

A Relational Memory-based Embedding Model for Triple Classification and Search Personalization

Dai Quoc Nguyen¹, Tu Dinh Nguyen², Dinh Phung¹

¹Monash University, Australia

²Trusting Social

¹{dai.nguyen, dinh.phung}@monash.edu

²tu@trustingsocial.com

Abstract

Knowledge graph embedding methods often suffer from a limitation of memorizing valid triples to predict new ones for triple classification and search personalization problems. To this end, we introduce a novel embedding model, named R-MeN, that explores a relational memory network to encode potential dependencies in relationship triples. R-MeN considers each triple as a sequence of 3 input vectors that recurrently interact with a memory using a transformer self-attention mechanism. Thus R-MeN encodes new information from interactions between the memory and each input vector to return a corresponding vector. Consequently, R-MeN feeds these 3 returned vectors to a convolutional neural network-based decoder to produce a scalar score for the triple. Experimental results show that our proposed R-MeN obtains state-of-the-art results on SEARCH17 for the search personalization task, and on WN11 and FB13 for the triple classification task.

1 Introduction

Knowledge graphs (KGs) – representing the genuine relationships among entities in the form of triples (*subject, relation, object*) denoted as (s, r, o) – are often insufficient for knowledge presentation due to the lack of many valid triples (West et al., 2014). Therefore, research work has been focusing on inferring whether a new triple missed in KGs is likely valid or not (Bordes et al., 2011, 2013; Socher et al., 2013). As summarized in (Nickel et al., 2016; Nguyen, 2017), KG embedding models aim to compute a score for each triple, such that valid triples have higher scores than invalid ones.

Early embedding models such as TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransR (Lin et al., 2015), TransD (Ji et al., 2015), DISTMULT (Yang et al., 2015) and ComplEx (Trouillon et al., 2016) often employ simple linear oper-

ators such as addition, subtraction and multiplication. Recent embedding models such as ConvE (Dettmers et al., 2018) and CapsE (Nguyen et al., 2019b) successfully apply deep neural networks to score the triples.

Existing embedding models are showing promising performances mainly for knowledge graph completion, where the goal is to infer a missing entity given a relation and another entity. But in real applications, less mentioned, such as triple classification (Socher et al., 2013) that aims to predict whether a given triple is valid, and search personalization (Vu et al., 2017) that aims to re-rank the relevant documents returned by a user-oriented search system given a query, these models do not effectively capture potential dependencies among entities and relations from existing triples to predict new triples.

To this end, we leverage the relational memory network (Santoro et al., 2018) to propose R-MeN to infer a valid fact of new triples. In particular, R-MeN transforms each triple along with adding positional embeddings into a sequence of 3 input vectors. R-MeN then uses a transformer self-attention mechanism (Vaswani et al., 2017) to guide the memory to interact with each input vector to produce an encoded vector. As a result, R-MeN feeds these 3 encoded vectors to a convolutional neural network (CNN)-based decoder to return a score for the triple. In summary, our main contributions are as follows:

- We present R-MeN – a novel KG embedding model to memorize and encode the potential dependencies among relations and entities for two real applications of triple classification and search personalization.
- Experimental results show that R-MeN obtains better performance than up-to-date embedding models, in which R-MeN produces new state-of-the-art results on SEARCH17

for the search personalization task, and a new highest accuracy on WN11 and the second-highest accuracy on FB13 for the triple classification task.

2 The proposed R-MeN

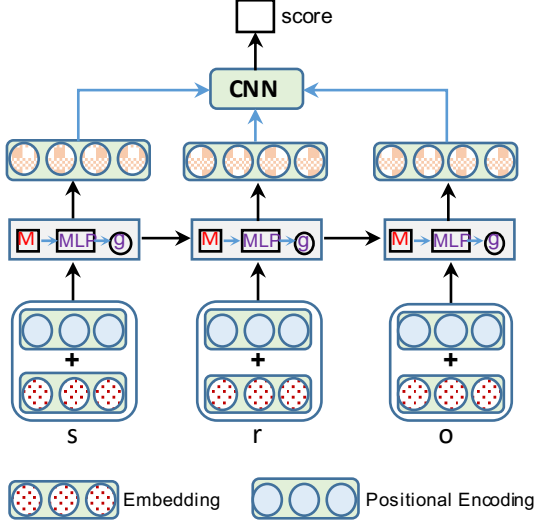


Figure 1: Processes in our proposed R-MeN for an illustration purpose. “M” denotes a memory. “MLP” denotes a multi-layer perceptron. “g” denotes a memory gating. “CNN” denotes a convolutional neural network-based decoder.

Let \mathcal{G} be a KG database of valid triples in the form of $(subject, relation, object)$ denoted as (s, r, o) . KG embedding models aim to compute a score for each triple, such that valid triples obtain higher scores than invalid triples.

We denote $\mathbf{v}_s, \mathbf{v}_r$ and $\mathbf{v}_o \in \mathbb{R}^d$ as the embeddings of s, r and o , respectively. Besides, we hypothesize that relative positions among s, r and o are useful to reason instinct relationships; hence we add to each position a positional embedding. Given a triple (s, r, o) , we obtain a sequence of 3 vectors $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ as:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{W}(\mathbf{v}_s + \mathbf{p}_1) + \mathbf{b} \\ \mathbf{x}_2 &= \mathbf{W}(\mathbf{v}_r + \mathbf{p}_2) + \mathbf{b} \\ \mathbf{x}_3 &= \mathbf{W}(\mathbf{v}_o + \mathbf{p}_3) + \mathbf{b} \end{aligned}$$

where $\mathbf{W} \in \mathbb{R}^{k \times d}$ is a weight matrix, and $\mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{p}_3 \in \mathbb{R}^d$ are positional embeddings, and k is the memory size.

We assume we have a memory M consisting of N rows wherein each row is a memory slot. We use $M^{(t)}$ to denote the memory at timestep t , and $M_{i,:}^{(t)} \in \mathbb{R}^k$ to denote the i -th memory slot

at timestep t . We follow Santoro et al. (2018) to take \mathbf{x}_t to update $M_{i,:}^{(t)}$ using the multi-head self-attention mechanism (Vaswani et al., 2017) as:

$$\begin{aligned} \hat{M}_{i,:}^{(t+1)} &= [\hat{M}_{i,:}^{(t+1),1} \oplus \hat{M}_{i,:}^{(t+1),2} \oplus \\ &\quad \dots \oplus \hat{M}_{i,:}^{(t+1),H}] \\ \text{with } \hat{M}_{i,:}^{(t+1),h} &= \alpha_{i,N+1,h} (\mathbf{W}^{h,V} \mathbf{x}_t) \\ &\quad + \sum_{j=1}^N \alpha_{i,j,h} (\mathbf{W}^{h,V} M_{j,:}^{(t)}) \end{aligned}$$

where H is the number of attention heads, and \oplus denotes a vector concatenation operation. Regarding the h -th head, $\mathbf{W}^{h,V} \in \mathbb{R}^{n \times k}$ is a value-projection matrix, in which n is the head size and $k = nH$. Note that $\{\alpha_{i,j,h}\}_{j=1}^N$ and $\alpha_{i,N+1,h}$ are attention weights, which are computed using the softmax function over scaled dot products as:

$$\begin{aligned} \alpha_{i,j,h} &= \frac{\exp(\beta_{i,j,h})}{\sum_{m=1}^{N+1} \exp(\beta_{i,m,h})} \\ \alpha_{i,N+1,h} &= \frac{\exp(\beta_{i,N+1,h})}{\sum_{m=1}^{N+1} \exp(\beta_{i,m,h})} \\ \text{with } \beta_{i,j,h} &= \frac{(\mathbf{W}^{h,Q} M_{i,:}^{(t)})^\top (\mathbf{W}^{h,K} M_{j,:}^{(t)})}{\sqrt{n}} \\ \beta_{i,N+1,h} &= \frac{(\mathbf{W}^{h,Q} M_{i,:}^{(t)})^\top (\mathbf{W}^{h,K} \mathbf{x}_t)}{\sqrt{n}} \end{aligned}$$

where $\mathbf{W}^{h,Q} \in \mathbb{R}^{n \times k}$ and $\mathbf{W}^{h,K} \in \mathbb{R}^{n \times k}$ are query-projection and key-projection matrices, respectively. As following Santoro et al. (2018), we feed a residual connection between \mathbf{x}_t and $\hat{M}_{i,:}^{(t+1)}$ to a multi-layer perceptron followed by a memory gating to produce an encoded vector $\mathbf{y}_t \in \mathbb{R}^k$ for timestep t and the next memory slot $M_{i,:}^{(t+1)}$ for timestep $(t+1)$.

As a result, we obtain a sequence of 3 encoded vectors $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$ for the triple (s, r, o) . We then use a CNN-based decoder to compute a score for the triple as:

$$f(s, r, o) = \max(\text{ReLU}([\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3] * \Omega))^\top \mathbf{w}$$

where we view $[\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3]$ as a matrix in $\mathbb{R}^{k \times 3}$; Ω denotes a set of filters in $\mathbb{R}^{m \times 3}$, in which m is the window size of filters; $\mathbf{w} \in \mathbb{R}^{|\Omega|}$ is a weight vector; $*$ denotes a convolution operator; and \max denotes a max-pooling operator. Note that we use the max-pooling operator – instead of the vector

concatenation of all feature maps used in ConvKB (Nguyen et al., 2018) – to capture the most important feature from each feature map, and to reduce the number of weight parameters.

We illustrate our proposed R-MeN as shown in Figure 1. In addition, we employ the Adam optimizer (Kingma and Ba, 2014) to train R-MeN by minimizing the following loss function (Trouillon et al., 2016; Nguyen et al., 2018):

$$\mathcal{L} = \sum_{(s,r,o) \in \{\mathcal{G} \cup \mathcal{G}'\}} \log(1 + \exp(-t_{(s,r,o)} \cdot f(s,r,o)))$$

$$\text{in which, } t_{(s,r,o)} = \begin{cases} 1 & \text{for } (s,r,o) \in \mathcal{G} \\ -1 & \text{for } (s,r,o) \in \mathcal{G}' \end{cases}$$

where \mathcal{G} and \mathcal{G}' are collections of valid and invalid triples, respectively. \mathcal{G}' is generated by corrupting valid triples in \mathcal{G} .

3 Experimental setup

3.1 Task description and evaluation

3.1.1 Triple classification

The triple classification task is to predict whether a given triple (s, r, o) is valid or not (Socher et al., 2013). Following Socher et al. (2013), we use two benchmark datasets WN11 and FB13, in which each validation or test set consists of the same number of valid and invalid triples. It is to note in the test set that Socher et al. (2013) did not include triples that either or both of their subject and object entities also appear in a different relation type or order in the training set, to avoid reversible relation problems. Table 1 gives statistics of the experimental datasets.

Dataset	# \mathcal{E}	# \mathcal{R}	#Triples in train/valid/test		
FB13	75,043	13	316,232	11,816	47,466
WN11	38,696	11	112,581	5,218	21,088

Table 1: Statistics of the experimental datasets. # \mathcal{E} is the number of entities. # \mathcal{R} is the number of relations.

Each relation r has a threshold θ_r computed by maximizing the micro-averaged classification accuracy on the validation set. If the score of a given triple (s, r, o) is above θ_r , then this triple is classified as a valid triple, otherwise, it is classified as an invalid one.

3.1.2 Search personalization

In search personalization, given a submitted *query* for a *user*, we aim to re-rank the *documents* returned by a search system, so that the more the

returned documents are relevant for that query, the higher their ranks are. We follow (Vu et al., 2017; Nguyen et al., 2019a,b) to view a relationship of the submitted query, the user and the returned document as a (s, r, o) -like triple (*query, user, document*). Therefore, we can adapt our R-MeN for the search personalization task.

We evaluate our R-MeN on the benchmark dataset SEARCH17 (Vu et al., 2017) as follows: (i) We train our model and use the trained model to compute a score for each (*query, user, document*) triple. (ii) We sort the scores in the descending order to obtain a new ranked list. (iii) We employ two standard evaluation metrics: mean reciprocal rank (MRR) and Hits@1. For each metric, the higher value indicates better ranking performance.

3.2 Training protocol

3.2.1 Triple classification

We use the common Bernoulli strategy (Wang et al., 2014; Lin et al., 2015) when sampling invalid triples. For WN11, we follow Guu et al. (2015) to initialize entity and relation embeddings in our R-MeN by averaging word vectors in the relations and entities, i.e., $\mathbf{v}_{\text{american_arborvitae}} = \frac{1}{2}(\mathbf{v}_{\text{american}} + \mathbf{v}_{\text{arborvitae}})$, in which these word vectors are taken from the Glove 50-dimensional pre-trained embeddings (Pennington et al., 2014) (i.e., $d = 50$). For FB13, we use entity and relation embeddings produced by TransE to initialize entity and relation embeddings in our R-MeN, for which we obtain the best result for TransE on the FB13 validation set when using l_2 -norm, learning rate at 0.01, margin $\gamma = 2$ and $d = 50$.

Furthermore, on WN11, we provide our new fine-tuned result for TransE using our experimental setting, wherein we use the same initialization taken from the Glove 50-dimensional pre-trained embeddings to initialize entity and relation embeddings in TransE. We get the best score for TransE on the WN11 validation set when using l_1 -norm, learning rate at 0.01, margin $\gamma = 6$ and $d = 50$.

In preliminary experiments, we see the highest accuracies on the validation sets for both datasets when using a single memory slot (i.e., $N = 1$); and this is consistent with utilizing the single memory slot in language modeling (Santoro et al., 2018). Therefore, we set $N = 1$ to use the single memory slot for the triple classification task. Also from preliminary experiments, we select the batch size $bs = 16$ for WN11 and $bs = 256$ for FB13, and

set the window size m of filters to 1 (i.e., $m = 1$).

Regarding other hyper-parameters, we vary the number of attention heads H in $\{1, 2, 3\}$, the head size n in $\{128, 256, 512, 1024\}$, the number of MLP layers l in $\{2, 3, 4\}$, and the number of filters $F = |\Omega|$ in $\{128, 256, 512, 1024\}$. The memory size k is set to be $nH = k$. To learn our model parameters, we train our model using the Adam initial learning rate lr in $\{1e^{-6}, 5e^{-6}, 1e^{-5}, 5e^{-5}, 1e^{-4}, 5e^{-4}\}$. We run up to 30 epochs and use a grid search to select the optimal hyper-parameters. We monitor the accuracy after each training epoch to compute the relation-specific threshold θ_r to get the optimal hyper-parameters (w.r.t the highest accuracy) on the validation set, and to report the final accuracy on the test set.

3.2.2 Search personalization

We use the same initialization of user profile, query and document embeddings used by Nguyen et al. (2019b) on SEARCH17 to initialize the corresponding embeddings in our R-MeN respectively. From the preliminary experiments, we set $N = 1$, $bs = 16$ and $m = 1$. Other hyper-parameters are varied as same as used in the triple classification task. We monitor the MRR score after each training epoch to obtain the highest MRR score on the validation set to report the final scores on the test set.

4 Main results

4.1 Triple classification

Table 2 reports the accuracy results of our R-MeN model and previously published results on WN11 and FB13. R-MeN sets a new state-of-the-art accuracy of 90.5% that significantly outperforms other models on WN11. R-MeN also achieves a second highest accuracy of 88.9% on FB13. Overall, R-MeN yields the best performance averaged over these two datasets.

Regarding TransE, we obtain the second-best accuracy of 89.2% on WN11 and a competitive accuracy of 88.1% on FB13. Figure 2 shows the accuracy results for TransE and our R-MeN w.r.t each relation. In particular, on WN11, the accuracy for the one-to-one relation “similar_to” significantly increases from 50.0% for TransE to 78.6% for R-MeN. On FB13, R-MeN improves the accuracies over TransE for the many-to-many relations “institution” and “profession”.

Method	WN11	FB13	Avg.
NTN (Socher et al., 2013)	86.2	87.2	86.7
TransH (Wang et al., 2014)	78.8	83.3	81.1
TransR (Lin et al., 2015)	85.9	82.5	84.2
TransD (Ji et al., 2015)	86.4	89.1	87.8
TransR-FT (Feng et al., 2016)	86.6	82.9	84.8
TransSparse-S (Ji et al., 2016)	86.4	88.2	87.3
TransSparse-US (Ji et al., 2016)	86.8	87.5	87.2
ManifoldE (Xiao et al., 2016a)	87.5	87.2	87.4
TransG (Xiao et al., 2016b)	87.4	87.3	87.4
lppTransD (Yoon et al., 2016)	86.2	88.6	87.4
ConvKB (Nguyen et al., 2019a)	87.6	88.8	88.2
TransE (Bordes et al., 2013) (ours)	<u>89.2</u>	88.1	<u>88.7</u>
Our R-MeN model	90.5	<u>88.9</u>	89.7
TransE-NMM (Nguyen et al., 2016)	86.8	88.6	87.7
TEKE.H (Wang and Li, 2016)	84.8	84.2	84.5
Bilinear-COMP (Guu et al., 2015)	87.6	86.1	86.9
TransE-COMP (Guu et al., 2015)	84.9	87.6	86.3

Table 2: Accuracy results (in %) on the WN11 and FB13 test sets. The last 4 rows report accuracies of the models that use relation paths or incorporate with a large external corpus. The best score is in bold while the second best score is in underline. “Avg.” denotes the averaged accuracy over two datasets.

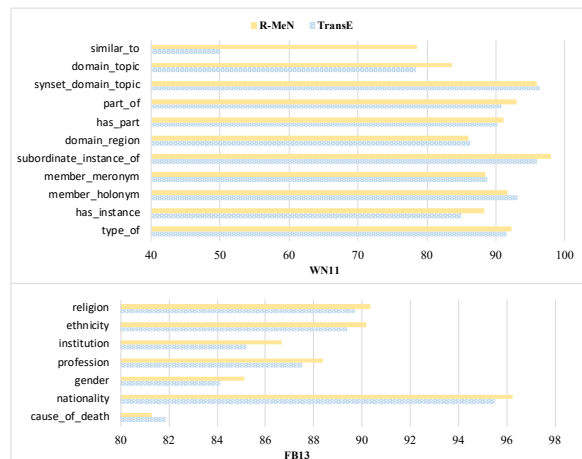


Figure 2: Accuracies for R-MeN and TransE w.r.t each relation on WN11 and FB13.

4.2 Search personalization

Table 3 presents the experimental results on SEARCH17, where R-MeN outperforms up-to-date embedding models and obtains the new highest performances for both MRR and Hits@1 metrics. We restate the prospective strategy proposed by Vu et al. (2017) in utilizing the KG embedding methods to improve the ranking quality of the personalized search systems.

Method	MRR	H@1
SE (Original rank)	0.559	38.5
CI (Teevan et al., 2011)	0.597	41.6
SP (Vu et al., 2015)	0.631	45.2
TransE (Bordes et al., 2013)	0.669	50.9
ConvKB (Nguyen et al., 2019a)	0.750	59.9
CapsE (Nguyen et al., 2019b)	0.766	62.1
Our R-MeN	0.778	63.6

Table 3: Experimental results on the SEARCH17 test set. Hits@1 (H@1) is reported in %. Our improvements over all baselines are statistically significant with $p < 0.05$ using the paired t-test.

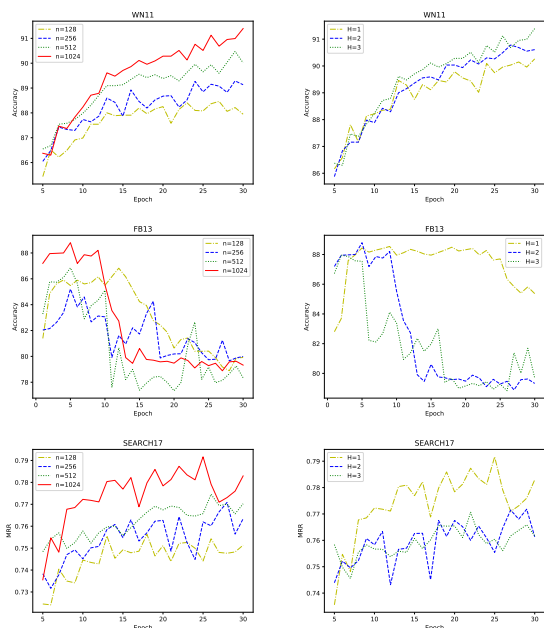


Figure 3: Effects of the head size n and the number H of attention heads on the validation sets.

4.3 Effects of hyper-parameters

Next, we present in Figure 3 the effects of hyper-parameters consisting of the head size n , and the number H of attention heads. Using large head sizes (e.g., $n = 1024$) can produce better performances on all 3 datasets. Additionally, using multiple heads gives better results on WN11 and FB13, while using a single head (i.e., $H = 1$) works best on SEARCH17 because each query usually has a single intention.

4.4 Ablation analysis

For the last experiment, we compute and report our ablation results over 2 factors in Table 4. In particular, the scores degrade on FB13 and SEARCH17 when not using the positional embeddings. More importantly, the results degrade on

Model	WN11	FB13	SEARCH17
Our R-MeN	91.3	88.8	0.792
(a) w/o Pos	91.3	88.7	0.787
(b) w/o M	89.6	88.4	0.771

Table 4: Ablation results on the validation sets. (i) Without using the positional embeddings. (ii) Without using the relational memory network, thus we define $f(s, r, o) = \max(\text{ReLU}([\mathbf{v}_s, \mathbf{v}_r, \mathbf{v}_o] * \Omega))^T \mathbf{w}$.

all 3 datasets without using the relational memory network. These show that using the positional embeddings can explore the relative positions among s , r and o ; besides, using the relational memory network helps to memorize and encode the potential dependencies among relations and entities.

5 Conclusion

We propose a new KG embedding model, named R-MeN, where we integrate transformer self-attention mechanism-based memory interactions with a CNN decoder to capture the potential dependencies in the KG triples effectively. Experimental results show that our proposed R-MeN obtains the new state-of-the-art performances for both the triple classification and search personalization tasks. In future work, we plan to extend R-MeN for multi-hop knowledge graph reasoning. Our code is available at: <https://github.com/daiquocnguyen/R-MeN>.

Acknowledgements

This research was partially supported by the ARC Discovery Projects DP150100031 and DP160103934.

References

- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26*, pages 2787–2795.
- Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning Structured Embeddings of Knowledge Bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 301–306.
- Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 1811–1818.

- Jun Feng, Minlie Huang, Mingdong Wang, Mantong Zhou, Yu Hao, and Xiaoyan Zhu. 2016. Knowledge Graph Embedding by Flexible Translation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference*, pages 557–560.
- Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing Knowledge Graphs in Vector Space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327.
- Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 687–696.
- Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In *Proceedings of the Thirtieth Conference on Artificial Intelligence*, pages 985–991.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.
- Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Learning*, pages 2181–2187.
- Dai Quoc Nguyen, Dat Quoc Nguyen, Tu Dinh Nguyen, and Dinh Phung. 2019a. Convolutional Neural Network-based Model for Knowledge Base Completion and Its Application to Search Personalization. *Semantic Web*, 10(5):947–960.
- Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 327–333.
- Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2019b. A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2180–2189.
- Dat Quoc Nguyen. 2017. An Overview of Embedding Models of Entities and Relationships for Knowledge Base Completion. *arXiv preprint, arXiv:1703.08098*.
- Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Neighborhood Mixture Model for Knowledge Base Completion. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 40–50.
- Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gaborovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 104(1):11–33.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.
- Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. 2018. Relational Recurrent Neural Networks. In *Advances in Neural Information Processing Systems*, pages 7299–7310.
- Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *Advances in Neural Information Processing Systems 26*, pages 926–934.
- Jaime Teevan, Daniel J. Liebling, and Gayathri Ravichandran Geetha. 2011. Understanding and Predicting Personal Navigation. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 85–94.
- Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2071–2080.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *NIPS*, pages 5998–6008.
- Thanh Vu, Dat Quoc Nguyen, Mark Johnson, Dawei Song, and Alistair Willis. 2017. Search Personalization with Embeddings. In *Proceedings of the European Conference on Information Retrieval*, pages 598–604.
- Thanh Vu, Alistair Willis, Son Ngoc Tran, and Dawei Song. 2015. Temporal Latent Topic User Profiles for Search Personalisation. In *Proceedings of the European Conference on Information Retrieval*, pages 605–616.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119.

- Zhigang Wang and Juan-Zi Li. 2016. Text-Enhanced Representation Learning for Knowledge Graph. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1293–1299.
- Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge Base Completion via Search-based Question Answering. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 515–526.
- Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016a. From One Point to A Manifold: Knowledge Graph Embedding for Precise Link Prediction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1315–1321.
- Han Xiao, Minlie Huang, and Xiaoyan Zhu. 2016b. TransG : A Generative Model for Knowledge Graph Embedding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2316–2325.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *Proceedings of the International Conference on Learning Representations*.
- Hee-Geun Yoon, Hyun-Je Song, Seong-Bae Park, and Se-Young Park. 2016. A Translation-Based Knowledge Graph Embedding Preserving Logical Property of Relations. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 907–916.