learning word representations with hierarchical sparse coding. In *Proc. of ICML*. pages 87 – 96.