Incorporating Relational Knowledge into Word Representations using Subspace Regularization

Abhishek Kumar
IBM Research
Yorktown Heights, NY 10598, USA
abhishk@us.ibm.com

Jun Araki
Carnegie Mellon University
Pittsburgh, PA 15213, USA
junaraki@cs.cmu.edu

Abstract

Incorporating lexical knowledge from semantic resources (e.g., WordNet) has been shown to improve the quality of distributed word representations. This knowledge often comes in the form of relational triplets (x, r, y) where words x and y are connected by a relation type r. Existing methods either ignore the relation types, essentially treating the word pairs as generic related words, or employ rather restrictive assumptions to model the relational knowledge. We propose a novel approach to model relational knowledge based on low-rank subspace regularization, and conduct experiments on standard tasks to evaluate its effectiveness.

1 Introduction

Distributed word representations, also known as *word embeddings*, are low-dimensional vector representations for words that capture semantic aspects (Bengio et al., 2003; Pennington et al., 2014; Mikolov et al., 2013a). The algorithms for learning the word embeddings rely on *distributional hypothesis* (Harris, 1954) that words occurring in similar contexts tend to have similar meanings. Word embeddings have been shown to capture interesting linguistic regularities by simple vector arithmetic (e.g., *v(king)-v(man)+v(woman)*$\approx$ *v(queen)*) (Mikolov et al., 2013c). They have also been used to derive downstream features for various NLP tasks, such as named entity recognition, chunking, dependency parsing, sentiment analysis, paraphrase detection and machine translation (Turian et al., 2010; Dhillon et al., 2011; Bansal et al., 2014; Maas et al., 2011; Socher et al., 2011; Zou et al., 2013). Their promise as semantic word representations has led to increasing research efforts on improving their quality.

To this end, researchers have attempted to incorporate lexical knowledge into word embeddings by using additional regularization or loss terms in the learning objective. This lexical knowledge is often available in the form of triplets $\{(w_i, r, w_j)\}$, where the words w_i and w_j are connected by relation type r. These methods can be broadly classified into two categories. First family of methods use a (over-)generalized notion of similarity between words and ignore the type of relations, essentially treating the two words as generic similar words (Yu and Dredze, 2014; Faruqui et al., 2015; Liu et al., 2015). This places an implicit restriction on the types of relations that can be used with these methods. Second family of methods model each relation type by a distinct operator. Bordes et al. (2013) assumed a distinct *relation vector* $\mathbf{r}$ for every relation and minimize the distance between the translated first word and the second word, i.e., $d(\mathbf{w}_i + \mathbf{r}, \mathbf{w}_j)$ for every triplet (w_i, r, w_j). Socher et al. (2013) proposed a neural tensor network which uses a distinct tensor operator for every relation. These methods were used to learn entity and relation embeddings from a large collection of relation triplets for the task of knowledge base completion. Since these methods did not use any co-occurrence information from a text corpus, all entities were required to appear at least once in the training data, ruling out generalization to unseen entities[1]. More recently, Xu et al. (2014) combined the training objective of `SKIP-GRAM` (Mikolov et al., 2013a) with the training objective of (Bordes et al., 2013) to incorporate lexical

[1]There exists work on relation extraction and knowledge-base completion that combines structured relation triplets and logical rules with unstructured text using various forms of latent variable models (Riedel et al., 2013; Chang et al., 2014; Toutanova et al., 2015; Rocktäschel et al., 2015).

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 506–511,
Berlin, Germany, August 7-12, 2016.

knowledge into word embeddings. Fried and Duh (2014) combine the training objective of (Bordes et al., 2013) with that of neural language model (Collobert et al., 2011) using *alternating direction method of multipliers* (Boyd et al., 2011).

Constant translation model (Bordes et al., 2013; Xu et al., 2014; Fried and Duh, 2014) (referred as CTM from now on), although an important step in modeling relational knowledge, makes a rather restrictive assumption requiring all triplets (w_i, r, w_j) pertaining to a relation type r to satisfy $\mathbf{w}_i + \mathbf{r} \approx \mathbf{w}_j, \forall(i,j)$. This restriction can be severe when learning from a large text corpus since vector representation of a word also needs to respect a huge set of co-occurrence instances with other words. CTM is also not suitable for (i) modeling symmetric relations (e.g., synonyms, antonyms), and (ii) modeling transitive relations (e.g., synonyms, hypernyms). In this paper, we propose a novel formulation for modeling the relational knowledge which addresses these issues by relaxing the constant translation assumption and modeling each relation by a low-rank subspace, i.e., all the word pairs pertaining to a relation are assumed to lie in a low-rank subspace. We demonstrate effectiveness of the learned word representations on the tasks of knowledge-base completion and word analogy.

2 Subspace-regularized word embedding

Although our proposed framework for relational modeling is general enough to use with any existing word embedding method, we work with Word2Vec model (Mikolov et al., 2013a) in this paper for illustrating our ideas and later for empirical evaluations. Word2Vec is a neural network model trained on sequence of words and its hidden layer activations can be read out as the word representations. Two variants were proposed in (Mikolov et al., 2013a) – `SKIP-GRAM`, which maximizes the log likelihood of the local context words given the target word, and `CBOW`, which maximizes the log likelihood of the target word given its local context. More specifically, `CBOW` maximizes the objective

$$\frac{1}{T}\sum_{t=1}^{T}\log p(w_t|w_{t-c}^{t+c}) = \frac{1}{T}\sum_{t=1}^{T}\frac{\exp(\mathbf{w}_t'^{\top}\mathbf{v}_t)}{\sum_{w\in V}\exp(\mathbf{w}'^{\top}\mathbf{v}_t)} \tag{1}$$

where w_{t-c}^{t+c} represents the words (or tokens) in the local context window around the t'th word (or token) and $\mathbf{v}_t = \sum_{-c\le i\le c, i\neq 0}\mathbf{w}_{t+i}$ can be seen as the average context vector. The vectors $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$ denote the *input* and *output* embeddings for word w, respectively. The *input embeddings* are taken as the final word representations. Negative sampling was proposed to efficiently optimize Eq. 1 (Mikolov et al., 2013b). We report empirical results with `CBOW` since it was computationally faster than `SKIP-GRAM` while giving similar results in our early explorations.

We assume access to relational knowledge in the form of triplets $R_k = \{(w_i, r_k, w_j)\}\,\forall 1 \le k \le m$, where words w_i and w_j are connected by relation r_k and R_k is the set of all triplets corresponding to relation r_k with $|R_k| = n_k$. This form of knowledge is commonly available from Knowledge Bases like WordNet (Fellbaum, 1998). Our framework is suitable for both symmetric relations where words can be interchanged (e.g., synonyms) and asymmetric relations which have a directional nature (e.g., hypernyms).

Let $\mathbf{d}_{ij} = (\mathbf{w}_j - \mathbf{w}_i) \in \mathbb{R}^d$ denote the *difference vector* for the triplet (w_i, r_k, w_j) which points from the vector of word w_i to that of word w_j. Let us construct a matrix $\mathbf{D}_k \in \mathbb{R}^{d\times n_k}$ by stacking the *difference vectors* corresponding to all the triplets in relation r_k, i.e.,

$$\mathbf{D}_k = [\cdots \mathbf{d}_{ij} \cdots]\,\forall\{(i,j) : (w_i, r_k, w_j) \in R_k\}. \tag{2}$$

To incorporate this relational knowledge into word embeddings, we enforce an approximate low-rank constraint on $\mathbf{D}_k$ assuming

$$\mathbf{D}_k \approx \mathbf{U}_k\mathbf{A}_k^{\top}, \tag{3}$$

where $\mathbf{U}_k \in \mathbb{R}^{d\times p}$, $p \ll d$ is the relation basis whose linear span contains all the difference vectors pertaining to relation r_k. For $p = 2$, this assumption implies that all the difference vectors pertaining to a relation lie in a 2-D plane. For $p = 1$, it reduces to $\mathbf{D}_k \approx \mathbf{u}_k\boldsymbol{\alpha}_k^T$, $\mathbf{u}_k \in \mathbb{R}^d, \boldsymbol{\alpha}_k \in \mathbb{R}^{n_k}$, implying that all the difference vectors for a relation are collinear. In this paper, we mainly study the rank-1 model (p=1) since it seems to be a natural starting point for evaluating the idea of subspace-regularized relational modeling. The study of higher rank models will potentially require a careful exploration of various structural regularizers for reconstruction matrix $\mathbf{A}_k$ as well as a different evaluation scheme. We leave this study for future work.

Rank-1 subspace regularization can also be motivated from the fact that word embeddings are able to capture some linguistic regularities (Mikolov et al., 2013c) along certain directions in the vector space. For example, the *difference vector* for word pair *(king, queen)* is approximately aligned with the difference vector for *(man,woman)*, encoding the *gender* relation. The direction of the difference vectors carries significant information for these regularities which is evident from the success of *cosine* similarity metrics in the word analogy problems (Levy et al., 2014). CTM that assumes $\mathbf{w}_i + \mathbf{u}_k = \mathbf{w}_j \, \forall (w_i, r_k, w_j) \in R$ enforces an additional equal length constraint on the *difference vectors*, which may be rather restrictive, especially when the word vectors are also influenced by co-occurrence statistics (apart from relational knowledge). Moreover, it may face following challenges in relational modeling:

- It does not have a natural interpretation for modeling symmetric relations (e.g., synonyms, antonyms) that allow interchangeability of words in a given relation triplet (i.e., $(w_i, r_k, w_j) \iff (w_j, r_k, w_i)$). Having a constant translation of $\mathbf{u}_k \in \mathbb{R}^d$ from the first word to the second word leads to contradiction.

- It is also not natural for modeling relations with transitive property (i.e., $(w_i, r_k, w_j) \wedge (w_j, r_k, w_l) \implies (w_i, r_k, w_l)$), again leading to contradictions. Common examples of such relations are synonyms and hypernyms.

The proposed rank-1 subspace relation model naturally allows for modeling such relations by doing away with the constant length restriction on the difference vectors. Our empirical evaluations verify that this relaxation indeed leads to improved quality of word vectors with respect to capturing linguistic regularities.

We incorporate the proposed relational model into the learning objective for word vectors by regularizing the matrix of *difference vectors* towards a rank-1 matrix. We impose a nonnegativity constraint on the reconstruction coefficients $\boldsymbol{\alpha}_k$ if relation r_k is asymmetric. This respects the unidirectional nature of asymmetric relations. To ensure uniqueness of solution for $\mathbf{u}_k$ and $\boldsymbol{\alpha}_k$, we constrain $\|\mathbf{u}_k\|_2 = 1$. Leaving $\boldsymbol{\alpha}_k$ completely free can end up creating spurious relations between any two words that are arbitrarily far but whose difference vector is directionally aligned with any of the relation basis vectors $\{\mathbf{u}_k\}_{k=1}^m$. To avoid this, we further impose a upper limit of c on the absolute value of elements of $\boldsymbol{\alpha}_k$. We minimize the following joint objective for word vectors $\{\mathbf{w}_i\}_{i=1}^{|V|}$ and relation parameters $\{\mathbf{u}_k, \boldsymbol{\alpha}_k\}_{k=1}^m$:

$$- \frac{1}{T} \sum_{t=1}^{T} \log p(w_t | w_{t-c}^{t+c}) + \frac{\lambda}{2\,|R|} \sum_{k=1}^{m} \left\| \mathbf{D}_k - \mathbf{u}_k \boldsymbol{\alpha}_k^\top \right\|_F^2$$
$$\text{s.t. } \boldsymbol{\alpha}_k \geq 0 \, \forall \, \text{asymmetric } r_k, \|\mathbf{u}_k\|_2 = 1, |(\boldsymbol{\alpha}_k)_l| \leq c. \tag{4}$$

where $\mathbf{D}_k$ is the matrix of *difference vectors* as defined earlier and λ is the regularization parameter. The first term in the objective takes into account the co-occurrence information text corpus while the second term incorporates the relational knowledge.

Optimizing for word vectors: We adopt parallel asynchronous stochastic gradient descent (SGD) with negative sampling approach of (Mikolov et al., 2013b). The model parameters for optimization are input embeddings (weights connecting input and hidden layer) and output embeddings (weights connecting hidden and output layer). Input embeddings are taken as the final word embeddings. Each computing thread works with a predefined segment of the text corpus and updates parameters that are stored in a shared memory. In each gradient step of `CBOW`, a thread samples a target word and its local context window and updates the parameters of the neural network. It can be seen as sampling one of the $f_t(\cdot)$, $t = 1, 2, \ldots, T$ and taking a gradient step with it. A small number of random target words are also sampled for the same context, treating them as negative examples for the gradient update. In the `CBOW` architecture, representations for context words are directly encoded as columns of the linear weight matrix $\mathbf{W} \in \mathbb{R}^{d \times |V|}$ that maps input bag-of-words layer to the hidden layer. The columns of $\mathbf{W}$ are taken as the word embeddings for the corresponding words in the vocabulary V. The reader is referred to (Mikolov et al., 2013b; Goldberg and Levy, 2014) for more details on the optimization procedure for `CBOW`. If a word appears in the set of relation triplets R, our regularization term gets activated. Since we place the regularizer only on input embeddings, the following gradient updates due to the regularization term act only on input

embeddings.

$$\mathbf{w}_i \longleftarrow \mathbf{w}_i - \eta \frac{\lambda}{|R|} \Bigg[\sum_{j:(w_i,r_k,w_j)\in R} \left(\mathbf{w}_i - \mathbf{w}_j + \mathbf{u}_k \alpha_{k_{ij}}\right) + \sum_{j:(w_j,r_k,w_i)\in R} \left(\mathbf{w}_i - \mathbf{w}_j - \mathbf{u}_k \alpha_{k_{ji}}\right) \Bigg], \tag{5}$$

where η is the learning rate, and $\alpha_{k_{ij}}$ denotes the element of $\boldsymbol{\alpha}_k$ corresponding to the column of matrix $\mathbf{D}_k$ which contains *difference vector* $(\mathbf{w}_j - \mathbf{w}_i)$ (and similarly for $\boldsymbol{\alpha}_{k_{ji}}$). The modifications in the learning rate as the SGD progresses are kept same as in the original implementation of CBOW[2].

Optimization for $\mathbf{u}_k$ and $\boldsymbol{\alpha}_k$: Instead of having stochastic gradient updates, we adopt an asynchronous batch update strategy for relation basis $\{\mathbf{u}_k\}_{k=1}^m$ and reconstruction coefficients $\{\boldsymbol{\alpha}_k\}_{k=1}^m$. We launch one compute thread that keeps solving the batch optimization problem for $\{\mathbf{u}_k\}_{k=1}^m$ and $\{\boldsymbol{\alpha}_k\}_{k=1}^m$ in an infinite loop until the optimization for word embeddings finishes. The batch optimization problem for a symmetric relation r_k is:

$$\min_{\mathbf{u}_k,\boldsymbol{\alpha}_k} \left\| \mathbf{D}_k - \mathbf{u}_k \boldsymbol{\alpha}_k^\top \right\|_F^2, \text{ s.t. } \|\mathbf{u}_k\|_2 = 1, |\boldsymbol{\alpha}_k| \leq c. \tag{6}$$

where $\mathbf{D}_k \in \mathbb{R}^{d \times n_k}$ is the matrix of difference vectors for all triplets corresponding to relation r_k as defined in Eq. 2. Without the absolute value constraint on $\boldsymbol{\alpha}_k$, this problem can be exactly solved by SVD. We follow an alternating optimization procedure for solving this problem. We initialize $\mathbf{u}_k$ to the top left singular vector of $\mathbf{D}_k$ and then alternate between solving two least-squares sub-problems for $\mathbf{u}_k$ and $\boldsymbol{\alpha}_k$ respectively with the corresponding constraints. For asymmetric relations, there is an additional nonnegativity constraint on $\boldsymbol{\alpha}_k$. We use projected gradient descent to solve these constrained least-squares problems.

3 Empirical Observations

We report preliminary evaluations of the proposed model (termed as RELSUB) on the tasks of word analogy and knowledge base completion. We use English Wikipedia for training which contains approximately 4.8 million articles and 2 billion tokens. We lowercase all the text and and tokenize using Stanford NLP tokenizer.

Relation-type	RELCONST	RELSUB
capital-cities	48.15	**59.26**
currency	**58.33**	50.00
city-in-state	17.88	**18.94**
gender	44.44	**50.00**
similar-to	5.44	**7.26**
made-of	0	0
has-context	**10.00**	8.26
is-a	1.35	**1.83**
part-of	17.50	**19.00**
instance-of	8.40	**12.98**
derived-from	9.14	**10.27**
antonym	20.00	**20.62**
entails	0	**4.35**
causes	0	0
member-of	13.43	**26.87**
related-to	0	0
attribute	**11.76**	8.82
SEMANTIC	7.47	**8.44**
adjective-to-adverb	10.14	**47.83**
plural-verbs	61.25	**71.77**
plural-nouns	66.70	**71.89**
comparative	70.00	**75.00**
superlative	66.67	**77.78**
nationality	**85.71**	**85.71**
past-tense	42.20	**66.84**
present-participle	45.76	**47.62**
SYNTACTIC	54.88	**65.38**
TOTAL	24.61	**29.03**

Table 1: WordRep data: Accuracy on knowledge-base completion

We use two datasets for evaluating the proposed method. **Google word analogy data** (Mikolov et al., 2013a) contains 19544 analogy relations (14 relation types – 5 semantic, 9 syntactic) of the form *a:b::c:d* constructed from 550 unique relation triplets. We use this data only for evaluation (test phase). **WordRep** (Gao et al., 2014) contains a large collection of relation triplets (44584 triplets in total, 25 relation types – 18 semantic, 7 syntactic) extracted from WordNet, Wikipedia and Dictionary. For each relation type, we randomly split the triplets in $4:1$ ratio with larger split used for training and smaller split used for test. We make sure that there is no word overlap between training and test triplets. We also remove triplets containing words from Google Analogy data from the training set.

We compare the proposed RELSUB model with two methods: (i) **CBOW** (Mikolov et al., 2013a), and (ii) **RELCONST** which is based on constant translation model for relations which was originally proposed in (Bordes et al., 2013) for embedding knowledge-bases and was recently used by

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

Relation-type	CBOW	RELCONST	RELSUB
SEMANTIC	68.37	69.85	**70.96**
SYNTACTIC	**66.69**	65.42	65.96
TOTAL	67.48	67.43	**68.22**

Table 2: Google analogy data: Accuracy on word analogy task

(Xu et al., 2014) for learning word embeddings. Our objective for RELCONST is same as Eq. 4 except that $\{\boldsymbol{\alpha}_k\}_{k=1}^m$ are set equal to the vector of all 1's and norm constraint on $\mathbf{u}_k$ are removed. This enables us to directly test the merit of the proposed rank-1 subspace relational model over that of constant translational model in the same regularization framework. Note that this objective is similar in spirit to (Xu et al., 2014) in the sense that it also uses a constant translation model for relations. However, Xu et al. (Xu et al., 2014) employ a maximum margin objective on the relation triplets as originally proposed in (Bordes et al., 2013). It encourages the loss (measured in terms of ℓ_2 distance) for true relation triplets to be smaller than the loss for randomly corrupted relation triplets. Instead of a maximum margin objective for relational knowledge, our model uses a simpler regularization based objective. We could not obtain the implementation of RC-NET (Xu et al., 2014) due to copyright issues cited by its authors. We also cannot compare with approaches that use only knowledge-base for training (Faruqui et al., 2015) since they do not learn or modify the embeddings of unseen words and our evaluation triplets do not overlap with training triplets.

We use the CBOW implementation in publicly available Word2Vec code[3] for our experiments. Our vocabulary has $400k$ words and we use a dimensionality of 300 for embeddings. For all other parameters, we use default values that the Word2Vec code comes with including a context window size of 5 tokens to each side, 5 negative samples per positive sample for negative sampling technique, etc. For both RELSUB and RELCONST, we set the regularization parameter to $\frac{\lambda}{|R|} = 1e^{-4}$ in all our experiments. We set the upper limit c in Eq. 4 to 1. The parameters were not fine tuned rigorously but these values seemed to work reasonably well for us. We do total 5 epochs of SGD over the text corpus for all methods.

In knowledge-base completion task, we want to predict the missing word of a relation triplet. For a triplet (x, r, y), we assume that x (first word) and r (relation type) are observed and the task is to predict the missing word y. We restrict the search for the missing word to the most frequent $300k$ words (75% of the vocabulary). The missing word is predicted to be the closest word along the rank-1 subspace spanned by the relation vector (restricted by c in Eq. 4). For RELCONST, the missing word is predicted by translating the first word by the relation vector and then searching for nearest word. The accuracy results on WordRep data are shown in Table 1. Relaxing the constant translation to rank-1 subspace assumption results in significant improvements on this task.

In the analogy task, we want to predict the missing word in an analogy tuple *a:b::c:?*. We use the Google word-analogy data (Mikolov et al., 2013a) for this evaluation. We observe considerable gains with RELSUB over CBOW for semantic categories. The accuracy of knowledge regularized methods on syntactic categories is a little worse than CBOW and only slightly better than RELCONST, which is contrary to our observation on the knowledge-base completion task. This is due to the fact that analogy task uses the difference vector $(\mathbf{b} - \mathbf{a})$ instead of the learned relation vector which is assumed to be unknown.

4 Concluding Remarks

We proposed a novel framework for modeling relational knowledge in word embeddings using rank-1 subspace regularization. Our model can be seen as a generalization of the constant translational model for relations (Bordes et al., 2013; Xu et al., 2014). In the future, we would like to study the interplay between word frequencies and the strength of regularization, and perform an exhaustive empirical evaluation. The study of higher rank subspaces for relation modeling is also an important future direction.

References

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of ACL 2014*, pages 809–815.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko.

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

2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS 2013*, pages 2787–2795.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *EMNLP*.

Paramveer S. Dhillon, Dean Foster, and Lyle Ungar. 2011. Multi-view learning of word embeddings via CCA. In *Proceedings of NIPS 2011*, pages 199–207.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL-HLT 2015*, pages 1606–1615.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Daniel Fried and Kevin Duh. 2014. Incorporating both distributional and relational semantics in word representations. *arXiv preprint arXiv:1412.4369*.

Bin Gao, Jiang Bian, and Tie-Yan Liu. 2014. Wordrep: A benchmark for research on learning word representations. *arXiv preprint arXiv:1407.1640*.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Omer Levy, Yoav Goldberg, and Israel Ramat-Gan. 2014. Linguistic regularities in sparse and explicit word representations. *CoNLL-2014*, page 171.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015. Learning semantic word embeddings based on ordinal knowledge constraints. *Proceedings of ACL 2015*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL 2011*, pages 142–150.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of ICLR 2013*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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

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

Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. 2013. Relation extraction with matrix factorization and universal schemas. In *HLT-NAACL*.

Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. 2015. Injecting logical background knowledge into embeddings for relation extraction. In *HLT-NAALC*.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of NIPS 2011*, pages 801–809.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. *Proceedings of ACL*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL 2010*, pages 384–394.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. RC-NET: A general framework for incorporating knowledge into word representations. In *Proceedings of CIKM 2014*, pages 1219–1228.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of ACL 2014*, pages 545–550.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP 2013*, pages 1393–1398.