Time-Aware Representation Learning of Knowledge Graphs

Zikang Wang*†, Linjing Li*†‡, Daniel Dajun Zeng*†‡
*The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
†School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
‡Shenzhen Artificial Intelligence and Data Science Institute (Longhua), Shenzhen, China
{wangzikang2016, linjing.li, dajun.zeng}@ia.ac.cn

***Abstract*—Representation learning is a fundamental task in knowledge graph-related research and applications. Most existing approaches learn representations for entities and relations only based on static facts, where temporal information has been ignored completely. This paper aims to learn time-aware representations for entities and relations in knowledge graphs. Based on how temporal information affects the learned embeddings, we propose three assumptions and build three different models, BTS, ETS, and RTS, respectively. In these models, we build two separate embedding spaces for entities and relations, the standard translation condition is checked after projecting embedding vectors between these spaces by model-specific transformations. As to the performance, the proposed RTS model achieves state-of-the-art results in three experiments conducted on two datasets: YAGO11k and Wikidata12k, which validates the effectiveness of our model. Comparing the results of all three models, we find that relation embeddings are time-sensitive and form natural ordering, while the effects of time on entity embeddings can be safely ignored for translation-based methods. Experiments also show that our findings can be used to simplify other existing models like HyTE.**

I. Introduction

A knowledge graph is a large-scale knowledge base which encodes relational knowledge of entities into a graph structure. In knowledge graphs, a fact is typically stored as a triple (h, r, t), which indicates that entity h and entity t have relation r. There are many famous knowledge graphs such as WordNet [1], Freebase [2], YAGO [3], and Wikidata [4]. They have been widely adopted in various applications, such as recommendation systems [5], question answering [6], reading comprehension [7], to name a few.

A fundamental challenge in research and applications related to knowledge graphs is representation learning. It aims to learn low-dimensional continuous embeddings for the stored entities and relations with semantic knowledge encoded. Various methods have been proposed to achieve this goal, such as TransE [8], TransH [9], and HolE [10].

It is noticeable that most existing methods completely ignore available temporal information in both learning and inference phases, which means facts must be static or time-unaware [11], i.e., triple (h, r, t) cannot include temporal information. However, many facts are valid only in a specific period [12]. For example, the fact "Andre Agassi is a professional tennis player" is formalized as (Andre Agassi, is, professional tennis player) in a knowledge graph, but it only holds from 1986 to 2006 as he started playing tennis as a professional player in 1986 and retired in 2006.

It is evident that temporal information can help learn more credible embeddings for both entities and relations. The literature primarily proposed two kinds of approaches, one of them preserves temporal order in the learned embeddings like t-TransE [11], and the other encodes temporal information explicitly. TA-TransE [13] and HyTE [12] are two algorithms in this direction. TA-TransE first converts temporal information into temporal tokens (i.e., words), then learns the semantics of these tokens using recurrent neural networks. HyTE projects entities and relations onto the same time-dependent hyperplane, then applies the TransE condition [8] to the projected embedding vectors. Like the translation-based representation learning methods for static knowledge graphs, HyTE is simple yet effective, getting state-of-the-art results.

In this paper, we concentrate on encoding temporal information explicitly by extending HyTE to achieve much better results, while preserving the simplicity of it. First, instead of projecting embeddings of entities and relations to time-specific hyperplanes as in HyTE, we transform embeddings between spaces. Second, the effect of temporal information on the embeddings of entities and relations is still not well understood, existing models are based on different assumptions and no rigid analysis is provided. To address this problem, we make three assumptions on whether embeddings of entities or relations are time-sensitive, and build three different models corresponding to them. We compare their experimental results to choose the most effective one.

To be more specific, we incorporate starting and ending times in the fact triple (h, r, t) to encode the temporal information explicitly. Thus, a fact is then denoted as $(h, r, t, \tau_s, \tau_e)$, which indicates entities h and t have relation r from τ_s to τ_e. We make three assumptions: only entity is time-sensitive, only relation is time-sensitive, and both entity and relation are time-sensitive. Three models are built corresponding to these assumptions, referred to as BTS, ETS, and RTS. For all three models, we build two separate embedding spaces for relations and entities, different time-specific transformation matrices between these spaces are learned according to the model assumptions adopted.

We evaluate all the models on three tasks: entity prediction, relation prediction, and temporal scope prediction. The proposed RTS model achieves the best results on all tasks, showing the effectiveness of our models. We further compare and analyze the results of three proposed models and find that the influence of temporal information on relations are more important than on entities for translation-based embedding learning methods. This observation can also be adapted to simplify other algorithms, such as HyTE.

Our contributions are listed as follows:

- We propose three new models to learn time-aware representations of knowledge graphs, among which the RTS model is simple and effective, getting state-of-the-art results on three evaluation tasks.
- We explore how temporal information affects the embeddings of entities and relations. Experiments show that relation embeddings form nature ordering while the influence of time on entities can be ignored for translation-based methods, such as our models and HyTE.
- We show that existing models (e.g., HyTE) can be simplified based on our findings by only considering relations as time-sensitive and get equally good results.

II. Related Work

In this section, we introduce both static and time-aware representation learning approaches for knowledge graphs.

A. Static Representation Learning

Most research on representation learning concentrates on learning low-dimensional continuous embeddings for entities and relations in static knowledge graphs, where no temporal information is provided. Various approaches have been proposed in this field.

One family of the popular methods is the translational method [8], [9], [14], [15]. It measures the plausibility of triples as the distance between two entities aspect to specific relations [16]. Typical translational models include TransE [8], TransH [9], TransD [15] and TransR [14].

Other kinds of methods can achieve competitive results based on the latent semantics of entities and relations, such as ComplEx [17], DistMult [18] and HolE [10].

All these methods can only be used on static knowledge graphs, they cannot deal with temporal information.

B. Time-aware Representation Learning

Representation learning for time-aware knowledge graphs is a new research and application direction.

t-TransE [11] is the first attempt. It learns temporally consistent embeddings by adding a temporal order regular term to the TransE score function, thus maintains the temporal order. However, t-TransE does not encode temporal information into the learned embeddings directly.

Another approach, TA-TransE [13], can learn time-aware embeddings for relations by using RNN. TA-TransE converts timestamps into temporal tokens and other temporal information into modifier tokens. It learns the semantics of these tokens that are used to express time. As a result, TA-TransE cannot compute the temporal scope (i.e., the time interval between two timestamps [12]) with the learned embeddings.

Know-Evolve [19] describes the influence of temporal information using Rayleigh Process and models knowledge evolution using RNN. Know-Evolve has the highest time complexity and model complexity among the existing models.

In order to encode the temporal information explicitly and learn time-aware embeddings simply, HyTE [12] assumes that entities and relations should be projected onto the same time-dependent hyperplane at each timestamp, and the projected vectors must satisfy the TransE condition [8].

HyTE is simple yet effective, but it implicitly assumes that temporal information has the same effects on embeddings of entities and relations. However, this is not a validated assumption. In this paper, we extend HyTE to increase the learning capability and conduct experiments to explore the influence of temporal information on entities and relations.

III. Methodology

This section describes the three models we proposed to learn time-aware representations for knowledge graphs.

A. Problem Definition and Notations

Denote a *temporal knowledge graph* as $\mathcal{G} = \{(h, r, t, \tau_s, \tau_e)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T} \times \mathcal{T}$, where $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{T}$ denotes the set of entities, relations, and timestamps respectively. $h, t \in \mathcal{E}$, $r \in \mathcal{R}$, $\tau_s, \tau_e \in \mathcal{T}$. A tuple $(h, r, t, \tau_s, \tau_e)$ represents a time-aware fact that head entity h and tail entity t have relation r during time scope τ_s to τ_e. Given a time-aware fact $(h, r, t, \tau_s, \tau_e)$, we can equivalently describe it as a set of quadruples $\{(h, r, t, \tau_{k_1}), \cdots, (h, r, t, \tau_{k_n})\}$, if we let (h, r, t, τ_{k_i}) denote head entity h and tail entity t have relation r at the timestamp τ_{k_i}, where $\tau_{k_1}, \tau_{k_2}, \cdots, \tau_{k_n}$ are timestamps included in time scope τ_s to τ_e. In the following, we will adopt quadruple (h, r, t, τ) to formulate our models.

In many representation learning methods, all entities and relations are considered lying in the same space. In this paper, we differentiate the entity space and the relation space to extend the expressive power of the proposed models by allowing them to be affected by different transformation rules.

For each fact (h, r, t), the *time-aware representation learning* aims to learn time-specific embeddings $\boldsymbol{h}_{\tau_k}$ and $\boldsymbol{t}_{\tau_k}$ for entities h and t, and $\boldsymbol{r}_{\tau_k}$ for relation r when given a timestamp $\tau_k \in \mathcal{T}$, where $k = 1, 2, \cdots, N$, $N \in \mathbb{N}$. In general, all entities and relations can evolve over time. Therefore, it is not affordable to track all the entity and relation vectors at each timestamp. To address this challenge, we assume that for each entity and relation, there is a constant embedding vector which is employed to represent it. We denote $\boldsymbol{h}, \boldsymbol{t} \in \mathbb{R}^{k_e}$ as the constant embeddings of entities h and t, $\boldsymbol{r} \in \mathbb{R}^{k_r}$ as the constant embedding of relation r, where k_e is the dimension of the entity space $\mathbb{R}^{k_e}$, and k_r is the dimension of the relation space $\mathbb{R}^{k_r}$. For each timestamp τ_k, to obtain the time-specific embeddings, we learn a time-specific transformation to convert the constant embeddings, $\boldsymbol{h}, \boldsymbol{t}, \boldsymbol{r}$, into the time-specific embeddings, $\boldsymbol{h}_{\tau_k}, \boldsymbol{t}_{\tau_k}, \boldsymbol{r}_{\tau_k}$. In this paper, we adopt the convention that all embeddings are represented by row vectors.

B. Models

According to the above description, we propose a model that considering both entities and relations are time-sensitive. Therefore we call this the BTS (i.e., Both Time-Sensitive) model. There are two simplifications toward the BTS model, one assumes only entities are time-sensitive and the other considers only relations are time-sensitive. We call the former the ETS (i.e., Entity Time-Sensitive) model and the latter the RTS (i.e., Relation Time-Sensitive) model.

1) Model I: BTS: BTS assumes that both entities and relations are affected by time. At a specific timestamp τ_k, we project the entity and relation embeddings into a new space respectively. We refer to it as "time space" in this paper. This model is illustrated in Fig. 1.

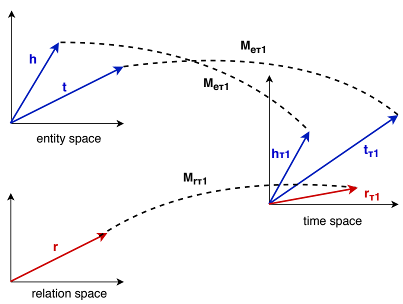


Fig. 1: Illustration of the BTS model. In this model, both relations and entities are time-sensitive, translations are built between projected time-specific relation $\boldsymbol{r}_\tau$ and entity embeddings $\boldsymbol{h}_\tau$ and $\boldsymbol{t}_\tau$ in a separated time space.

At time τ_k, we denote $M_{e\tau_k} \in \mathbb{R}^{k_e \times k_t}$ as the projection matrix from the entity space to the time space, and $M_{r\tau_k} \in \mathbb{R}^{k_r \times k_t}$ as the projection matrix from the relation space to the time space, where $k_t \in \mathbb{R}$ is the dimension of the time space. Then time specific embeddings of entities and relations at timestamp τ_k can be obtained following:

$$\boldsymbol{h}_{\tau_k} = \boldsymbol{h} M_{e\tau_k}, \tag{1}$$
$$\boldsymbol{r}_{\tau_k} = \boldsymbol{r} M_{r\tau_k}, \tag{2}$$
$$\boldsymbol{t}_{\tau_k} = \boldsymbol{t} M_{e\tau_k}. \tag{3}$$

2) Model II: ETS: For ETS, we assume that entities and relations belong to different spaces and only the entity space is affected by time. At a specific timestamp, we project entity embeddings into the relation space based on a time-specific transformation matrix. The illustration is shown in Fig. 2.

We denote $M_{\tau_k} \in \mathbb{R}^{k_e \times k_r}$ as the projection matrix at time τ_k, which projects embeddings from the entity space to the relation space. The time-specific entity embeddings at time τ_k can be computed by the follow project transformation:

$$\boldsymbol{h}_{\tau_k} = \boldsymbol{h} M_{\tau_k}, \tag{4}$$
$$\boldsymbol{t}_{\tau_k} = \boldsymbol{t} M_{\tau_k}. \tag{5}$$

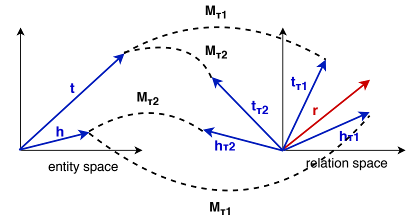


Fig. 2: Illustration of the ETS model. In this model, only entity is time-sensitive, translations are built between the constant relation embeddings $\boldsymbol{r}$ and projected time-specific entity embeddings $\boldsymbol{h}_\tau$ and $\boldsymbol{t}_\tau$.

Since relation embedding is not affected by time, it equals to its corresponding constant embedding,

$$\boldsymbol{r}_{\tau_k} = \boldsymbol{r}. \tag{6}$$

3) Model III: RTS: In the RTS model, we still put entities and relations in two different spaces, while only relation space is affected by time. At a specific time τ_k, we project relation embeddings into the entity space based on time-specific transformation matrix. It is illustrated in Fig. 3.

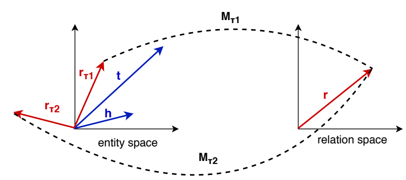


Fig. 3: Illustration of the RTS model. In this model, only relation is time-sensitive, translations are built between projected time-specific relation embeddings $\boldsymbol{r}_\tau$ and the constant entity embeddings $\boldsymbol{h}$ and $\boldsymbol{t}$.

We denote $M_{\tau_k} \in \mathbb{R}^{k_r \times k_e}$ as the projection matrix at time τ_k, which projects embeddings from the relation space to the entity space. Then the relation embedding at time τ_k is

$$\boldsymbol{r}_{\tau_k} = \boldsymbol{r} M_{\tau_k}. \tag{7}$$

Embeddings of entities h and t always stay the same as they are not affected by time,

$$\boldsymbol{h}_{\tau_k} = \boldsymbol{h}, \tag{8}$$
$$\boldsymbol{t}_{\tau_k} = \boldsymbol{t}. \tag{9}$$

C. Model Training

For all three models constructed above, we want the projected embeddings at a specific time τ_k to satisfy the TransE hypothesis, i.e., relation embeddings can be seen as translation vectors from head entities to tail entities

$$\boldsymbol{h}_{\tau_k} + \boldsymbol{r}_{\tau_k} \approx \boldsymbol{t}_{\tau_k}, \tag{10}$$

which has been proved simple and effective in various translational representation learning models.

To achieve this goal, we define a score function $\varphi : \mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{T} \to \mathbb{R}$ for quadruple (h, r, t, τ). The score is calculated based on the learned time-specific embeddings $\boldsymbol{h}_{\tau_k}$, $\boldsymbol{r}_{\tau_k}$, and $\boldsymbol{t}_{\tau_k}$. For a given quadruple, its score is proportional to the likelihood of it being correct. We want the score to be lower for correct quadruples in the given knowledge graph and higher for false quadruples not in the given knowledge graph. In this paper, we apply the same score function for all three models

$$f_{\tau_k}(h,r,t) = \|\boldsymbol{h}_{\tau_k} + \boldsymbol{r}_{\tau_k} - \boldsymbol{t}_{\tau_k}\|. \tag{11}$$

Note that the embeddings $\boldsymbol{h}_{\tau_k}, \boldsymbol{r}_{\tau_k}$, and $\boldsymbol{t}_{\tau_k}$ are normalized before calculating scores.

We train our models by minimizing a margin-based ranking loss over the training set

$$L = \sum_{(h,r,t)\in\mathcal{S}} \sum_{(h',r',t')\in\mathcal{S}'} \max(0, [\gamma + f_{\tau_k}(h,r,t) - f_{\tau_k}(h',r',t')]). \tag{12}$$

where γ is the margin, $\max(0,x)$ returns the positive part of x, $\mathcal{S}$ is the set of correct quadruples in the given knowledge graph, and $\mathcal{S}'$ is the set of false quadruples constructed by negative sampling.

There are two ways to construct $\mathcal{S}'$ in the literature. The widely adopted way is

$$\begin{aligned}\mathcal{S}'_{h,r,t,\tau} =& \{(h',r,t,\tau)|h'\in\mathcal{E}, (h',r,t)\notin\mathcal{S}\} \\ &\cup\{(h,r,t',\tau)|t'\in\mathcal{E}, (h,r,t')\notin\mathcal{S}\}.\end{aligned} \tag{13}$$

It ignores temporal information and selects triples that do not exist in the whole knowledge graph. We use this negative sampling technique in both entity and relation prediction tasks to evaluate the learned embeddings.

In time scope prediction task, we use time dependent negative sampling [12]. For a given quadruple (h,r,t,τ), besides the negative samples obtained by the above method, it also samples negative quadruples that exist in knowledge graphs at times other than τ. These extra negative samples are obtained as following:

$$\begin{aligned}\mathcal{S}'_{h,r,t,\tau} =& \{(h',r,t,\tau)|h'\in\mathcal{E}, \\ &(h',r,t)\in\mathcal{S}, (h',r,t,\tau)\notin\mathcal{S}_\tau\}\cup \\ &\{(h,r,t',\tau)|t'\in\mathcal{E}, \\ &(h,r,t')\in\mathcal{S}, (h,r,t',\tau)\notin\mathcal{S}_\tau\}.\end{aligned} \tag{14}$$

We train our models using gradient descent over training set and choose optimal parameters on validation set.

IV. Experiments and Results

We evaluate the three models on three tasks: entity prediction, relation prediction, and temporal scope prediction. Results are reported in this section.

A. Datasets

We employ two temporal knowledge graph datasets for evaluation: YAGO11k and Wikidata12k, both are proposed to evaluate time-aware representations [12]. The statistics of both datasets are shown in detail in Table I.

- **YAGO11k** is extracted from YAGO3 [3]. It contains 20.5k facts, with start time and end time attached to each entity-relation triple.
- **Wikidata12k** is a subset of Wikidata [4]. it has a much bigger size than YAGO11k, with 40k facts and 12.5k entities.

TABLE I: Statistics of YAGO11k and Wikidata12k.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	#Train	#Valid	#Test
YAGO11k	10,623	10	16,408	2,050	2,051
Wikidata12k	12,554	24	32,497	4,062	4,062

B. Experimental Settings

1) Baselines: We compare the three models against five approaches, including both static representation learning methods and time-aware methods.

- **Static representation learning baselines**
  We use five embedding learning methods for static knowledge graph, including TransE [8], TransH [9], HolE [10], ComplEx [17], and DistMult [18]. Among them TransE and TransH are both simple yet effective translational methods, while others are non-translation based models. For these models, we ignore the temporal information and treat them as (h,r,t) triples.
- **Time-aware representation learning baselines**
  For time-aware models, we use t-TransE [11] and HyTE [12] as baselines. t-TransE models temporal information based on the ordering of relations. HyTE projects entities and relations onto a time-specific hyperplane. HyTE is used as the SOTA model for time-aware baselines.

2) Timestamp Settings: Since there are a massive number of timestamps in a given knowledge graph, uniquely treating each of them will cause additional complexity to the model. We redefine a set of timestamps for the given knowledge graph, and deal with this new set in training and evaluation.

In general, the number of facts per original time interval is not balanced, there may be hundreds of facts in one year, while only few facts in the other ten years. Thus, we define a threshold of the time interval to balance the number of facts in different time intervals. Original intervals with few facts are merged into one larger interval, and the intervals with lots of facts are self-contained. The threshold determining instance numbers per time interval is a hyperparameter, chosen based on valid datasets. This procedure is the same as in HyTE [12].

To simplify the symbols used, we still denote the refined set of timestamps as $\mathcal{T} = \{\tau_1, \tau_2, \cdots, \tau_N\}$.

3) Implementation Details: We set the batch size as 500, set the dimensions of entity, relation, and time space equal to each other. We choose the dimension among $\{64, 128, 256\}$, margin from $\{1, 5, 10\}$. The learning rate is chosen from $\{0.00001, 0.0001, 0.001, 0.01\}$, and the threshold of time interval sample number is chosen from $\{100, 200, 300, 400, 500, 600\}$.

We set the value of parameters based on the experiments on the validation set, and choose parameters for each model

TABLE II: Entity prediction results on datasets YAGO11k and Wikidata12k. The lower mean rank and higher Hits@10 is expected.

Metric	YAGO11k				Wikidata12k			
	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	tail	head	tail	head	tail	head	tail	head
TransE [8]	504	2020	4.4	1.2	520	740	11.0	6.0
TransH [9]	354	1808	5.8	1.5	423	648	23.7	11.8
HolE [10]	1828	1953	29.4	13.7	734	808	25.0	12.3
ComplEx [17]	1543	1853	29.0	15.6	612	703	35.1	27.3
DistMult [18]	748	1757	30.4	14.4	450	563	38.7	29.1
t-TransE [11]	292	1692	6.2	1.3	283	413	24.5	14.5
HyTE [12]	**107**	1069	38.4	16.0	179	237	41.6	25.0
BTS	334	1092	25.5	14.0	321	388	42.8	33.0
ETS	294	1247	21.7	7.4	330	393	44.1	34.5
RTS	146	**722**	**38.6**	**22.3**	**90**	**127**	**50.6**	**39.0**

TABLE III: Relation prediction results on datasets YAGO11k and Wikidata12k. Lower mean rank and higher Hits@1 is expected.

Metric	YAGO11k		Wikidata12k	
	Mean Rank	Hits@1(%)	Mean Rank	Hits@1(%)
TransE [8]	1.7	78.4	1.35	88.4
TransH [9]	1.53	76.1	1.4	88.1
HolE [10]	2.57	69.3	2.23	83.96
t-TransE [11]	1.66	75.5	1.97	74.2
HyTE [12]	1.23	81.2	**1.13**	**92.6**
BTS	1.83	73.3	1.25	85.3
ETS	1.99	69.6	1.31	87.6
RTS	**1.10**	**93.3**	**1.13**	86.7

respectively. Optimal parameters for the three models are: dimension $k_r = k_e = k_t = 128$, margin $\gamma = 10$, and the time interval equals 400 for YAGO11k, and 500 for Wikidata12k. We use optimizer Adam. The optimal learning rate equals 0.0001 while training RTS, and 0.00001 while training ETS and BTS. When using the same learning rate with RTS, ETS and BTS both perform slightly worse. We limit the training procedure in 50 epochs. Note that the results we report in different experiments are obtained with the same parameter setting.

C. Entity Prediction

Aimed at predicting missing entities in triples, entity prediction has been widely adopted since being proposed in [20].

1) Evaluation Protocol: We follow the protocol proposed in [20]. Firstly we construct negative samples by replacing the head entity with every other entity in $\mathcal{E}$, then rank them based on their scores. Ranks of tail entities are get in the same way except replacing tail entities instead of head entities.

We use two metrics: mean rank and Hits@10. Mean rank is the average rank of correct entities. Hits@10 is the proportion of correct entities ranked within top 10. A more capable model requires a lower mean rank and a higher Hits@10.

2) Evaluation Results: Evaluation results on YAGO11k and Wikidata12k are shown in Table II. Model RTS gets the best results with significant improvements. It can encode temporal information into embeddings effectively.

However, ETS and BTS cannot get such outstanding results. Results of ETS and BTS are better than static models in most cases but worse than HyTE. It indicates that although ETS and BTS can use time information, the effect is not as good as model RTS and HyTE. From these experiments, we find that temporal information may have less influence on entities than on relations. We will discuss it further in the next section.

D. Relation Prediction

Relation prediction is similar to entity prediction, aiming at predicting relations between entities.

1) Evaluation Protocol: We construct negative samples by replacing relations with every other relation in relation set $\mathcal{R}$ and rank these triples based on their scores.

We report two metrics for this task: mean rank and Hits@1. Mean rank is the average rank for correct relations, Hits@1 is the proportion of correct relations ranked top 1. In this task, we want a lower mean rank and a higher Hits@1.

2) Evaluation Results: Evaluation results are shown in Table III. Model RTS gets the best results, while the results of ETS and BTS are not good enough. Given that ETS performs the worst among the three models, and it does not consider the effects of time on relation, we can conclude that the effects of temporal information on relation are valuable. This will be discussed further in next section.

E. Temporal Scope Prediction

Proposed in the paper [12], temporal scope prediction is a task aiming to predict in which time interval the fact holds.

1) Evaluation Protocol: We follow the protocol described in [12]. Intervals are set based on the threshold of instance numbers per interval, we use the optimal threshold 300. For all models, YAGO11k and Wikidata12k are treated as 61 and 78 intervals respectively. We replace time interval with every other possible interval and rank them based on their scores. A lower rank is preferred in this task.

As static methods do not take time into account, t-TransE [11] does not explicitly encode temporal information into embeddings. Thus they cannot perform this task. Only HyTE is used as the baseline in this task.

2) Evaluation Results: Temporal scope prediction results are shown in Table IV. RTS outperforms baseline HyTE, while ETS and BTS get slightly worse results than HyTE. It shows that RTS can encode temporal information more efficiently, which is consistent with the results of the first two tasks.

TABLE IV: Temporal scope prediction results. Model RTS outperforms baseline HyTE on both datasets.

Model	YAGO11k	Wikidata12k
HyTE [12]	9.88	17.6
BTS	10.95	18.22
ETS	11.08	18.03
RTS	**9.55**	**12.0**

V. Analysis

We conduct additional experiments in this section to further analyze and explore the influence of temporal information.

A. Influence of Temporal Information

First, we analyze the performance of the three models we proposed. Since link prediction (includes entity prediction and relation prediction) is employed to test the performance of models in most literature, we mainly focus on these two tasks as well. The results in the last section indicate that RTS surpasses almost all baselines, while both ETS and BTS are worse than time-aware baselines. Thus, we form the hypothsis that the effect of time on relation plays a more important role than on entities for translation-based methods.

To further verify this hypothesis, we build two simplified versions of the HyTE model: HyTE-Rel only projects relations to the time-specific hyperplanes, and HyTE-Ent only projects entities to the time-specific hyperplanes. We report the optimal results in Table V. Note that the best results we get are slightly different from those reported in the original HyTE paper. The results show that HyTE-Rel gets competitive results with the original HyTE. HyTE-Rel only projects relation embeddings to time-specific hyperplanes. These results are consistent with the above hypothesis we formed.

This experiment also indicates that we can use this finding to simplify models, which will lead to a simpler model with equally good results. We also find that HyTE-Ent gets similar results with HyTE, however, different from the newly proposed model ETS whose results are worse than the time-aware baselines. It suggests that the impact of temporal information on embeddings is also sensitive to the model.

B. Influence of Time Intervals

The time interval determines how the embeddings are affected by time, it is a key hyperparameter of our models. To investigate how it affects the performance of the models, we set the time interval to $\{100, 200, 300, 400, 500, 600\}$ respectively. The performances of task entity prediction of model RTS with different time intervals are shown in Fig. 4.

We can see that the time interval has a large impact on metric MR. The model performance is worse when the time interval is smaller or larger than the optimal one. The optimal time interval varies for different datasets, which is 400 for YAGO11k and 500 for Wikidata12k. For the metric Hits@10, the effect of time interval is relatively small.

The effect of different time intervals on relation prediction of RTS is shown in Fig. 5. Time interval affects the performance slightly, its influence on relation prediction is smaller compared to task entity prediction.

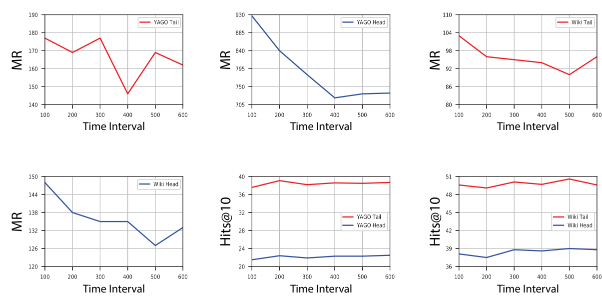


Fig. 4: Influence of time intervals on entity prediction of model RTS. Red lines show the results of entity prediction on tail, while the blue ones show the results on head entities.

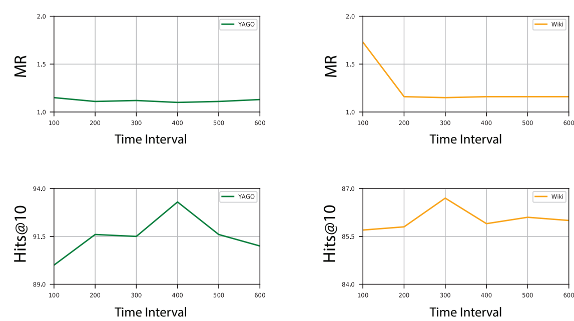


Fig. 5: Influence of time intervals on relation prediction of model RTS. Green lines show the results of dataset YAGO11k, and the orange lines show the results of Wikidata12k.

C. Qualitative Analysis: Embedding Visualization

To investigate how temporal information affects embeddings, we choose to visualize the 2D PCA projection of

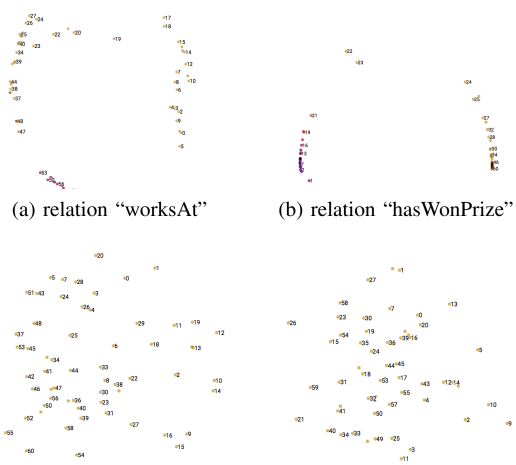
TABLE V: Performance comparison of original HyTE and its simplified versions. It is shown that we can simplify models by restricting only relation as time-sensitive to get equally good results.

Metric	YAGO11k				Wikidata12k			
	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	tail	head	tail	head	tail	head	tail	head
HyTE [12]	**110**	1047	36.2	13.6	**186**	250	41.2	**26.1**
HyTE-Rel	111	**1014**	**37.2**	**14.0**	190	**248**	**41.4**	25.8
HyTE-Ent	111	1067	36.6	13.2	192	250	40.2	25.3

embeddings at different timestamps generated by BTS. Visualization of two randomly chosen relations, "worksAt" and "hasWonPrize", and two randomly chosen entities, "Gatefe_CF", "Netanya", are shown in Fig 6. The numbers beside the embeddings denote the timestamps (the smaller, the earlier). Other relations and entities yield similar results.

For relations, we can find that embeddings form natural ordering, indicating our model can learn time ordering of relations without following explicit ordering constraints like t-TransE [11]. It also shows that relations are affected by temporal information in different ways.

For entities, however, no obvious ordering is learned in embeddings. This result can be treated as another explanation on the time insensitiveness of entity embeddings.



(a) relation "worksAt"

(b) relation "hasWonPrize"

(c) entity "Getafe_CF"

(d) entity "Netanya"

Fig. 6: 2D PCA visualization of embeddings. Numbers beside the embeddings denote the timestamps (the smaller, the earlier). Other relations and entities yield similar results.

We further show the t-SNE [21] visualization of all the relation embeddings at different timestamps for benchmark YAGO11k. Embeddings at different timestamps of the same relation are with the same color. It shows that embeddings of the same relation cluster together. Thus, the effects of temporal information is smaller than semantics for most relations.

D. Temporal Translation on Non-Translation Models

We further explore whether the technique of translating embeddings according to temporal information can also be

Fig. 7: t-SNE visualization of relation embeddings. Embeddings of the same relation at different timestamps are with the same color. Best viewed in color.

applied to the non-translation models, such as HolE, ComplEx, and DistMult. These models are designed based on different motivations compared to translation-based models. Instead of modeling relationships based on translations, HolE employs circular correlation to create compositional representations, DistMult employs a bilinear formulation, and ComplEx makes use of the complex valued embeddings.

In our experiments, we directly apply the temporal translation to HolE, ComplEx, and DistMult. To be specific, we first replace the relation embeddings with the transposed embeddings according to temporal information following RTS, then train each model following their original training process. The results are shown in Table VI.

It shows that for non-translation models, the performance gets worse after directly applied the temporal translation. It indicates the direct combination of temporal translation and non-translation models do not work. Therefore, the non-translation models need to be modified based on their own unique structures to learn time-sensitive representations.

VI. Conclusions and Future Work

This paper focused on time-aware representation learning for knowledge graphs. We made three assumptions about how temporal information affects embeddings of entities and relations, then built three different models corresponding to them. In all of these models, we kept the entities and relations in separate spaces to increase the expressive power of the models. We evaluated these models on three tasks: entity prediction, relation prediction, and temporal scope prediction.

TABLE VI: Transition of time applied on other models.

Metric	YAGO11k				Wikidata12k			
	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	tail	head	tail	head	tail	head	tail	head
HolE [10]	1828	1953	29.7	13.7	734	808	25.0	12.3
HolE+time	2771	1843	1.3	0.2	4270	1483	3.4	0.6
ComplEx [17]	1543	1853	29.0	15.6	612	703	35.1	27.3
ComplEx+time	1687	1785	13.8	2.8	1005	1333	14.3	6.0
DistMult [18]	748	1757	30.4	14.4	450	563	38.7	29.1
DistMult+time	1241	1577	7.6	1.5	1106	1349	6.4	2.5

The proposed RTS model gets state-of-the-art results on all of the tasks. Comparing the results of three models, we found that projecting embeddings into time-specific space is better than just projecting them onto hyperplanes as done by HyTE.

We found that temporal information on relation plays a significant role in representation learning, while entity embeddings can be considered as static in translation-based methods. In addtion, the embeddings of the same relation in different timestamps form nature ordering, while the set of entity embeddings does not exhibit any pattern. Based on this finding, we showed that existing methods like HyTE can be simplified and get competitive results.

For future work, we plan to conduct more theoretical analysis and experiments to explore how to introduce the temporal information to other non-translation models.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2020AAA0103405, the National Natural Science Foundation of China under Grant 71621002, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA27030100. Linjing Li is the corresponding author.

REFERENCES

[1] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 1247–1250.

[3] F. Mahdisoltani, J. Biega, and F. M. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *CIDR*, 2013.

[4] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandeǎić, "Introducing wikidata to the linked data web," in *Proceedings of the 13th International Semantic Web Conference - Part I*, ser. ISWC '14. New York, NY, USA: Springer-Verlag New York, Inc., 2014, pp. 50–65.

[5] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The World Wide Web Conference*, ser. WWW '19, New York, USA, 2019, pp. 3307–3313.

[6] A. Sydorova, N. Poerner, and B. Roth, "Interpretable question answering on knowledge bases and text," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, Jul. 2019, pp. 4943–4951.

[7] A. Yang, Q. Wang, J. Liu, K. Liu, Y. Lyu, H. Wu, Q. She, and S. Li, "Enhancing pre-trained language representations with rich knowledge for machine reading comprehension," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, Jul. 2019, pp. 2346–2357.

[8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Lake Tahoe, USA: Curran Associates, Inc., 2013, pp. 2787–2795.

[9] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, ser. AAAI'14. AAAI Press, 2014, pp. 1112–1119.

[10] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. Phoenix, Arizona: AAAI Press, 2016, pp. 1955–1961.

[11] T. Jiang, T. Liu, T. Ge, L. Sha, S. Li, B. Chang, and Z. Sui, "Encoding temporal information for time-aware link prediction," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016, pp. 2350–2354. [Online]. Available: http://aclweb.org/anthology/D16-1260

[12] S. S. Dasgupta, S. N. Ray, and P. Talukdar, "Hyte: Hyperplane-based temporally aware knowledge graph embedding," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 2001–2011.

[13] A. Garcia-Duran, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 4816–4821. [Online]. Available: http://aclweb.org/anthology/D18-1516

[14] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. Austin, Texas: AAAI Press, 2015, pp. 2181–2187.

[15] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *ACL (1)*. Beijing, China: The Association for Computer Linguistics, 2015, pp. 687–696.

[16] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications." *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, 2017.

[17] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, pp. 2071–2080. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045390.3045609

[18] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[19] R. Trivedi, H. Dai, Y. Wang, and L. Song, "Know-evolve: Deep temporal reasoning for dynamic knowledge graphs," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[20] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, ser. AAAI'11. San Francisco, California: AAAI Press, 2011, pp. 301–306.

[21] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.