Knowledge Graph Embedding With Attentional Triple Context

Huan Gao[1,2], **Jun Shi**[1,2], **Guilin Qi**[3]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, China

Abstract

Knowledge graph embedding can represents entities and relations with efficient low-dimensional embedding vectors. The outstanding performance on knowledge graph completion has led to an increase in the knowledge graph embedding research. State-of-the-art knowledge graph embedding approaches treat each triple independent and neglect structure information. However, as a fact, the rich graph features in knowledge graph can be considered as contexts of a triple which contain large information to describe entities and relations. In this paper, we proposes an Attentional-Triple-Context-based knowledge Embedding model(ATCE), which formulates a local structures around a triple as a context. For each triple, two kinds of structure information are considered as its context, which we refer to *as triple context* : 1) Neighbor context is the outgoing relations and neighboring entities of an entity; 2) Path context is connective relation paths between a pair of entities, both of which contains rich useful and unrelated information for entities and relations. ATCE learns embedding for entities and relations with a attention mechanism and is expected to select the useful information in triple context. The experimental results show that our model outperforms the state-of-the-art methods for link prediction and entity prediction.

1 Introduction

Recent advances in information extraction have led to huge Knowledge graphs(KGs),such as DBpedia,YAGO,Freebase and NELL. These KGs contain facts which represent relations between entities as triples $< h, r, t >$. A triple indicate that entities h and t are connected by relation r. Even a KG contains a very large number of triples, it is still far from complete. The completeness of KGs damage their usefulness in downstream task. Knowledge graph completion or link predictions is thus important approaches for populating existing KGs.

Knowledge graph embedding models for KG completion have attracted much attention, due to their outstanding performance. These embedding model is to represent entitles and relations in a KG into a low dimensional continuous vector space, such vectors contain rich semantic information, and can benefit many downstream tasks especially knowledge graph completion or linked predictions. Whether two entities have a previously unknown relationship can be predicted by simple functions of their corresponding vectors.

Despite the success of previous approaches in KG embedding, most of them mainly model triples individually, ignore lots of information implicitly provided by the structure of the KG. In fact, triples are connected to each other and many triples around a triple could be regarded as a description of it. Recently, Several authors have addressed this issue by incorporating relation path information into model learning and have shown that the relation paths between entities in KGs provide useful information and improve KG completion. These approaches only consider relation information while miss more structure information, such as K-degree neighbors of a given entity, a connected subgraph which n could be exploited for better KB completion. For instance, the whole neighborhood of entities and a connected subgraph between two entities could provide lots of useful information for predicting the relationship between two entities.

In this paper, we present a novel approach to embed a knowledge graph by utilizing the structure information called Attentional-Triple-Context-based knowledge Embedding model(ATCE) which utilizes and chooses the proper context of each triple in the knowledge graph. We define triple context consisting of neighbor context and path context, and define a new score function to evaluate the correlation between a triple and its contexts. Instead of using each triple independently, we incorporate triple context into the score function which is used to evaluate the confidence of a triple. In this way, we make use of a triple context while learning embeddings.

The advantages of our approach are three-fold: 1) We embed a triple by utilizing a local subgraph around a triple instead of a set of independent triples, and extract two kinds of context.

2) Based on the local structure information, we proposed a novel embedding learning approach which named ATCE and a new loss function which convert the score function in TransE to a conditional probability.

3) In order to overcome the noisy data in the context of a triple, an attention mechanism in our approach are proposed

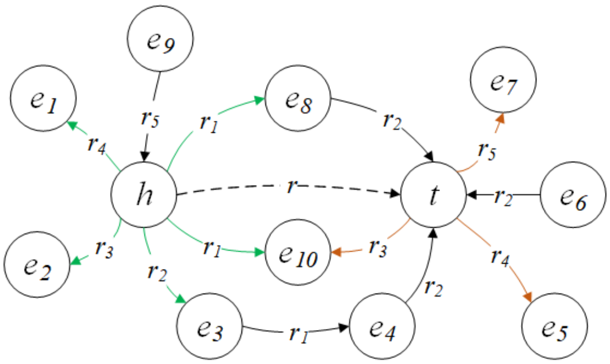

Figure 1: An illustration of the *triple context* of a triple (h, r, t) in a knowledge graph.

to choose the proper information for embedding. In the meanwhile, the attention mechanism can learn the representation power of different neighbor entities and connective path in its context.

Finally, We have conducted preliminary experiment on two benchmark data sets and assessed our method on link prediction task and triple classification. In the experiments we shows chosen context through the attention mechanism to improve the effectiveness of this mechanism. The experimental results show impressive improvements on predictive accuracy compared to other baselines.

2 Triple Context

Firstly, we introduce some notations that are used in this paper. Let $\mathcal{K}$ be a knowledge graph, $\mathcal{E}$ and $\mathcal{R}$ the set of all entities and relations respectively in $\mathcal{K}$. Each triple is denoted as (h, r, t), in which h is the head entity, t is the tail entity and r is the relation between h and t. The embeddings of each entity and relation are denoted in bold, e.g., $\mathbf{h}$ is the embedding of h. All the embeddings are in d-dimensional space $\mathbb{R}^d$. Our goal is to learn embeddings of all entities and relations, which is denoted as Θ. In the following subsections, we define neighbor context and path context, and then give the framework of our model.

2.1 Neighbor Context

Neighbor context of an entity is the surroundings of it in KG. It is the local structure that interacts most with the entity and can reflect various aspects of the entity. Specifically, given an entity e, the neighbor context of e is a set $C_N(e) = \{(r, t) | \forall r, t, (e, r, t) \in \mathcal{K}\}$, where r is an outgoing edge (relation) from e and t is the entity it reaches through r. In other words, the neighbor context of e is all the *relation-tail* pairs appearing in triples with e as the head. For example, as shown in Figure 1, the neighbor context of entity h is $C_N(h) = \{(r_4, e_1), (r_3, e_2), (r_2, e_3), (r_1, e_8), (r_1, e_{10})\}$. We predict the appearance of an entity based on its neighbor context in our model, as a measurement of the compatibility of the entity and its neighbor context.

2.2 Path Context

Path context of a pair of entities is the set of paths that starts from an entity to the other in a KG. It is helpful in modeling the relation and capturing interactions between the pair of entities. Given a pair of entities (h, t), the path context of (h, t) is a set $C_P(h, t) = \{p_i | \forall r_{m_1}, \cdots, r_{m_i}, e_1, \cdots, e_{m_i-1}, p_i = (r_{m_1}, \cdots, r_{m_i}), (h, r_{m_1}, e_1) \in \mathcal{K}, \cdots, (e_{m_i-1}, r_{m_i}, t) \in \mathcal{K}\}$, where $p_i =$ is a list of relations (labeled edges) through which it can traverse from h to t, m_i is the length of path p_i. In Figure 1, the path context between h and t is $C_P(h, t) = \{(r_1, r_2), (r_2, r_1, r_2)\}$. We use the path context to predict the tail entity of a triple given the head entity.

3 Knowledge Graph Embedding With Attentional Triple Context

So far, we have introduced neighbor context and path context, based on which we can define triple context. The triple context of triple (h, r, t) is composed of the neighbor context of the head entity h , the path context of the entity pair (h, t), which can be formalized as:

$$C(h, r, t) = C_N(h) \cup C_P(h, t) \tag{1}$$

The triple context of a triple can be considered to embody the surrounding structures of it in the graph, which makes the model aware of the information contained in graph structures.

We then introduce our approach in detail. In general KG embedding models, the score function of a triple is only related to the embeddings of entities and relations. For example, TransE defines the score function as $f_{TransE}(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{L_1/L_2}$. In our method, triple context is introduced in the score function. Given a candidate triple (h, r, t), the score function is the conditional probability that the triple holds given the triple context and all the embeddings, as follows:

$$f(h, r, t) = P((h, r, t) | C(h, r, t); \Theta) \tag{2}$$

where $C(h, r, t)$ is the triple context of (h, r, t). A higher score of a triple indicates that it holds to a greater extent.

We define an objective function by maximizing the joint probability of all triples in knowledge graph $\mathcal{K}$, which can be formulated as:

$$P(\mathcal{K}|\Theta) = \prod_{(h,r,t) \in \mathcal{K}} f(h, r, t) \tag{3}$$

For the score function in Eq. (2), we use conditional probability formula to decompose the probability $P((h, r, t) | C(h, r, t); \Theta)$ as:

$$\begin{aligned} f(h, r, t) = &P(h | C(h, r, t); \Theta) \\ &\cdot P(t, r | C(h, r, t), h; \Theta) \end{aligned} \tag{4}$$

where the evaluation of the whole triple is decomposed into two parts. The probabilities that h, t and r appear given respective condition are determined in turn in these two parts.

The first part $P(h | C(h, r, t); \Theta)$ in Eq. (4) represents the conditional probability that h is the head entity given the triple context and all the embeddings. Since whether h appears is decided mostly by the neighboring structures of h in the KG, we can approximate $P(h | C(h, r, t); \Theta)$ as

$P(h|C_N(h);\Theta)$, where $C_N(h)$ is the *neighbor context* of h in the KG. The approximated probability $P(h|C_N(h);\Theta)$ can be considered as the compatibility between h and its neighbor context, it is formalized as a softmax-like representation, which is also used in **?** and has been validated, as follows:

$$P(h|C_N(h);\Theta) = \frac{\exp(g_N(h, C_N(h)))}{\sum_{h'\in\mathcal{E}} \exp(g_N(h', C_N(h)))} \quad (5)$$

where $g_P(\cdot,\cdot)$ is the function that describes the correlation between an arbitrary entity h' and entity context of the specific entity h. In reality, different neighbors may have different power of influence to represent h. For example, a target entity is $Terminate2 : JudgementDay$ which is a famous in 1991. The director of this entity or the type of this entity may more important than others. In order to unrelated entities and obtain the correlation, we first obtain a_i which means the represent power between the ith neighbor entity in $C_N(h)$ and an arbitrary triple $< h', r, t >$. Inspired by score function of TransE, we substitute the neighbor entities in $C_N(h)$ for the head h' in triple to compute a_i and then we define a_i

$$a_i = \|t_{n_i} - r_{n_i} + r - t\| \quad (6)$$

Where t_{n_i} is the ith neighbor entity in $C_N(h)$ and r_{n_i} is the relation between t_{n_i} and h'.

Then We use attention model α_i to represent how h' selectively focuses on $C_N(h)$, If the value of α_i is higher, the corresponding neighbor entity in $C_N(h)$ is more important.

$$\alpha_i = \frac{\exp(-a_i)}{\sum_j \exp(-a_j)} \quad (7)$$

In Eq. (8), given a neighbor context $C_N(h)$, the embedding vector of each neighbor has different weights by further considering attention mechanism. Finally, based on the attention results we obtain the correlation between an arbitrary entity h' and neighbor context $C_N(h)$:

$$g_N(h, C_N(h)) = -\sum_i \alpha_i \|\mathbf{h'} + \mathbf{r_{n_i}} - \mathbf{t_{n_i}}\| \quad (8)$$

The second part $P(r,t|C(h,r,t),h;\Theta)$ in Eq. (4) is the conditional probability that t is the tail entity and r is the relation given the head entity h, triple context and all the embeddings. In this part we introduce path context that means t could be related to h through a potential connective path in a knowledge graph. In the second part two kinds of relatedness should be considered that one is the relatedness between h and t in a a potential connective path p_i. And the other is the relatedness between r and p_i. Then We introduce path context among h , t and r to measure the relatedness of them and approximate $P(r,t|C(h,r,t),h;\Theta)$ as $P(t|C_P(h,t),h;\Theta)$, where $C_P(h,t)$ is the *path context* between h and t. The approximated probability $P(r,t|C_P(h,t),\ h;\Theta)$ is formalized as follows:

$$P(r,t|C_P(h,t),h;\Theta) = \frac{\exp(g_P(r,t,C_P(h,t)))}{\sum_{r'\in\mathcal{R},t'\in\mathcal{E}} \exp(g_P(r',t',C_P(h,t)))} \quad (9)$$

where $g_P(\cdot,\cdot)$ is a function of correlation among an arbitrary entity t' an arbitrary relation r and path context of the specific entity pair (h,t). Similar to the neighbor context, given a triple $< h, r, t >$ different pathes in a path context $C_P(h,t)$ have different power of influence the triple. For example when predicting the entity $Englis$, relations like $locate\_in\_Country$ will have less attentions, and the relation $Speak\_Language$ will have greater attention to represent $Engilsh$. In order to obtain $g_P(\cdot,\cdot)$, we firstly calculate the correlation b_i between path p_i in $C_P(h,t)$ and the triple:

$$b_i = \|\mathbf{h} + \mathbf{p_i} - \mathbf{t}\| \quad (10)$$

Where $\mathbf{p_i}$ composes all relations in p_i into a single vector by summing over all their embeddings and this approach is also used in Ptrans. For example, for path $p_i = (r_{m_1}, \cdots, r_{m_i})$, the embedding of it is $\mathbf{p}_i = \mathbf{r}_{m_1} + \cdots + \mathbf{r}_{m_i}$. Eq. (10) has a similar meaning with Eq. (6).

We then choose important pathes from $C_P(h,t)$ through the attention mechanisms. For each p_i, the weight β_i is then defined as

$$\beta_i = \frac{\exp(-b_i)}{\sum_j \exp(-b_j)} \quad (11)$$

If p_i is more closer to h and t, β_i will indicate p_i have greater attentions on translating from h to t.

Finally given an arbitrary entity t' and an arbitrary relation r', the correlation $g_P(\cdot,\cdot)$ between t',r' and $C_P(h,t)$ is:

$$g_P(r',t',C_P(h,t)) = -\sum_i \beta_i(\|\mathbf{h} + \mathbf{p}_i - \mathbf{t'}\| + \|\mathbf{p}_i - \mathbf{r'}\|) \quad (12)$$

In Eq. (12), given a path context $C_P(h,t)$, the embedding vector of each path p_i has different weight β_i. Eq. (12) has two parts, the first part is $\|\mathbf{h} + \mathbf{p}_i - \mathbf{t'}\|$ that indicates the correlation between t' and p_i. Similarly, the second part is $|\mathbf{p}_i - \mathbf{r'}\|$ is the correlation between r' and p_i. If p_i is more related to a triple, the probability of Eq. (9) will be greater.

To utilize these two kinds of context, we combine them by jointly maximizing the probability in E-q. (4) of a triple which is exist in a knowledge graph. $P(h|C(h,r,t);\Theta)$ and $P(t,r|C(h,r,t),h;\Theta)$ can be approximated as $P(h|C_N(h);\Theta)$ and $P(r,t|C_P(h,t),h;\Theta)$, respectively. Thus Eq. (4) Thus, Eq. (4) can be approximated as:

$$f(h,r,t) \approx P(h|C_N(h);\Theta) \cdot P(t,r|C_P(h,t),h;\Theta) \quad (13)$$

in which way the neighbor context and the path context of a triple are incorporated.

3.1 Model Learning

By feasible approximation, the score function is transformed to Eq. (13), each part is represented in softmax form as E-q. (5), Eq. (9) and Eq. (**??**). However, it is impractical to compute these softmax functions directly because of high computational overhead. Hence, we adopt negative sampling, which is proposed in **?** to approximate full softmax function efficiently, to approximate softmax functions in our model.

Taking $P(h|C_N(h);\Theta)$ in Eq.(5) as an example, it is approximated via negative sampling as follows:

$$\begin{aligned}-\log P(\mathcal{K}|\Theta) = -\sum_{(h,r,t)\in\mathcal{K}} &[\log\sigma(g_N(h, C_N(h))) \\ &+ \sum_{h'} \log\sigma(-g_N(h', C_N(h))) \\ &+ \log\sigma(g_P(r,t,C_P(h,t))) \\ &+ \sum_{r',t'} \sigma(-g_P(r',t',C_P(h,t)))]\end{aligned} \tag{14}$$

where $\mathcal{K}' = \{h', r, t\}$ is the corrupted triples by replacing the head entity with an arbitrary entity, n is the number of negative samples and $\sigma(\cdot)$ is the logistic function. $P(t|C_P(h,t),h;\Theta)$ in Eq. (9) and $P(r|h,t;\Theta)$ in Eq. (**??**) are approximated likewise.

In real data sets, the size of neighbor context and path context may be very large, which is computationally expensive for model learning. For this reason, we sample from neighbor context and path context to make triple context tractable. Specifically, we set a threshold n_N for neighbor context and n_P for path context; if the size of the original context exceeds the threshold, we sample a subset, size of which is the threshold, for model learning. Moreover, the length of relation path is constrained to 2 and 3 in our model.

$$\begin{aligned}P(\mathcal{K}|\Theta) &= \prod_{(h,r,t)\in\mathcal{K}} f(h,r,t) \\ &= \prod_{(h,r,t)\in\mathcal{K}} P(h|C(h,r,t);\Theta)\cdot P(r,t|C(h,r,t),h;\Theta) \\ &\approx \prod_{(h,r,t)\in\mathcal{K}} [\sigma(g_N(h,C_N(h)))\cdot \prod_{h'}\sigma(-g_N(h',C_N(h)))] \\ &\cdot [\sigma(g_P(r,t,C_P(h,t)))\cdot\prod_{r',t'}\sigma(-g_P(r',t',C_P(h,t)))]\end{aligned} \tag{15}$$

4 Experiments

4.1 Experimental Setup

Data Set. We use two widely-used benchmark data sets FB15k **?** and FB15k-237 for evaluation, which are extracted from Freebase. FB15k has 592,213 triples with 14,951 entities and 1,345 relationships.FB15k-237 triples are a subset of the FB15K set, that excludes redundant relations and direct training links for held-out triples, with the goal of making the task more realistic **?**.The two datasets are further divided into three parts for model training, tuning and evaluation. Specifically, we use FB15k and FB15k-237 since their triple are rich and closer to the real popular knowledge graph.

Evaluation protocol. Following the same protocol used in **?**, we use *Mean Rank* and *Hits@10* as evaluation protocals of our model. For each test triple (h,r,t), we replace tail head t(or the head h) with each entity e in $\mathcal{E}$ to generate *corrupted triples* and calculate the scores of each triple using the score function. After ranking the scores in descending order, we then get the rank of the correct entity. *Mean Rank* is the mean of all the predicted ranks, and *Hits@10* denotes the proportion of correct entities ranked in the top 10. Note that, a corrupted triple ranking above a test triple could be valid, which should not be counted as an error. To eliminate the effects of such condition, corrupted triples that already exist in the KG are filtered before ranking. In this case, the setting of evaluation is called "Filter", while the original one is called "Raw". To eliminate this effect , the "Filter" setting is more preferred. In both settings, a higher Hits@10 imply the better performance of a model.

Table 3: Link prediction results

Metric	Mean Rank		HITS@10(%)	
	Raw	Filter	Raw	Filter
TransE	243	125	34.9	47.1
TransH (unif)	211	84	42.5	58.5
TransH (bern)	212	87	45.7	64.4
TransR (unif)	226	78	43.8	65.5
TransR (bern)	198	77	48.2	68.7
CTransR (unif)	233	82	44.0	66.3
CTransR (bern)	199	75	48.4	70.2
PTransE	207	58	51.4	**84.6**
GAKE	228	119	44.5	64.8
TCE	**110**	**25**	**55.3**	83.1

Baselines. We use a few outstanding models in recent years as baselines and compare our model with them, including TransE **?**, TransH **?**, TransR **?** , CTransR **?**, PTransE **?** and GAKE **?**.

Implementation. We construct the knowledge graph using Apache TinkerPop[1], an open source graph computing framework. In a few cases, the reverse relation, an edge labeled r^{-1} from t to h for the triple (h,r,t), would be useful when representing some patterns in the graph. For instance, the relation path $a \xrightarrow{motherOf} b \xleftarrow{fatherOf} c$, i.e., $(a, motherOf, b)$ and $(c, fatherOf, b)$, indicates a potential relation $marriedTo$ between a and c. Therefore, we add reverse relation of each relation into KG. Specifically, for each edge labeled r from h to t in the graph, we add another edge labeled r^{-1} from t to h. In addition, the thresholds of neighbor context and path context are both set to 10.

For neighbor context generation, it's expensive to consider all the neighbors of each entity in the graph for the reason that there are some entities connecting with a large amount of other entities, which would lead to a huge size of neighbor context. We use sampling to reduce the size of neighbor context. For those entity whose neighbor context size is larger than a fixed size n, we sample n neighbors randomly from it's neighbor context. Similarly, a large number of paths between a pair of entities would result in high computational complexity. To solve the problem, firstly, we limit the length of paths by 2-step and 3-step, then, we use random walk to sample m paths between a pair of entities. In our experiment, n and m are all set as 10. Note that for some pairs of entities, there may be no 2 or 3 step relation paths. In such case, we suppose that the relatedness between those pairs of entities

[1] `http://tinkerpop.apache.org/`

Table 1: Results on FB15k by relation category

Task	Predicting head(HITS@10(%))				Predicting tail(HITS@10(%))			
Relation 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
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif)	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH (bern)	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR (unif)	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR (bern)	78.8	**89.2**	34.1	69.2	79.2	37.4	**90.4**	72.1
CTransR (unif)	78.6	77.8	36.4	68.0	77.4	37.8	78.0	70.3
CTransR (bern)	**81.5**	89.0	34.7	71.2	**80.8**	38.6	90.1	73.8
TCE	71.0	60.3	**83.9**	**81.9**	70.3	**89.9**	76.0	**89.2**

Table 2: Results on FB15k by relation category

Task	Predicting head(HITS@10(%))				Predicting tail(HITS@10(%))			
Relation 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
TransE	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0
TransH (unif)	66.7	81.7	30.2	57.4	63.7	30.1	83.2	60.8
TransH (bern)	66.8	87.6	28.7	64.5	65.5	39.8	83.3	67.2
TransR (unif)	76.9	77.9	38.1	66.9	76.2	38.4	76.2	69.1
TransR (bern)	78.8	**89.2**	34.1	69.2	79.2	37.4	**90.4**	72.1
CTransR (unif)	78.6	77.8	36.4	68.0	77.4	37.8	78.0	70.3
CTransR (bern)	**81.5**	89.0	34.7	71.2	**80.8**	38.6	90.1	73.8
TCE	71.0	60.3	**83.9**	**81.9**	70.3	**89.9**	76.0	**89.2**

are relatively low and the values of $f_2()$ in Eq. (10) are set as -100.

We use mini-batch SGD to train our model. We choose the learning rate α of SGD among $\{0.1, 0.01, 0.001\}$, the dimension of embeddings k among $\{50, 75, 100\}$, the batch size B among $\{120, 480, 960, 1920, 4800\}$. The best parameters are determined by the performance on valid set. The optimal parameters are $\alpha = 0.001$, $k = 50$ and $B = 4800$.

4.2 Link Prediction

Link prediction **?** is to predict the missing head or tail entity in a given triple based on training triples. Metrics *Mean Rank* and *Hits@10* are used to measure the performance of our model.

We collected the result of link prediction in Table 3. From the results we can see that, our model outperforms other baselines on most of the metrics significantly and consistently, while slightly worse than PTransE on HITS@10. The result implies that triple contexts do improve the performance on link prediction. Although using similar types of contexts in the graph, GAKE's performance is inferior to our model, which shows the superiority of our framework. Note that the experimental results of HOLE are absent here, for it uses a different metric, MRR (Mean Reciprocal rank), instead of *Mean rank* for evaluation. But according to *Hits@10* reported in **?**, the results of our model are better than HOLE.

In Table 2, we show separate evaluation results by category of relationships on FB15k. We can see that ATCE brings promising improvements on modeling complex relations, such as predicting tail of 1-To-N relations, predicting head of N-To-1 relations and N-To-N relations. Specifically, TCE behaves well when predicting the "N" side of 1-To-N and N-To-1 relations, indicating that valid triples have higher scores than invalid triples in general. In some other simpler scenarios, such as 1-To-1 relations and predicting the "1" side of 1-To-N and N-To-1 relations, the performance of ATCE is still acceptable although not so good as some other baselines, such as TransH and TransR. The results suggest that the incorporation of triple context is helpful when handling complex relations, at the cost of precision in modeling simple relations, which seems complementary to some other baselines.

4.3 Triple Classification

We also test our model on triple classification. In this task,given a knowledge base and a triple $< h, r, t >$ we aim to determine whether it is correct or not. FB15K has only positive examples, thus we generated negative triples for FB15K by following strategy of **?**. As the result, the classification accuracies on FB15K can be compare directly with previous studies. In this task, we choose TransE,TransH,TransR and TransD as baseline models. The parameter values for training TransE,TransH,TransR and TransD are borrowed from their reports.

In Table 4, we shows the accuracies of triple classification on two datasets. The context preserving embeddings in general outperform their base model. As the table shows, ATCE always has higher accuracy than TransE,TransR and TransD. One thing to note is that the improvements by the context preserving embeddings are always observed in FB15k, while those in FB15-237 are small or slightly negative. This can be explained with number of triples with a transitive or symmetric relation in the dataset. In FB15k-237, this kinds of triples are removed from FB15k. Thus the improvement in FB15K

Table 4: Triple Classification results

Start	FB15k	FB15k-237
TransE	0.8	0.8
TransE	0.8	0.8
TransE	0.8	0.8
TransE	0.8	0.8
TransE	0.8	0.8
TransE	0.8	0.8
TransE	0.8	0.8
TransE	0.8	0.8

Table 5: attention results in path context

Weight	Path	Relation
0.45	$contains \xrightarrow{contains}$	$partially\_contains$
0.35	$contains \xrightarrow{nationality}$	$marriage\_location$
0.24	$nationality$	$marriage\_location$
0.35	$contains \xrightarrow{place\_lived}$	$marriage\_location$
0.2	$nominated\_for$	$film\_edited\_by$
0.3	$nominated \xrightarrow{award\_nominee}$	$film\_edited\_by$

is remarkable, our approach outperforms others by 11.04% in terms of accuracy on average as the triple context brings more information especially relations between entities when learning the knowledge graph representations.

Furthermore, to better understand the attention mechanism in our approach, The examples of path selected by attention mechanism in path context are shown in Table **??**. The number in front each line is the weight score, which is computed by the attention mechanism for each relation. From the examples we can see that ATCE successfully combines structure learning and parameter learning. It not only choose multiple connective path between two entities to capture the complex structure in the knowledge base, but also learn weight score of the path for a specific relation.

For neighbor context, we also use attention mechanism to choose which neighbor is more related for the head. we demonstrate attentions of the 6 different neighbors when they are regards as the neighbor context of the entity $Terminate2 : JudgementDay$, which indicates a movie in 1991.Fingure shows the results, from the results we see two entities, $Action$ and $Sequel$,have largest attention to represent the target entity $Terminate2 : JudgementDay$, as $Action$ reflects the type of movie while only some of movies have sequels.

5 Related Work

6 Conclusion

In this paper, we proposed TCE, a KG embedding model which is able to take advantages of the triple context in the graph. By defining two kinds of context of a triple and representing them in a unified framework, our model can learn embeddings that are aware of their context. We evaluate our model on link prediction and the experimental results show significant improvements over the major baselines.

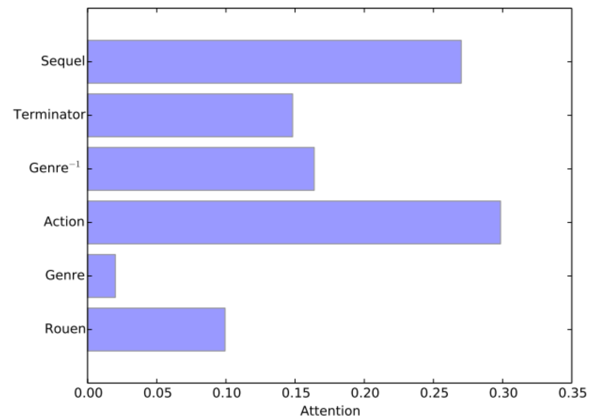

Figure 2: An illustration of the *triple context* of a triple (h, r, t) in a knowledge graph.

In the future, we will research on the following aspects: (1) Conduct experiments on more data sets and tasks to validate our model. (2) Current results show complementarity to some other methods such as TransH, TransR. We would think about a combination of those methods and our model.

References