AprilE: Attention with Pseudo Residual Connection for Knowledge Graph Embedding

Yuzhang Liu[†], **Peng Wang**[†,‡,≀*], **Yingtai Li**[‡], **Yizhan Shao**[‡], **Zhongkai Xu**[≀]
[†]School of Computer Science and Engineering, Southeast University, China
[‡]School of Artificial Intelligence, Southeast University, China
[≀]School of Cyber Science and Engineering, Southeast University, China
{yuzhangliu, pwang, lytai, syzmaxwell, xuzhongkai }@seu.edu.cn

Abstract

Knowledge graph embedding maps entities and relations into low-dimensional vector space. However, it is still challenging for many existing methods to model diverse relational patterns, especially symmetric and antisymmetric relations. To address this issue, we propose a novel model, AprilE, which employs triple-level self-attention and pseudo residual connection to model relational patterns. The triple-level self-attention treats head entity, relation, and tail entity as a sequence and captures the dependency within a triple. At the same time the pseudo residual connection retains primitive semantic features. Furthermore, to deal with symmetric and antisymmetric relations, two schemas of score function are designed via a position-adaptive mechanism. Experimental results on public datasets demonstrate that our model can produce expressive knowledge embedding and significantly outperforms most of the state-of-the-art works.

1 Introduction

Large scale knowledge graphs such as DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), and YAGO (Suchanek et al., 2007), have been shown useful to many applications including natural language understanding (Wang et al., 2017), question answering (Mohammed et al., 2018), and recommender systems (Wang et al., 2019). Knowledge graphs (KGs) (Dong et al., 2014; Hogan et al., 2020) are multi-relational graphs containing much factual information in the form of a triple (h, r, t), where h represents the head entity, t represents the tail entity, and r represents the relationship between h and t. Although the symbolic form of triples effectively represents structured data, it makes KGs hard to calculate semantic similarity between entities and relations. In this paper, we focus specifically on knowledge graph embedding, which maps entities and relations into low-dimensional vector space while preserving the inherent structures of entities and relations.

There are diverse relations in KGs. Figure 1 shows symmetric and antisymmetric relational patterns in a KG. *Barack Obama* and *Michelle Obama* are in a *marriage* relation, and vice versa, therefore, *marriage* is a symmetric relation. For another case, *Barack Obama Sr.* is the *father of Barack Obama* but *Barack Obama* is not the *father_of Barack Obama Sr.*, thus *father_of* is an antisymmetric relation. In addition, there are also different relational categories in Figure 1, which are one-to-one, one-to-many, many-to-one, and many-to-many. For example, both *Barack Obama* and *Michelle Obama* are *doctorate_of J.D.* (Juris Doctor), which belongs to many-to-one relational category. The *alma_mater* of *Barack Obama* are *Columbia University* and *Harvard University*, which belongs to one-to-many relational category.

Many existing models are effective to process different relational categories. TransE (Bordes et al., 2013) interprets the relation as a translation operation on entities in the low-dimensional space, which is straightforward to capture the one-to-one relational category. However, the representations learned by TransE suffer from poor expressiveness of complex relational categories, such as one-to-many, many-to-one, many-to-many. To address this issue, TransH (Wang et al., 2014), TransR (Lin et al., 2015), and TransD (Ji et al., 2015) extend TransE by projecting entities or relations into different vector space, which

*Corresponding author.

Proceedings of the 28th International Conference on Computational Linguistics, pages 508–518
Barcelona, Spain (Online), December 8-13, 2020

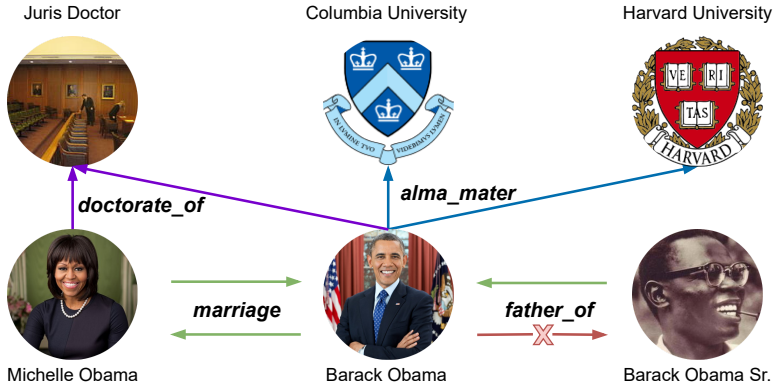

Figure 1: Relational patterns and relational categories in a KG.

achieve progress on processing complex relational categories and are helpful in handling symmetric and antisymmetric relation patterns. However, it is still challenging since these models remain difficulties in capturing richer semantic features due to their inadequate expressiveness.

It is crucial to preserve different relation patterns, especially the symmetry and the antisymmetry (Xu et al., 2020). Several efforts have been made for modelling and inferring symmetric and antisymmetric patterns. DistMult (Yang et al., 2015) exploits a bilinear diagonal model via element-wise product to capture pairwise interaction between head entity, relation and tail entity. However, DistMult can only handle symmetric relations and fail to model antisymmetric relations. ComplEx (Trouillon et al., 2016) represents entities and relations in a complex space thus both symmetric and antisymmetric relations can be learned. However, it causes high memory costs due to the need for high dimensional space.

Moreover, more neural network architectures have been proposed to learn deep expressive features of triples. ConvE (Dettmers et al., 2018) and ConvKB (Nguyen et al., 2018) employ convolutional neural network for KG embedding. CapsE (Nguyen et al., 2019) applies capsule neural network (Sabour et al., 2017) to model the entries at the same dimension in the entity and relation embeddings. Although deep neural networks can capture more expressive features, they are computationally expensive as they usually require pre-trained KG embeddings as input for the neural network.

To address above issues, this paper proposes a novel knowledge graph embedding model named **AprilE** (**A**ttention with **p**seudo **r**esi**d**ua**l** connection **E**mbedding) to model relational patterns, especially symmetric relation and antisymmetric relation. AprilE consists of a triple-level self-attention and a pseudo residual connection. We employ the triple-level self-attention to capture dependency within a triple by treating the factual triple as a sequence. To retain low-level semantic features, we propose a pseudo residual connection by assigning a new embedding vector to each element within the triple as a pseudo identity of original embedding, which is connected to the output of self-attention. Both embeddings construct the final representation of each element. To overcome the limitation of dealing only with symmetrical relations, we propose another schema to process antisymmetric relations, in which the role of each embedding can be switched through position-adaptive mechanism.

Namely, if an entity is the head of a triple, then the second half of its representation will be utilized in the self-attention layer, and the first half will be treated as the pseudo identity. If the entity is the tail of a triple, then the first half of its representation will become the input of the self-attention layer, and the second half will be the pseudo identity. In this way, triple-level self-attention is sensitive to position so that the model can preserve both symmetric and antisymmetric relational patterns. Moreover, two schemas of score function are designed to facilitate the position-adaptive mechanism. We conduct extensive experiments on real-world public datasets. The link prediction results on FB15k, WN18, FB15k-237, and WN18RR show that AprilE outperforms most of the state-of-the-art KG embedding models. Furthermore, the experimental results of each relation on the WN18 dataset indicate that AprilE achieves significant improvement than baselines in symmetric and antisymmetric relations. Moreover, further experiments on FB15k by relational category illuminate AprilE is effective to handle different

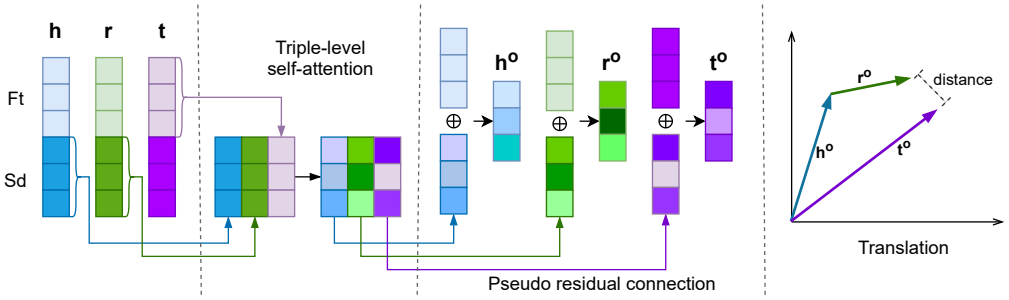

Figure 2: Model architecture of AprilE.

relational categories.

In summary, the main contributions of this paper are three-fold: (1) To learn sufficient semantic features and capture the interdependency of $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t}$ whin a triple, we propose a novel model AprilE that employs the well-designed triple-level attention and pseudo residual connection mechanisms. (2) Two schemas of score function are designed based on the combination of different embedding partitions to deal with symmetric and antisymmetric relations. (3) Extensive experiments on public datasets demonstrate that AprilE can effectively process not only symmetric/antisymmetric relational patterns but also different relation categories, and it significantly outperforms previous state-of-the-art works.

2 Model

2.1 Preliminary

Throughout this paper, given a triple (h, r, t), the lower-case letters h, r, t represents head entities, relations and tail entities, respectively. The corresponding boldface lower-case letters $\mathbf{h}$, $\mathbf{r}$, $\mathbf{t} \in \mathbb{R}^{2d}$, where d is the embedding dimension, are the embeddings of head, relation, and tail, respectively. We employ $\mathbf{E}$, $\mathbf{R}$, $\mathbf{S}$ and $\mathbf{S}'$ to denote the sets of all head and tail entities, all relations, valid triples and invalid triples, respectively.

There are different relational forms in KGs. In terms of relational categories, it refers to the mapping properties of relation including one-to-one, one-to-many, many-to-one, and many-to-many relations (Bordes et al., 2013). As far as relational patterns are concerned, there are symmetric, antisymmetric, inversion, and composition relations (Sun et al., 2019). This paper mainly focuses on symmetric and antisymmetric patterns, which are defined as follows:

Definition 1 (Symmetric pattern). Given a triple (h, r, t), if $\forall h, t,\ r(h,t) \Rightarrow r(t,h)$ holds, then the relation r is symmetric and the triple (h, r, t) is a symmetric pattern.

Definition 2 (Antisymmetric pattern). Given a triple (h, r, t), if $\forall h, t,\ r(h,t) \Rightarrow \neg r(t,h)$ holds, then the relation r is antisymmetric and the triple (h, r, t) is a antisymmetric pattern.

2.2 Overview of AprilE

The principle of AprilE is shown in Figure 2. To model both symmetric and antisymmetric relations, AprilE consists of triple-level self-attention and pseudo residual connection. First, instead of using the whole embedding for triple-level self-attention and residual connection, AprilE divides embeddings of $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$ into two equal-size partitions, $\mathrm{Ft} \in \mathbb{R}^d$ and $\mathrm{Sd} \in \mathbb{R}^d$. Second, AprilE applies the self-attention mechanism to capture the dependency of a triple. Then the pseudo residual connection is used to retain original information. Finally, AprilE introduces a new translation-based score function and a rank-based hinge loss function in model training.

We leverage the positional information of sub-embeddings to make the description concise and clear, for example, $\mathrm{Ft}(\mathbf{r})$ and $\mathrm{Sd}(\mathbf{r})$ represent the first and the second partitions of $\mathbf{r}$. By combining different embedding partitions, it is prone to design different schemas to model relational patterns. Two schemas *symmetric schema* and *antisymmetric schema* are proposed in this paper to deal with symmetric and antisymmetric relational patterns.

Symmetric schema The symmetric schema combines the second partition of the head, relation, and tail embedding for triple-level self-attention. The first partition serves as the pseudo-identity for pseudo residual connection, and is connected to the attention output. Therefore, the learned embedding encode both high-level and low-level semantic features.

Antisymmetric schema Different from the symmetric schema, the second partition of head and relation embedding and the first partition of tail embedding are selected for triple-level self-attention, and the rest embedding partitions act as pseudo-identity for pseudo residual connection. As a result, the learned embedding can express distinct features according to the position of entity, therefore, the model is able to process antisymmetric relational pattern.

Note that the antisymmetric schema also supports to preserve symmetric relational patterns, without the increase of computational complexity. Hence we set the antisymmetric schema as the default schema of AprilE.

2.3 Triple-level self-attention

We apply self-attention mechanism (Vaswani et al., 2017) to capture the dependency of a triple. We first expand the dimension of the embedding partitions to 2-dimension: $\mathrm{Sd}(\mathbf{h}) \in \mathbb{R}^{1\times d}$, $\mathrm{Sd}(\mathbf{r}) \in \mathbb{R}^{1\times d}$, and $\mathrm{Ft}(\mathbf{t}) \in \mathbb{R}^{1\times d}$. Then we concatenate matrices on the first axis to form a $3 \times d$ matrix, we call it union representation:

$$\mathbf{c} = [\mathrm{Sd}(\mathbf{h}); \mathrm{Sd}(\mathbf{r}); \mathrm{Ft}(\mathbf{t})], \tag{1}$$

where $[;]$ stands for the concatenation operation.

Inspired by Vaswani et al. (2017), we serve $\mathbf{c}$ as the input of three components: query, key, and value of self-attention. We first project $\mathbf{c}$ via a non-linear fully-connected layer to produce query $\mathbf{q}$ and key $\mathbf{k}$, respectively:

$$\begin{aligned} \mathbf{q} &= \mathrm{ReLU}(\mathbf{c} \cdot \mathbf{W}^q + \mathbf{b}^q), \\ \mathbf{k} &= \mathrm{ReLU}(\mathbf{c} \cdot \mathbf{W}^k + \mathbf{b}^k), \end{aligned} \tag{2}$$

where $\mathrm{ReLU}(\cdot)$ is an activation function, $\mathbf{W}^q$, $\mathbf{W}^k \in \mathbb{R}^{d\times d}$ are the weights, and $\mathbf{b}^q$, $\mathbf{b}^k \in \mathbb{R}^d$ are the bias terms of query and key, respectively. Next, we calculate element-wise multiplication between the query and the key representation to produce matched product $\mathbf{s}$:

$$\mathbf{s} = \mathbf{q} \odot \mathbf{k}, \tag{3}$$

where $\odot$ stands for element-wise multiplication. After the matching phase, we normalize $\mathbf{s}$ via softmax to get attention weight $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha}_i = \frac{\exp(\mathbf{s}_i)}{\sum_j \exp(\mathbf{s}_j)}. \tag{4}$$

Then we apply attention weight to the union representation $\mathbf{c}$:

$$\mathbf{a} = \sum_i \boldsymbol{\alpha}_i \cdot \mathbf{c}_i. \tag{5}$$

In order to calculate pseudo residual connection of each element, we unpack the weighted union representation $\mathbf{a}$ into three parts to get latent head, relation, and tail representation: $\mathrm{Sd}(\mathbf{h})^\circ$, $\mathrm{Sd}(\mathbf{r})^\circ$ and $\mathrm{Ft}(\mathbf{t})^\circ$.

To this end, we can see that the triple-level self-attention treats head, relation, tail as a whole and seizes the intrinsic dependency to adjust the weights of each element dynamically, which makes the learned representation more flexible and expressive.

2.4 Pseudo residual connection

Residual connection (He et al., 2016) creates shortcuts to combine shallow layers and deep layers, which can smooth the convergence as networks get deeper. Inspired by the residual connection, in our setting, we connect the attention outputs $\mathrm{Sd}(\mathbf{h})^\circ$, $\mathrm{Sd}(\mathbf{r})^\circ$, $\mathrm{Ft}(\mathbf{t})^\circ$ with their pseudo identities $\mathrm{Ft}(\mathbf{h})$, $\mathrm{Ft}(\mathbf{r})$, $\mathrm{Sd}(\mathbf{t})$, respectively, as follows:

$$\begin{aligned} \mathbf{h}^\circ &= \mathrm{Ft}(\mathbf{h}) + \mathrm{Sd}(\mathbf{h})^\circ, \\ \mathbf{r}^\circ &= \mathrm{Ft}(\mathbf{r}) + \mathrm{Sd}(\mathbf{r})^\circ, \\ \mathbf{t}^\circ &= \mathrm{Sd}(\mathbf{t}) + \mathrm{Ft}(\mathbf{t})^\circ. \end{aligned} \tag{6}$$

Apart from creating shortcuts between layers, the pseudo residual connection can also keep original information in the network. The pseudo identity and the attention output come from the same embedding, and are exclusive from each other. Therefore, AprilE can learn attention-transformed representation and, simultaneously, retain primitive embedding representation as much as possible. Integrating transformed representation and primitive embedding representation helps AprilE balance low-level and high-level semantic features and learn sufficient representation.

2.5 Score function and loss function

Similar to TransE (Bordes et al., 2013), we adopt the translation-based score function. For the symmetric schema, the score function is defined as follows:

$$\begin{aligned} f_r(\mathbf{h}, \mathbf{t}) &= \|\mathbf{h}^\circ + \mathbf{r}^\circ - \mathbf{t}^\circ\|_{L_1/L_2} \\ &= \|(\mathrm{Ft}(\mathbf{h}) + \mathrm{Sd}(\mathbf{h})^\circ) + (\mathrm{Ft}(\mathbf{r}) + \mathrm{Sd}(\mathbf{r})^\circ) - (\mathrm{Ft}(\mathbf{t}) + \mathrm{Sd}(\mathbf{t})^\circ)\|_{L_1/L_2}, \end{aligned} \tag{7}$$

where L_1/L_2 stands for L1/L2 norm. The score function above can not deal with antisymmetric relations due to the triple-level self-attention is insensitive to the position because switching positions of head and tail entities do not change respective representations. For antisymmetric schema, the score function is defined as follow:

$$\begin{aligned} f_r(\mathbf{h}, \mathbf{t}) &= \|\mathbf{h}^\circ + \mathbf{r}^\circ - \mathbf{t}^\circ\|_{L_1/L_2} \\ &= \|(\mathrm{Ft}(\mathbf{h}) + \mathrm{Sd}(\mathbf{h})^\circ) + (\mathrm{Ft}(\mathbf{r}) + \mathrm{Sd}(\mathbf{r})^\circ) - (\mathrm{Sd}(\mathbf{t}) + \mathrm{Ft}(\mathbf{t})^\circ)\|_{L_1/L_2}. \end{aligned} \tag{8}$$

For antisymmetric relations, the chosen embedding partitions for triple-level self-attention and pseudo residual connection exchange when changing the position of head and tail entities, which makes the head or tail entity has different representation in different positions. The score is expected to be lower for a valid triple, meanwhile, higher for an invalid triple. To achieve this, we adopt a rank-based hinge loss, which maximizes the discriminative margin between a valid triple $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ and an invalid triple $(\mathbf{h}', \mathbf{r}', \mathbf{t}')$, the loss function is defined as follows:

$$L = \sum_{(\mathbf{h},\mathbf{r},\mathbf{t}) \in \mathbf{S}} \sum_{(\mathbf{h}',\mathbf{r},\mathbf{t}') \in \mathbf{S}'_{(\mathbf{h},\mathbf{r},\mathbf{t})}} \max(0, f_{\mathbf{r}}(\mathbf{h}, \mathbf{t}) + \gamma - f_{\mathbf{r}}(\mathbf{h}', \mathbf{t}')), \tag{9}$$

where γ is the margin, $\mathbf{S}'_{(\mathbf{h},\mathbf{r},\mathbf{t})}$ stands for the set of invalid triples generated by randomly exchanging head or tail entity or both entities in a KG.

3 Experiments

3.1 Datasets, baselines and settings

Datasets We conduct experiments on four benchmark datasets: FB15k, WN18, FB15k-237 and WN18RR, of which the statistics are summarized in Table 1. FB15k (Bordes et al., 2013) is a subset of Freebase (Bollacker et al., 2008), it is related to movies, actors, awards, sports and sport teams

(Dettmers et al., 2018). WN18 (Bordes et al., 2013) is a subset of WordNet (Miller, 1995), in which relationships define lexical relations between words. Considering FB15k and WN18 suffer from test leakage (Toutanova et al., 2015; Dettmers et al., 2018), we also evaluate AprilE on two subsets: FB15k-237 (Toutanova et al., 2015) and WN18RR (Dettmers et al., 2018), in which inverse relations are removed.

Baselines We compare AprilE with some state-of-the-art works, including translation-based models TransE (Bordes et al., 2013) and TransH (Wang et al., 2014), convolution-based model ConvE (Dettmers et al., 2018), bilinear embedding models DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016) and RotatE (Sun et al., 2019) , and graph convolution-based model R-GCN+ (Schlichtkrull et al., 2018).

Table 1: Summary statistics for the datasets.

Dataset	Relations	Entities	Train	Validation	Test
FB15k	$1,345$	$14,951$	$483,142$	$50,000$	59,071
FB15k-237	237	$14,541$	$272,115$	$17,535$	$20,466$
WN18	18	$40,943$	$141,442$	$5,000$	5,000
WN18RR	11	$40,943$	$86,835$	$3,034$	$3,134$

Evaluation task and metrics We evaluate the performance of our method on the task of link prediction. Following (Bordes et al., 2013), for each valid test triple (h, r, t), we replace either h or t by each of all other entities to create a set of corrupted triples. We adopt MR (mean rank) and Hits@10 (the proportion of valid test triples ranking at the top 10 prediction) metrics. All metrics are reported in filtered setting (Bordes et al., 2013), by removing all triples in the graph from the set of corruptions.

Hyperparameters We adopt Adam (Kingma and Ba, 2014) as the optimizer. The hyper-parameters of our model are tuned by grid search and the range of hyperparameters are set as follows: embedding size $d \in \{100, 200, 500\}$, initial learning rate $lr \in \{1e-3, 5e-3, 1e-4, 5e-4\}$, batch size $b \in \{256, 512, 1024\}$, and margin $\gamma \in \{3, 6, 9, 12, 18, 24\}$.

Table 2: Link prediction results on FB15k, WN18, FB15K-237, and WN18RR.

Model	FB15k		WN18		FB15k-237		WN18RR	
	MR	Hits@10	MR	Hits@10	MR	Hits@10	MR	Hits@10
TransE	125	.471	251	.892	–	–	–	–
TransH	87	.644	388	.823	–	–	–	–
DistMult†	97	.824	902	.936	254	.419	5110	.490
ComplEx†	–	.840	–	.947	339	.428	5261	.510
ConvE	64	.873	504	.955	246	.491	5277	.480
RotatE	**40**	.884	309	.959	177	.533	3340	**.571**
R-GCN+	–	.842	–	**.964**	–	.417	–	–
AprilE(w/o PR.)	43	.880	257	.952	181	.521	3750	.532
AprilE(sym.)	41	.887	247	.955	178	.526	3239	.542
AprilE	**40**	**.889**	**244**	.959	**165**	**.535**	**3104**	.553

†: Results are obtained from (Dettmers et al., 2018). The rest of the baselines are retrieved from original papers respectively. Results of AprilE are the average of random five times.

3.2 Experimental results of link prediction

Link prediction aims to predict missing h or t for a relational fact triple (h, r, t), which is a valuable task to evaluate the performance of knowledge graph embedding. The experiment results on four datasets are reported in Table 2. For ablation study purpose, we report the results of three variants of our model, AprilE is the model in antisymmetric (default) schema, AprilE(sym.) is the model in symmetric schema, and AprilE(w/o PR.) is a variant model without pseudo residual connection. We can observe that AprilE

outperforms AprilE(sym.) on all datasets, which illuminates the importance of modelling and inferring more relational patterns.

It is noticed that AprilE performs competitively compared to the state-of-the-art baselines. AprilE achieves the best results on FB15k and its subset FB15k-237 across all metrics. On WN18, AprilE outperforms all baselines on MR, while RGCN+ achieves the best result on Hits@10. On WN18RR in which there are a number of symmetry relations, AprilE achieves the best result on MR while ConvE does not work very well. The reason is that ConvE cannot model symmetric patterns. We also noticed that RotatE achieves the best result on Hits@10 on WN18RR, which indicates that complex space-based embedding models is also powerful to solve symmetric patterns.

Table 3: Hits@10 for models tested on each relation of the WN18.

Relation name	TransE	DistMult	ComplEx	RotatE	AprilE
hypernym	.461	.921	.955	.949	**.961**
hyponym	.479	.916	.945	.952	**.956**
member_meronym	.632	.921	.925	.931	**.947**
member_holonym	.594	.937	.942	.944	**.953**
instance_hypernym	.676	.898	.943	.943	**.951**
instance_hyponym	.634	.889	.954	.958	**.972**
has_part	.642	.927	.936	.945	**.951**
part_of	.706	.936	.936	.939	**.948**
member_of_domain_topic	.491	.811	**.937**	.923	.923
synset_domain_topic_of	.491	.833	.930	.939	**.956**
member_of_domain_usage	.873	.936	.936	.936	**.942**
synset_domain_usage_of	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
member_of_domain_region	.500	.654	.885	.885	**.904**
synset_domain_region_of	.581	.919	.919	.919	**.932**
derivationally_related_form	.466	.943	.955	**.957**	.955
similar_to	.000	**1.00**	**1.00**	**1.00**	**1.00**
verb_group	.679	**.974**	**.974**	**.974**	**.974**
also_see	.143	.625	.643	.652	**.696**
Average	0.558	0.891	0.928	0.930	**0.940**

To compare the performance of different models for symmetric and antisymmetric relations, we also report the results of each relation on WN18, and the results are shown in Table 3. WN18 describes lexical and semantic hierarchies between concepts, which contains antisymmetric relations such as *hypernymy*, *hyponymy*, *part of* (Trouillon et al., 2016), and a some symmetric relations such as *similar_to*. The results indicate that all of the baseline models except TransE can deal with symmetric relations well, especially for *similar_to* relation, and four models achieve 100% performance on Hits@10. Though TransE can deal with antisymmetric relations, it fails to achieve reasonable performance due to the poor expressiveness of representation. DistMult can only process symmetric relations. For the performance on antisymmetric relations, there is still a gap compared with other models. ComplEx, RotatE, and AprilE can deal with symmetric and antisymmetric simultaneously, while AprilE can achieve 0.4% to 1.7% average performance compared with ComplEx and RotaE for antisymmetric relations since AprilE can learn richer semantic expression and capture dependency within a triple.

3.3 Experimental results on FB15k by relational category

Compare to the translation-based models, AprilE significantly outperforms TransE and TransH on FB15k and WN18 datasets. Considering the fact that TransE and TransH are able to process different relational categories, we assume that AprilE can deal with such complex relational categories as well. To prove the assumption, subsequently, we conduct a further experiment to investigate the performance of AprilE on different relational categories: one-to-one, one-to-many, many-to-one, and many-to-many relations.

The dataset of relational categories is built by the approach of Zhen Wang et al. (2014). The results are summarized in Table 4. Compared with baseline models, we can observe that AprilE outperforms in head prediction, and achieves the best results on one-to-many and many-to-one relations in tail prediction. The results illustrate that models producing expressive representations are capable of handling different relational categories. It is also remarkable that both ComplEx and RotatE (Trouillon et al., 2016; Sun et al., 2019) achieve the best results on one-to-one and many-to-many relations in prediction tail, which shows the importance of complex space-based embedding. We leave the work of AprilE on complex space-based embedding as our future work.

Table 4: Experimental results on FB15k by relational category.

Rel. Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
	Head Prediction (Hits@10)				Tail Prediction (Hits@10)			
TransE	.437	.657	.182	.472	.437	.197	.667	.500
TransH	.668	.876	.287	.645	.655	.398	.833	.672
ComplEx†	**.939**	.969	.692	.893	**.938**	.823	.952	.910
RotatE†	.922	.967	.602	.893	.923	.713	.961	**.922**
AprilE(w/o PR.)	.928	.953	.644	.880	.921	.785	.950	.907
AprilE(sym.)	.935	.967	.693	**.894**	.930	.812	.960	.917
AprilE	**.939**	**.971**	**.695**	.891	.934	**.836**	**.962**	.920

†: Results are taken from (Sun et al., 2019). The rest of the baselines are taken from original papers respectively.

4 Discussion

4.1 Effective of triple-level self-attention

In this paper, we propose triple-level self-attention which takes the dependency of a triple into account to produce expressive knowledge graph embedding. Triple-level self-attention is the key component of AprilE. Without triple-level self-attention, AprilE will degenerate into TransE. Similar to TransE, we adopt translation-based score function. Many translation-based models extend TransE by entity or relation projection. Instead, we adopt a novel approach by triple-level self-attention to improve TransE. Extensive experiment results show that the triple-level self-attention enables AprilE to produce more expressive representation, which makes AprilE capable of handling different relational patterns and different relational categories.

4.2 Effective of pseudo residual connection

To explore the effect of pseudo residual connection, we conduct ablation experiments to compare AprilE with and without pseudo residual connection. The contribution of pseudo residual connection is distinctly identified as the model AprilE with pseudo residual connection achieves better results than the model AprilE(w/o PR.) without pseudo residual connection, as shown in Table 2. We conclude that pseudo residual connection is important because it can not only retain useful low-level semantic features but also it enables AprilE to deal with symmetric and antisymmetric relations by designing different score functions. Besides, pseudo residual connection connects pseudo-identity (low-level semantic) and corresponding attention output (high-level semantic), which helps to trade-off different levels of semantic features so as to produce better knowledge graph embedding.

5 Related work

Knowledge graph embedding is a critical research issue of KG and a variety of embedding methods have been proposed. Table 5 summarizes some previous state-of-the-art models as well as AprilE model proposed in this paper in the aspect of scoring function, entity and relation representations, and the ability to model symmetry pattern and antisymmetry pattern.

Translation-based models all follow the translation principle $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. TransE (Bordes et al., 2013) represents relations as translations of the entities from head to tail. Despite its simplicity and efficiency, TransE is overwhelmed when dealing with complex one-to-many, many-to-one and many-to-many relationships (Wang et al., 2014). To overcome the disadvantages of TransE, TransH (Wang et al., 2014), TransR (Lin et al., 2015), TransD (Ji et al., 2015) introduce projection vectors or matrices to map entities embeddings to different relation vector spaces.

Table 5: Summary of several KG embedding models.

Models	Scoring Function $f_r(\mathbf{h}, \mathbf{t})$	Ent.& Rel. embed	Sym.	Antisym.
TransE	$\|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_1/L_2}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$	✗	✓
TransX	$\|\mathbf{h_r} + \mathbf{r} - \mathbf{t_r}\|_{L_1/L_2}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$	✓	✓
DisMult	$\langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$	✓	✗
ComplEx	$Re(\langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle)$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^d$	✓	✓
ConvE	$\sigma(vec(\sigma([\mathbf{M_h}; \mathbf{M_r}] * \omega))\mathbf{W})\mathbf{t}$	$\mathbf{M_h}, \mathbf{M_r} \in \mathbb{R}^{d_w \times d_h}, \mathbf{t} \in \mathbb{R}^d$	✗	✓
RotatE	$\|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{C}^d$	✓	✓
AprilE	$\|\mathbf{h}^\circ + \mathbf{r}^\circ - \mathbf{t}^\circ\|_{L_1/L_2}$	$\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$	✓	✓

$\langle \cdot \rangle$ denotes the generalized dot product, σ denotes activation function, $*$ denotes 2D convolution, $\circ$ of RotatE stands for element-wise product, Sym. stands for Symmetric, and Antisym. stands for Antisymmetric.

Bilinear embedding methods utilize product-based scoring functions to match the potential semantics of entities and relations contained in the vector space representation. RESCAL (Nickel et al., 2011) captures the interaction between the head entity and the tail entity through the relationship-related matrix. DisMult (Yang et al., 2015) simplifies RESCAL for multi-relational representation learning by restricting the relationship-related matrix to a diagonal matrix. Although the number of parameters is greatly reduced, it can not handle the antisymmetric relations in general KGs. To model symmetric and antisymmetric relations, ComplEx (Trouillon et al., 2016) firstly introduces complex space to model triple. RotatE (Sun et al., 2019) takes each relation as a rotation from head entity to tail entity in complex space.

Recently, CNN-based models have been proposed to learn deep expressive features. ConvE (Dettmers et al., 2018) uses 2D convolution over embedding and multiple layers of nonlinear features to model the interactions between entities and relationships, of which the head entity and relation are reshaped into a 2D matrix. ConvKB (Nguyen et al., 2018) adopts CNN to explore the global relationships among same dimensional entries of entities and relational embedding and generalize the transitional characteristics in the transition-based models.

Our proposed model AprilE belongs to the translational embedding methods. More specially, AprilE employs triple-level self-attention and pseudo residual connection, which aims to model the relational patterns including symmetric and antisymmetric relations while it can also model different relational categories including one-to-one, one-to-many, many-to-one and many-to-many.

6 Conclusion and future work

In this paper, we have proposed a novel model AprilE for knowledge graph embedding. The well-designed triple-level self-attention and pseudo residual connection enable AprilE to model symmetric and antisymmetric relations effectively while it can also deal with the complex one-to-many, many-to-one and many-to-many relational categories. Moreover, extensive experiments on public benchmark datasets show that AprilE outperforms most state-of-the-art baselines. In the future, we plan to extend our method on complex space embedding and plan to handle more relational patterns, thereby providing a deeper insight analysis of the proposed model.

Acknowledgements

The work was supported by National Key R&D Program of China (NO.2018YFD1100302) and 13th Five-Year All-Army Common Information System Equipment Pre-Research Project (No.31511110310).

References

Sören Auer, Christian Bizer, Georgi Kobilarov, et al. 2007. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, et al. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 27th ACM SIGMOD International Conference*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, et al. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 27th Neural Information Processing Systems*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, et al. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, et al. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *In Proceedings of the 20th ACM SIGKDD International Conference*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, et al. 2016. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*.

Aidan Hogan, Eva Blomqvist, Michael Cochez, et al. 2020. Knowledge graphs. *ArXiv*, abs/2003.02320.

Guoliang Ji, Shizhu He, Liheng Xu, et al. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: a method for stochastic optimization. *arXiv*, cs.LG/1412.6980.

Yankai Lin, Zhiyuan Liu, Maosong Sun, et al. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 16th North American Chapter of the Association for Computational Linguistics*.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, et al. 2018. A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of the 16th North American Chapter of the Association for Computational Linguistics*.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 17th North American Chapter of the Association for Computational Linguistics*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on Machine Learning*.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, et al. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of the 17th International Semantic Web Conference*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, et al. 2019. Rotate: knowledge graph embedding by relational rotation in complex space. In *Proceedings of the 7th International Conference on Learning Representations*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, et al. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Empirical Methods in Natural Language Processing*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, et al. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. 2017. Attention is all you need. In *Proceedings of the 31st in Neural Information Processing Systems*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, et al. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*.

Jin Wang, Zhongyuan Wang, Dawei Zhang, et al. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*.

Xiang Wang, Xiangnan He, Yixin Cao, et al. 2019. KGAT: knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference*.

Wentao Xu, Shun Zheng, Liang He, et al. 2020. Seek: segmented embedding of knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Bishan Yang, Wen-tau Yih, Xiaodong He, et al. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of 3rd International Conference on Learning Representations*.