



TKGFrame: A Two-Phase Framework for Temporal-Aware Knowledge Graph Completion

Jiasheng Zhang^{1,2}, Yongpan Sheng^{3(✉)}, Zheng Wang^{2,4}, and Jie Shao^{2,5}

¹ Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China
zjss12358@gmail.com

² School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
shaojie@uestc.edu.cn

³ School of Big Data & Software Engineering, Chongqing University, Chongqing 401331, China
shengyp2011@gmail.com

⁴ Institute of Electronic and Information Engineering of UESTC, Dongguan 523808, China
zh.wang@hotmail.com

⁵ Sichuan Artificial Intelligence Research Institute, Yibin 644000, China

Abstract. In this paper, we focus on temporal-aware knowledge graph (TKG) completion, which aims to automatically predict missing links in a TKG by making inferences from the existing temporal facts and the temporal information among the facts. Existing methods conducted on this task mainly focus on modeling temporal ordering of relations contained in the temporal facts to learn the low-dimensional vector space of TKG. However, these models either ignore the evolving strength of temporal ordering relations in the structure of relational chain, or discard more consideration to the revision of candidate prediction results produced by the TKG embeddings. To address these two limitations, we propose a novel two-phase framework called TKGFrame to boost the final performance of the task. Specifically, TKGFrame employs two major models. The first one is a relation evolving enhanced model to enhance evolving strength representations of pairwise relations pertaining to the same relational chain, resulting in more accurate TKG embeddings. The second one is a refinement model to revise the candidate predictions from the embeddings and further improve the performance of predicting missing temporal facts via solving a constrained optimization problem. Experiments conducted on three popular datasets for entity prediction and relation prediction demonstrate that TKGFrame achieves more accurate prediction results as compared to several state-of-the-art baselines.

Keywords: Two-phase framework · Temporal evolution · Temporal-aware knowledge graph completion

1 Introduction

Knowledge graphs (KGs) such as Freebase [2], YAGO [17] and DBpedia [13] have proven to be highly valuable resources for many applications including information extraction [21], semantic search [1] and question answering [6]. However, in fact, KGs with large scales are usually far from complete, because the facts contained in KG are mainly mined from unstructured sources using machine learning and information extraction techniques, and no single source is sufficiently comprehensive as well as technique is sufficiently perfect. Therefore, KG completion¹ is a long-standing but increasingly important task in the research field.

KG embedding, which aims to embed KG elements (i.e., entities and relations) into the latent, low-dimensional, and real-valued vector representations, has been proven to be a powerful technique for KG completion. Over the last few years, several approaches have been developed for this task and two main groups can be distinguished: translational distance-based methods such as TransE [3] and its numerous variants, and semantic matching methods such as RESCAL [16]. Although most of these methods have exhibited both effectiveness and strong generalization capabilities for KG completion, we observed that they only treat KG as a static graph and the assumption in the background is that the facts involved in KG are universally true. Obviously, the assumption is inadequate and inconceivable in many real-world scenarios. In fact, quite of facts in KG are extremely ephemeral or tend to be valid only in a specified time period. An illustration of the comparison between KG and temporal-aware knowledge graph (TKG) based on the topic “*soccer player*” is in Fig. 1. Intuitively, existing approaches may make mistakes without considering the temporal aspects of facts when learning KG embeddings.

In recent years, most of contemporary researchers turn to distributed representations of temporal knowledge graphs (a.k.a TKG embeddings) to deal with the TKG completion problem. It aims to automatically predict missing links in a TKG by making inferences from the existing temporal facts and the temporal information among the facts. A series of representation learning methods for TKG completion, e.g., t-TransE [10], TransE-TAE [9], TTransE [12] and HyTE [5], have been implemented to model the temporal ordering of relations contained in the temporal facts to learn a low-dimensional vector space of TKG. See Sect. 2.2 for more details on TKG embedding methods.

Although existing TKG embedding methods on the above task have achieved preliminary performance improvements, they still suffer from two major limitations: (1) The model ignores the evolving strength of pairwise relations pertaining to the same relational chain, which results in some temporal information loss in the learned embeddings. (2) The model only relies on the learned embeddings to predict the plausibility of missing links in the TKG, and lacks more consideration

¹ KG completion, as known as link prediction in KG, aims to automatically predict missing links between entities based on known facts involved in KG.

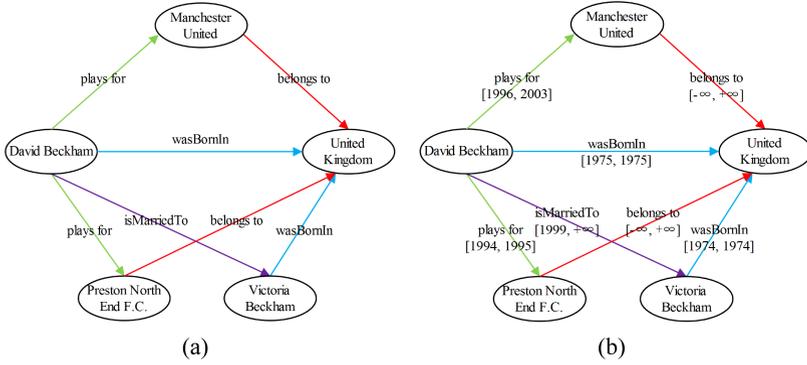


Fig. 1. (a) An example of KG on the topic “soccer player”, in which five entities (e.g., “David Beckham”, “United Kingdom”) are represented as vertices, and four types of relations (e.g., “plays for”, “wasBornIn”) are represented as directed edges with labels in four different colors between vertices; (b) An example of TKG augmented from (a), in which each directed edge as relation with timestamp represented as $[t_{start}, t_{end}]$ denotes its connected fact that starts in t_{start} and ends in t_{end} , e.g., (“David Beckham”, “plays for”, “Manchester United”) is true only during 1996–2003. Note that, as for some facts with special time intervals, if the facts do not end yet, we represent their timestamps as $[t_{start}, +\infty]$, e.g., the fact (“David Beckham”, “isMarriedTo”, “Victoria Beckham”), with the timestamp $[1999, +\infty]$; for some facts are factual ones, we represent their timestamps as $[-\infty, +\infty]$, e.g., the fact (“Manchester”, “belongs to”, “United Kingdom”) with the timestamp $[-\infty, +\infty]$.

to the further revision of prediction results, leading to a suboptimal performance on TKG completion.

To address the above limitations, we propose a novel two-phase framework called TKGFrame to boost the final performance of TKG completion task. TKGFrame addresses these issues by applying two models, namely relation evolving enhanced model and refinement model. Specifically, in the relation evolving enhanced model, based on the work of Jiang *et al.* [9], a refined temporal evolving matrix is introduced to enhance evolving strength representations of pairwise relations in the same relational chain. For example, for the relation-based chain associated with the same person, the temporal ordering relation *wasBornIn* needs to use greater strength to evolve into *diedIn* than into *graduatedFrom* in the temporal-aware embedding space measured by the refined temporal evolving matrix. This kind of temporal ordering enhanced information can be regarded as a regularization term for a joint optimization problem to learn TKG embeddings. The refinement model formulates the plausibility prediction of missing temporal facts in the TKG from the embeddings as an integer linear programming (ILP) problem, in which two types of additional common-sense constraints for temporality are utilized to effectively filter out those implausible predictions for the purpose of improving the prediction quality. In addition, another advantage of the refinement model is that it would benefit to improve the

explainability for the final prediction results by better handling temporal conflicts in relations. By integrating the above two models seamlessly into a complete framework, it can achieve more accurate prediction results. It is worthwhile to highlight our contributions as follows:

- We propose a novel two-phase framework called TKGFrame for TKG completion.
- We present three extensions of TKGFrame based on the idea of incorporating temporal order among relations for learning TKG embeddings [9]: (1) refine a new temporal evolving matrix for better modeling evolving strength representations of pairwise relations pertaining to the same relational chain following the timeline; (2) formulate plausibility measure of the candidate predictions of missing temporal facts as a constrained optimization problem, and propose an ILP approach to solve it as well as avoid implausible predictions from the embedding results; and (3) integrate two models into the proposed TKGFrame seamlessly.
- We conduct extensive experiments on three real-world datasets, newly collected from two popular KG projects, namely YAGO 3 and Wikidata, and compare our results against some state-of-the-art baseline methods on both entity prediction and relation prediction tasks. Experimental results have verified the effectiveness of TKGFrame.
- To illustrate the evaluation of our model and facilitate further research on this topic, we have made the experimental details and source code of the model publicly available².

The remainder of this paper is organized as follows. We review related research in this area in Sect. 2. Section 3 provides the details of each model derived from TKGFrame. In Sect. 4, we conduct extensive experimental evaluations and provide an analysis of the effectiveness of our model in terms of entity prediction and relation prediction, respectively. Finally, the conclusions and future work are described in Sect. 5.

2 Related Work

In this section, we provide an overview of the typical methods for KG embedding learning. These approaches have offered state-of-the-art results for KG and TKG completion on several benchmarks. According to whether the temporal-aware information is considered or not in the learned KG embeddings, the methods can be summarized into two major branches, including static KG embedding methods and temporal-aware KG embedding methods.

² The experimental details and source code of the model are publicly available at <https://github.com/zjs123/TKGComplt>.

2.1 Static Knowledge Graph Embedding Methods

Static KG embedding methods aim to map each entity and each relation in a KG to a latent, low-dimensional, and real-valued vector representation and compute a score to measure the plausibility for each triple by applying a scoring function to these representations. The well-known TransE model [3] maps entities to vectors and regards r as translations from a head entity h to a tail entity l . Based on TransE, a number of improved models have been proposed, such as TransH [20] and TransR [14]. Specifically, TransE [3] attempts to make $h+r$ and l be as close as possible by adjusting the vectors for the head entity h , relation r , and tail entity l . The TransH model [20] models relations as vectors on a relation-specific hyperplane with an associated translation operation. TransE and TransH both embed the entities and relations into the same space. The TransR model [14] considers separate entity and relation spaces to better capture the differences between entities and relations.

2.2 Temporal-Aware Knowledge Graph Embedding Methods

In recent years, an increasing number of researchers have paid attention to this promising area, and many efforts have been made for learning temporal property among relations and relation embedding simultaneously. Jiang *et al.* [10] took the happen time of facts into account, and proposed a TKG embedding model by simply extending standard TransE model. Jiang *et al.* [9] extended the above work and made an attempt to incorporate temporal order information for TKG completion. Specifically, in this approach, instead of incorporating temporal-aware information in the learned embeddings, it first learns temporal order among relations (e.g., *wasBornIn* \rightarrow *worksAt* \rightarrow *diedIn*), and then these relation orders are incorporated as consistency constraints to learn TKG embeddings. However, some explicit temporal relation dependencies in relational chains in the model are not fully considered, which affects the actual quality of the TKG embeddings. In contrast to [9], Dasgupta *et al.* [5] proposed a TKG embedding model called HyTE inspired from the objective of TransH [20], which is able to directly incorporate temporal information in the learned embeddings. Specifically, they firstly divide an input TKG into multiple static subgraphs, each of which is pertinent for a timestamp, and then project all the entities and relations of each subgraph onto the hyperplane specific with a timestamp for joint learning of the hyperplane vectors and the representations of the TKG elements distributed in the subgraphs. TTransE [12] investigated temporal scope prediction over unannotated triples, and extended existing TransE-style scoring functions. TA-TransE [8] utilized digit-level LSTM to learn TKG embeddings combining with existing scoring functions such as TransE and DistMult. For both TTransE and TA-TransE, they verify the effectiveness of the joint learning framework which is based on existing scoring function, with temporal information regularization. In addition, another study in Know-Evolve [19] is mostly focused on factual knowledge evolving. It uses a bilinear model (RESCAL) and employs a deep recurrent neural network in order to learn non-linearly evolving entities.

3 The Proposed Two-Phase Framework

In this section, two models and detailed steps within the proposed TKGFrame are introduced in detail.

3.1 Phase I: Relation Evolving Enhanced Model for TKG Embedding

Our TKG embedding model is expected to better model temporal evolution among facts in the temporal-aware embedding space. For this, Jiang *et al.* [9] firstly propose a key assumption that temporal ordering relations occurring in the facts are associated with each other and evolve in a time dimension. In the guide of this assumption, they attempt to capture the temporal order among relations by using a *temporal evolving matrix* $\mathbf{T} \in \mathbb{R}^{n \times n}$, where n is the dimension of relation embedding, While experimental evidence indicates that \mathbf{T} is indeed helpful for incorporating temporal-aware information to the learned TKG embeddings.

Inspired by the above idea, we refine a new *temporal evolving matrix* $\mathbf{T}_e \in \mathbb{R}^{n \times n}$ for better modeling temporal dependencies in a relational chain. The key distinction between these two forms of *temporal evolving matrices* (i.e., \mathbf{T} and \mathbf{T}_e) is that in the former, \mathbf{T} only can separate given prior relation and subsequent relation which share the same head entity in the temporal-aware embedding space, whereas in the latter, \mathbf{T}_e is able to enhance evolving strength representations of pairwise relations pertaining to the same relational. Specifically, for the same person, there exists a temporal dependency, denoted as *wasBornIn* \rightarrow *graduatedFrom* \rightarrow *worksAt* \rightarrow *diedIn*. As a result, for the first case, *wasBornIn* can evolve into *graduatedFrom*, *graduatedFrom* can evolve into *worksAt*, and *worksAt* can evolve into *diedIn* in a time dimension, with the same intensity. For the second case, *wasBornIn* can evolve into *graduatedFrom*, *worksAt* and *diedIn* with the strength as once, twice, and three times, which can be measured by \mathbf{T}_e , \mathbf{T}_e^2 , and \mathbf{T}_e^3 . Following the similar process introduced above, *graduatedFrom* can also evolve into *worksAt* and *diedIn* in once and twice strength as presented by \mathbf{T}_e and \mathbf{T}_e^2 , respectively. A simple graphical illustration for this example is shown in Fig. 2. In this way, we enhance the evolvement among temporal ordering relations by exploiting different evolving strength measures. It indicates that the farther distance of the pairwise relations in a relation-based chain, the more evolving strength they need.

As studied in [9], we also formulate TKG embedding as an optimization problem based on a temporal-aware regularization term. Given any two positive training quadruples (e_i, r_i, e_j, t_{r_i}) and (e_i, r_j, e_m, t_{r_j}) , they share the same *head entity* and a temporal ordering relation pair $\langle r_i, r_j \rangle$. If $t_{r_i} < t_{r_j}$, we have a pair of positive temporal ordering relations, denoted as $r^+ = \langle r_i, r_j \rangle$, and corresponding negative relation pair $r^- = \langle r_i, r_j \rangle^{-1}$ by inverse. Our optimization requires that positive temporal ordering relation pairs in each relational chain³ should have lower scores than negative pairs. Therefore, we have a temporal scoring function:

³ The relational chain can be constructed by connecting temporal relations sharing the same head entity ranked by an order of their timestamps.

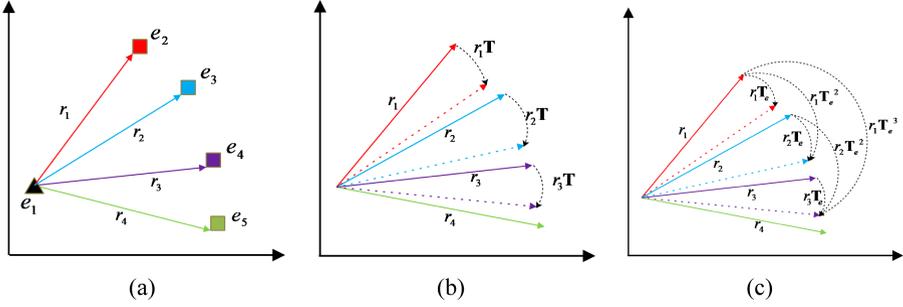


Fig. 2. An illustration of two forms of temporal evolving matrices (i.e., \mathbf{T} and \mathbf{T}_e) in the temporal-aware embedding space. For instance, there exist four temporal ordering relations for the same person, denoted as $r_1 = wasBornIn$, $r_2 = graduatedFrom$, $r_3 = worksAt$, $r_4 = diedIn$, ranked in chronological order in a relation-based chain l_r . The relation between entity e_i and entity e_j corresponds to a translation vector r_i by the TransE model, i.e., $\mathbf{e}_i + \mathbf{r}_i \approx \mathbf{e}_j$ when $(\mathbf{e}_i, \mathbf{r}_i, \mathbf{e}_j)$ holds ($j = i + 1, j \leq 5, i = 1$), as shown in (a). We obtain prior relation’s projection $r_i \mathbf{T}$ near subsequent relation r_{i+1} in the space by projection by \mathbf{T} , i.e., $\mathbf{r}_i \mathbf{T} \approx \mathbf{r}_{i+1}$, but $\mathbf{r}_{i+1} \mathbf{T} \neq \mathbf{r}_i$ ($i = 1, 2, 3$), as shown in (b). Similar to (b), we obtain prior relation’s projection $\mathbf{r}_i \mathbf{T}_e^{j-i}$ near subsequent relation r_j based on their dependency strength in l_r in the space by projection by \mathbf{T}_e , i.e., $\mathbf{r}_i \mathbf{T}_e^{j-i} \approx \mathbf{r}_j$, but $\mathbf{r}_j \mathbf{T}_e^{j-i} \neq \mathbf{r}_i$ ($j = i + 1, j \leq 4, i = 1, 2, 3$), as shown in (c).

$$g(\langle r_i, r_j \rangle) = \|\mathbf{r}_i \mathbf{T}_e^{j-i} - \mathbf{r}_j\|_{l_1/l_2}, \quad (1)$$

where \mathbf{T}_e is a parameter to be learned by our model from the training data, and \mathbf{T}_e^n is to enhance the evolving strengths of pairwise relations with different dependency distances in the relation-based chain, which has introduced in the earlier parts of this section. We are expected to obtain a low score when the temporal ordering relation pair is in chronological order, and a high score otherwise. Note that \mathbf{T}_e is also an asymmetric matrix, resulting in loss function with asymmetric property, so as to a better modeling of temporal ordering relation pairs.

However, in practice, Eq. 1 cannot ensure that the scores of positive temporal ordering relation pairs are absolutely low to fulfill the projection when the chronological order exists, as pointed by Zhou *et al.* [22]. Hence, we follow the strategy adopted in modeling previous work [18] and leverage an optimized objective function as the temporal loss function:

$$\mathcal{O}_r = \sum_{r^+ \in \mathcal{D}_r^+} [g(r^+) - \gamma_1]_+ + \mu \sum_{r^- \in \mathcal{D}_r^-} [\gamma_2 - g(r^-)]_+, \quad (2)$$

where \mathcal{D}_r^+ and \mathcal{D}_r^- denote the sets of positive temporal ordering relation pairs and negative temporal ordering relation pairs covering all the relation-based chains, respectively. $[\cdot]_+ = \max(\cdot, 0)$, $\gamma_1, \gamma_2 > 0$ are two hyperparameters, $\mu > 0$ is used for smoothing as well as to strike a trade-off between the two terms in Eq. 2, and is fixed to 0.5 in our implementation. To summarize, the advantages

of Eq. 2 is two-fold: firstly, we are able to directly control the absolute scores of positive and negative temporal ordering relation pairs as needed, with setting $g(r^+) \leq \gamma_1$ and $g(r^-) \geq \gamma_2$ ($\gamma_2 > \gamma_1$ and γ_1 is a relatively small positive value); secondly, we are still expected to preserve the characteristic of the margin-based ranking criterion deriving from TransE model [3], with setting $g(r^-) - g(r^+) \geq \gamma_2 - \gamma_1$.

Moreover, in order to make the temporal-aware embedding space compatible with the observed temporal facts, we make use of the temporal fact set Δ and follow the same scoring function applied in TransE model:

$$f(\mathbf{e}_s, \mathbf{r}, \mathbf{e}_o) = \|\mathbf{e}_s + \mathbf{r} - \mathbf{e}_o\|_{l_1/l_2}. \quad (3)$$

Combining Eq. 2 and Eq. 3, the final optimization problem can be solved by minimizing the joint scoring function as follows:

$$\mathcal{L} = \sum_{t^+ \in \Delta^+} \sum_{t^- \in \Delta^-} [\gamma + f(t^+) - f(t^-)]_+ + \lambda \Theta_r, \quad (4)$$

where $t^+ \in \Delta^+$ is a positive temporal fact, and $t^- \in \Delta^-$ is a negative temporal fact by randomly replacing head or tail entity of the positive one. In our settings, the constrains are: $\|\mathbf{e}_s\|_2 \leq 1$, $\|\mathbf{r}_i\|_2 \leq 1$, $\|\mathbf{e}_o\|_2 \leq 1$, $\|\mathbf{r}_j\|_2 \leq 1$, $\|\mathbf{r}_i \mathbf{T}_e^n\|_2 \leq 1$, and $\|\mathbf{r}_j \mathbf{T}_e^n\|_2 \leq 1$ for presenting the model from overfitting during training.

The first term in Eq. 4 enforces the resulting temporal-aware embedding space compatible with the whole of observed temporal triples, and the second term further requires the space to be consistent and more accurate. Hyperparameter λ strikes a trade-off between the two terms. We use stochastic gradient descent over shuffled mini-batches to solve this minimization problem.

3.2 Phase II: Refinement Model

After obtaining the TKG embeddings generated from the previous phase, the plausibility prediction of the missing temporal triples from these embeddings inevitably suffers from inferior embedding performance. Hence, in this section, we further model plausibility measure of the candidate predictions as a constrained optimization problem, and propose an integer linear programming (ILP) approach to eliminate implausible predictions from the embedding results. ILP is an optimization model with constraints and the whole of variables are required to be non-negative integers [4].

Objective Function. We first define a decision variable $x_{s,r,o}$ for each candidate quadruple $s_{s,o}^r = (e_s, r, e_o, t_r)$. These variables are binary and indicate whether quadruple $s_{s,o}^r$ is true or false.

$$x_{s,r,o} = \begin{cases} 1, & \text{if } s_{s,o}^r = (e_s, r, e_o, t_r) \text{ is selected to be retained} \\ 0, & \text{otherwise} \end{cases}$$

We then define the objective function as follows:

$$\min \sum_{x_{s,r,o}} (\min f(s_{s,o}^r) - \theta) \times x_{s,r,o}, \quad (5)$$

where $f(s_{s,o}^r)$ represents the plausibility predicted by the prior embedding model, as computed in Eq. 3. The lower score, the more likely to be valid. θ is the threshold used to select temporal facts with sufficiently high possibility (see implementation details in Sect. 4.2 for its value in our experiments).

Constraints. Inspired in part by the considerations given by [9], we then illustrate two categories of common-sense constraints associated temporality for our ILP model: *temporal disjointness constraint* and *temporal ordering constraint*.

(1) Temporal Disjointness Constraint. It claims that the time intervals of any two temporal facts with a same functional relation (i.e., relation type) and a same head entity, or a same tail entity are non-overlapping. For example, a person can only be spouse of one person during a specified time interval, so $(e_1, wasSpouseOf, e_2, [1994, 1998]) \cap (e_1, wasSpouseOf, e_3, [1992, 2000]) \cap (e_2 \neq e_3) \rightarrow \text{false}$. These constraints can be represented as:

$$\begin{aligned} x_{s,r,l} + x_{s,r,m} &\leq 1, x_{n,r,o} + x_{p,r,o} \leq 1, \\ \forall r \in \mathcal{C}_1, t_{x_{s,r,l}} \cap t_{x_{s,r,m}} &\neq \emptyset, t_{x_{n,r,o}} \cap t_{x_{p,r,o}} \neq \emptyset \end{aligned} \quad (6)$$

where \mathcal{C}_1 are functional relations described such as *wasBornIn*, and $t_{x_{s,r,l}}, t_{x_{s,r,m}}$ are time intervals for two temporal facts with a common head entity as well as diverse tail entities, respectively. Similarly, $t_{x_{n,r,o}}, t_{x_{p,r,o}}$ are also time intervals for two temporal facts with a common tail entity as well as diverse head entities, respectively.

(2) Temporal Ordering Constraint. It claims that some temporal relations occur in order. Correspondingly, the fact related to the relation always happens before another one. For example, a person must be born before he graduates, so $(e_1, wasBornIn, e_2, t_1) \cap (e_1, graduatedFrom, e_3, t_2) \cap (t_1 > t_2) \rightarrow \text{false}$. These constraints can be represented as:

$$x_{s,r_i,l} + x_{s,r_j,m} \leq 1, \forall (r_i, r_j) \in \mathcal{C}_2, t_{x_{s,r_i,l}} \leq t_{x_{s,r_j,m}} \quad (7)$$

where $\mathcal{C}_2 = \{(r_i, r_j)\}$ are relation pairs in which each pair has precedent order such as $(wasBornIn, graduatedFrom)$. These relation pairs can be discovered automatically in experimental datasets by statistical strategies and finally manually calibrated.

ILP Model. With the two constraints as described above, we define our final ILP model as follows:

$$\text{min} \quad \sum_{x_{s,r,o}} (\min f(s_{s,o}^r) - \theta) \times x_{s,r,o} \quad (8)$$

$$\begin{aligned} \text{s.t.} \quad \forall r \in \mathcal{C}_1, t_{x_{s,r,l}} \cap t_{x_{s,r,m}} &\neq \emptyset, t_{x_{n,r,o}} \cap t_{x_{p,r,o}} \neq \emptyset \\ x_{s,r,l} + x_{s,r,m} &\leq 1 \\ x_{n,r,o} + x_{p,r,o} &\leq 1, \\ \forall (r_i, r_j) \in \mathcal{C}_2, t_{x_{s,r_i,l}} &\leq t_{x_{s,r_j,m}} \\ x_{s,r_i,l} + x_{s,r_j,m} &\leq 1 \\ x_{s,r,o} &\in \{0, 1\}. \end{aligned} \quad (9)$$

Table 1. Statistics of datasets.

Dataset	#Entity	#Relation	#Triples			Time Interval
			#Train	#Valid	#Test	
YAGO11k	10,623	10	16.4k	2k	2k	[100 - 2017]
Wikidata12k	12,554	24	32.5k	4k	4k	[19 - 2020]
Wikidata11k	11,134	95	112k	14.3k	14.2k	[25 - 2020]

It is obvious that the first constraint restricts the selection of facts with a common head/tail entity and diverse tail/head entity, along with non-overlapping time intervals will be selected. The second constraint restricts that when a pair of facts with precedent ordering relation is selected, but their time intervals do not satisfy this order, they will also be excluded. We minimize this objective function in order to find the best assignment of indicator variables to minimize the overall score of test quadruples while complying with the temporal constraints. We use *PuLP*⁴, which is an LP modeler written in python, to solve the problem.

4 Experiments and Analysis

In this section, we first provide an overview of the datasets used in the experiments, and then conduct an extensive experimental evaluation and provide an analysis of the experimental results in terms of entity prediction task and relation prediction task, respectively.

4.1 Datasets

We evaluate our model and baselines on three datasets, which are derived from two popular KG projects, namely YAGO 3 [15] and Wikidata [7]. We distill out all the facts with timestamps and select those ones pertaining to top-N types of frequent time-sensitive relations in each dataset for our experiments. Simple statistics of the datasets are summarized in Table 1. In the following, we detail each dataset.

- **YAGO11k**: This is a subset of YAGO 3 [15] released by HyTE 2018 [5], containing 10,623 distinct entities, 10 types of most frequent time-sensitive relations, and in a total of 20.4k temporal triples. Here the temporal facts in this dataset are in the form of $(\#factID, OccurSince, t_s), (\#factID, OccurUntil, t_e)$ indicating the fact is valid during $[t_s : t_e]$, where $\#factID$ denotes a specific fact (e_s, r, e_e) .
- **Wikidata12k**: This is a subset of Wikidata released by HyTE 2018 [5], containing 12,554 distinct entities, 24 types of most frequent time-sensitive relations, and in a total of 40.5k temporal triples. This is almost 2 times larger than YAGO11k.

⁴ <https://pypi.python.org/pypi/PuLP>.

- **Wikidata11k**: This is a subset of Wikidata released by TA-TransE [8], containing 11,134 distinct entities, 95 types of time-sensitive relations, and in total of 28.5k temporal triples.

For getting the test and validation set for each dataset, we randomly sample roughly 80% of instances as training, 10% as validation and 10% for testing on each dataset.

4.2 Entity Prediction

Compared Baseline Methods. In order to evaluate our model more comprehensively, a suite of state-of-the-art baselines are compared, including the following:

- **TransE** [3]⁵. This is a simple but effective traditional distance-based model.
- **TransH** [20] (See footnote 5). This model instead models entities as vectors on a relation-specific hyperplane with an associated translation operation for dealing with the complex relations.
- **TransE-TAE** [9]. This model utilizes a temporal ordering of relations to model the knowledge evolution in the time dimension. Observed relation ordering with respect to the same head entity is modeled as a regularization term in conjunction with TransE scoring function. As no code is available, we implemented it by ourselves.
- **TransH-TAE** [9]. This model performs the operations as same as TransE-TAE for learning temporal ordering among relations, and then incorporates them to TransH scoring function as a regularization term. As no code is available, we implemented it by ourselves.
- **TTransE** [12]⁶. This model studies scoring functions that incorporate temporal representation into a TransE-style scoring function, with a focus on the temporal relation prediction task.
- **TA-TransE** [8]⁷. This model utilizes digit-level LSTM to learn TKG embeddings combining with existing scoring functions such as TransE and DistMult.
- **HyTE** [5]⁸. This model incorporates time associated information in the entity-relation spaces by associating each timestamp with a corresponding hyperplane.
- **TKGFrame_{without_ILP}**. This is the variant of our model. We remove the refinement model in the second phase from TKGFrame, which degenerates to only performed the temporal ordering enhanced model in the first phase for TKG completion. We use the subscript `without_ILP` to denote this setting.

⁵ The code for TransE and TransH is from <https://github.com/thunlp/OpenKE>.

⁶ The code for TTransE is from <https://github.com/INK-USC/RE-Net/tree/master/baselines>.

⁷ The code for TA-TransE is from <https://github.com/nle-ml/mmkb>.

⁸ The code for HyTE is from <https://github.com/mallabiisc/HyTE>.

Evaluation Metrics. The entity prediction task aims to predict missing head or tail entity of a triple as introduced in Definition 3.3. In the testing stage, for each quadruple (e_s, r, e_e, t) , we also regard it as a triple without considering its time dimension t , and replace its head/tail entities with all entities in the TKG to construct candidate triples. Then we rank all these entities in descending order of the scores, which are calculated by our scoring function as Eq. 3. Based on the entity ranking list, we adopt two standard metrics from [3]: (1) the mean rank of correct entities (MR), and (2) the proportion of correct entities ranked in top-10 rank entities called Hit@10. As pointed out by Bordes *et al.* [3], the two metrics are desirable but flawed when a corrupted triple exists in the test set. To address this problem, we filter out the whole of corrupted triples that occurred in the test set before ranking. We call the former dataset as *Raw* and the latter one as *Filter* in the evaluation.

Implementation Details. We implement our model and the baselines in PyTorch. All the experiments are performed on an Intel Xeon CPU E5-2640 (v4) with 128 GB main memory, and Nvidia Tesla P100. We initialize all the baselines with the parameter settings in the corresponding papers and then turn them on our datasets for best performance for a fair comparison⁹. For the temporal ordering enhanced model in TKGFrame, we create 100 mini-batches for each epoch during training. The embedding dimension $d \in \{50, 100, 200\}$, margin γ_1 and γ_2 are set in the range of $\{1, 2, 4\}$, learning rate $l \in \{10^{-2}, 10^{-3}, 10^{-4}\}$, negative sampling ratio $n \in \{1, 3, 5\}$, hyperparameter $\lambda \in \{10^{-2}, 10^{-3}, 10^{-4}\}$, and threshold $\theta \in \{10, 11, 12, 13\}$. The best configuration is chosen based on *Raw* MR on the validation dataset. The final parameters are $d = 100$, $\gamma_1 = \gamma_2 = 4$, $l = 10^{-2}$, $n = 3$, $\lambda = 10^{-2}$, $\theta = 11$ and taking l_2 -norm for YAGO11k dataset and $d = 100$, $\gamma_1 = \gamma_2 = 4$, $l = 10^{-2}$, $n = 3$, $\lambda = 10^{-2}$, $\theta = 11$ and taking l_1 -norm for Wikidata12k dataset. For Wikidata11k, the final configuration are $d = 100$, $\gamma_1 = \gamma_2 = 4$, $l = 10^{-3}$, $n = 3$, $\lambda = 10^{-2}$, $\theta = 11$ and l_1 -norm.

Results. Table 2 illustrates the results for entity prediction. We have four major findings. (1) Not surprisingly, temporal-aware embedding models have more obvious advantages than traditional translation-based models such as TransE and TransH on all metrics. This verifies that incorporating temporal information to guide the TKG embedding learning improves the performance of entity predic-

⁹ We train TransE and TransH on all datasets with embedding dimension $d = 100$, margin $\gamma = 1.0$, learning rate $l = 10^{-3}$ and taking l_1 -norm. The configuration of TAE-TransE and TAE-TransH are set as embedding dimension $d = 100$, margin $\gamma_1 = \gamma_2 = 4$, learning rate $l = 10^{-4}$, regularization hyperparameter $t = 10^{-3}$ and taking l_1 -norm for YAGO11k and Wikidata12k datasets, and $d = 100$, $\gamma_1 = \gamma_2 = 2$, $l = 10^{-5}$, $t = 10^{-3}$, taking l_1 -norm for Wikidata11k. We train TA-TransE and TTransE with the same parameter setting as introduced in [11]. For TA-TransE model, the configuration are embedding dimension $d = 100$, margin $\gamma = 1$, batch size $bs = 512$, learning rate $l = 10^{-4}$ and taking l_1 -norm for all the datasets. For HyTE, we initialize the same parameter setting as HyTE, in which embedding dimension $d = 128$, margin $\gamma = 10$, learning rate $l = 10^{-5}$, negative sampling ratio $n = 5$ and using l_1 -norm for all the datasets.

Table 2. The experimental results of different methods on three datasets for entity prediction task. The best and second best baseline results in each column are boldfaced and underlined, respectively (the lower are better for MR, and the higher are better for Hit@10).

Dataset	YAGO11k				Wikidata12k				Wikidata11k			
	MR		Hit@10(%)		MR		Hit@10(%)		MR		Hit@10(%)	
	<i>Raw</i>	<i>Filter</i>	<i>Raw</i>	<i>Filter</i>	<i>Raw</i>	<i>Filter</i>	<i>Raw</i>	<i>Filter</i>	<i>Raw</i>	<i>Filter</i>	<i>Raw</i>	<i>Filter</i>
TransE	1535.7	1522.2	13.7	15.6	1309.8	1296.1	28.1	37.3	200.5	187.5	47.2	72.3
TransH	1431.9	1419.5	14.1	16.2	1063.6	1050.0	30.2	39.9	193.5	180.9	47.3	73.0
TransE-TAE	1118.1	<u>1105.1</u>	16.7	<u>22.5</u>	738.0	723.8	33.0	42.6	55.7	42.3	60.4	80.2
TransH-TAE	1124.9	1111.8	17.1	22.4	488.2	<u>474.1</u>	36.1	<u>48.3</u>	55.1	<u>42.1</u>	61.0	<u>81.8</u>
TTransE	1172.0	1171.5	12.8	13.3	505.6	503.4	<u>36.3</u>	40.4	58.9	56.6	61.5	64.6
TA-TransE	1547.9	1534.9	11.7	13.2	663.0	653.9	34.7	43.7	94.1	85.9	61.2	74.6
HyTE	<u>590</u>	–	<u>18.6</u>	–	<u>237.7</u>	–	32.6	–	<u>36.3</u>	–	73.8	–
TKGFrame _{without_ILP}	671.0	662.3	27.5	29.9	439.9	428.1	38.0	50.9	47.1	34.1	64.7	88.0
TKGFrame	549.6	542.1	29.0	31.3	165.5	153.8	38.1	51.7	30.9	17.6	<u>65.4</u>	88.6

tion. (2) Compared with TransE-TAE, TransH-TAE, and TA-TransE, HyTE obtains better performance on *Raw* MR metrics. This demonstrates its superiority to structure the hyperplanes in the entity-relation space compatible with these temporal facts on the datasets. (3) TKGFrame outperforms all the baselines by a significant improvement. The *Raw* MR drops by nearly 6.8%, 30.3% and 14.8%, and *Filter* Hit@10 rises about 39.1%, 7.0% and 8.3% on YAGO11k, Wikidata12k and Wikidata11k dataset, respectively. This demonstrates that the relation evolving enhanced model is beneficial for generating more accurate TKG embeddings, and the refinement model is useful to remove more implausible predictions. In addition, TKGFrame achieves better performance results than TKGFrame_{without_ILP} on all metrics, illustrating the importance of filtering out implausible predictions from the candidates provided by the relation evolving enhanced model. (4) One interesting observation is that TKGFrame does not outperform HyTE on Wikidata11k with *Raw* Hit@10. One explanation is that because it contains variety of relation types and the distribution of temporal ordering relations especially for the ones in the structure of relational chain is more sparse, this affects the actual quality of the learned TKG embeddings.

4.3 Relation Prediction

Evaluation Metrics. Similar to the entity prediction task, following [3], we adopt two standard metrics for predict missing relation of a quadruple (fact), including MR and Hit@1.

Results. Table 3 shows the results for relation prediction. We have three major findings. (1) TKGFrame outperforms all the baselines by a significant improvement, we verify that the relation evolving enhanced model is valuable to improving the representations of temporal ordered relations in pairwise. (2) The refinement model is indeed able to improve the performance of relation prediction, the

Table 3. The experimental results of different methods on three datasets for relation prediction task. The best and second best baseline results in each column is boldfaced and underlined, respectively (the lower are better for MR, and the higher are better for Hit@1).

Dataset	YAGO11k				Wikidata12k				Wikidata11k			
	MR		Hit@1(%)		MR		Hit@1(%)		MR		Hit@1(%)	
	<i>Raw</i>	<i>Filter</i>										
TransE	1.61	1.60	65.2	66.9	1.26	1.21	85.3	88.2	1.18	1.12	93.6	<u>94.2</u>
TransH	1.57	1.56	67.0	68.1	1.25	1.14	85.5	87.4	1.21	1.15	1.15	92.8
TransE-TAE	1.53	1.51	71.4	72.7	1.23	1.20	85.9	88.9	1.11	<u>1.09</u>	89.6	90.2
TransH-TAE	1.44	<u>1.42</u>	75.5	<u>76.4</u>	1.21	<u>1.11</u>	86.5	<u>89.0</u>	1.23	1.14	84.1	85.4
TTransE	1.47	–	73.8	–	1.22	–	86.0	–	<u>1.08</u>	–	96.3	–
TA-TransE	–	–	–	–	–	–	–	–	–	–	–	–
HyTE	<u>1.38</u>	–	<u>78.4</u>	–	<u>1.12</u>	–	<u>88.8</u>	–	1.10	–	<u>97.2</u>	–
TKGFrame _{without_ILP}	1.29	1.28	77.3	78.2	1.20	1.11	85.0	88.4	1.05	1.03	97.7	98.1
TKGFrame	1.18	1.07	86.5	86.6	1.12	1.06	92.0	92.2	1.04	1.02	97.8	98.3

Raw MR drops by nearly 14.4%, 0% and 3.7%, and the *Filter* Hit@1 rises about 13.3%, 3.5% and 4.3% on YAGO11k, Wikidata12k and Wikidata11k dataset, respectively. This main reason is that two categories of common-sense constraints associated temporality are leveraged to better handle temporal conflicts in relations. Relation prediction can be viewed as a multi-label problem that the same entity pair may have multiple relation labels. For example, (“Einstein”, “ETH Zürich”) could have two valid relations: *graduatedFrom* and *worksAt*. Though using the temporal constraints, we are aware that the two relations have different valid periods, and therefore we would remove the implausible one to improve Hit@1 accuracy.

5 Conclusion

This paper presents a novel two-phase framework, called TKGFrame, to further improve the performance of the TKG completion task. TKGFrame consists of two major models, namely (1) relation evolving enhanced model and (2) refinement model, corresponding to two phases of TKGFrame. To be specific, the first model attempts to enhance evolving strength representations of pairwise relations in the same relational chain by learning a new temporal evolving matrix, resulting in more accurate TKG embeddings. In the second model, we formulate plausibility measure of candidate predictions of unseen temporal facts in the TKG provided by the embeddings from the first model as a constrained optimization problem, and an ILP model is proposed to solve this problem as well as effectively filter out those implausible prediction results presented conflict strictly in each other. The above two models are seamlessly integrated into TKGFrame, which is beneficial to produce more accurate predictions for TKG completion. Comprehensive experiments on three popular datasets show that

the proposed solution outperforms state-of-the-art baselines in terms of entity prediction and relation prediction tasks.

In terms of future work, we attempt to exploit our solution in several kinds of extensions and follow-up studies. One direction is to give greater consideration to enhance the representations of time-sensitive facts by including side information (e.g., type consistency information, accurate textual information) beyond their temporal ordering relations in a relation-based chain that we have considered thus far. A second direction of extension is to a specific consideration regarding the enrichment of temporal ordering relations in pairwise. We will explore using comparing with similar popular facts and timestamp-based label propagation method, to further mine temporal ordering relations, even for some low-resource datasets. A third direction for studying is to further consider using our proposed framework for temporal scope prediction of news facts.

Acknowledgments. This work was supported by Major Scientific and Technological Special Project of Guizhou Province (No. 20183002), Sichuan Science and Technology Program (No. 2020YFS0057, No. 2020YJ0038 and No. 2019YFG0535), Fundamental Research Funds for the Central Universities (No. ZYGX2019Z015) and Dongguan Songshan Lake Introduction Program of Leading Innovative and Entrepreneurial Talents. Yongpan Sheng’s research was supported by the National Key Research and Development Project (No. 2018YFB2101200).

References

1. Barbosa, D., Wang, H., Yu, C.: Shallow information extraction for the knowledge web. In: ICDE, pp. 1264–1267 (2013)
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD, pp. 1247–1250 (2008)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: NIPS, pp. 2787–2795 (2013)
4. Clarke, J., Lapata, M.: Global inference for sentence compression: an integer linear programming approach. *J. Artif. Intell. Res.* **31**, 399–429 (2008)
5. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HYTE: hyperplane-based temporally aware knowledge graph embedding. In: EMNLP, pp. 2001–2011 (2018)
6. Dong, L., Wei, F., Zhou, M., Xu, K.: Question answering over freebase with multi-column convolutional neural networks. In: ACL-IJCNLP (vol. 1: Long Papers), pp. 260–269 (2015)
7. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing wikidata to the linked data web. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 50–65. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_4
8. García-Durán, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion (2018). <https://arxiv.org/abs/1809.03202>
9. Jiang, T., et al.: Towards time-aware knowledge graph completion. In: COLING, pp. 1715–1724 (2016)
10. Jiang, T., et al.: Encoding temporal information for time-aware link prediction. In: EMNLP, pp. 2350–2354 (2016)
11. Jin, W., et al.: Recurrent event network: global structure inference over temporal knowledge graph (2019). <https://arxiv.org/abs/1904.05530>

12. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: WWW, pp. 1771–1776 (2018)
13. Lehmann, J., et al.: DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semant. Web* **6**(2), 167–195 (2015)
14. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI, pp. 2081–287 (2015)
15. Mahdisoltani, F., Biega, J., Suchanek, F.M.: Yago3: a knowledge base from multilingual wikipedias. In: CIDR (2013)
16. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML, vol. 11, pp. 809–816 (2011)
17. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW, pp. 697–706 (2007)
18. Sun, Z., Hu, W., Zhang, Q., Qu, Y.: Bootstrapping entity alignment with knowledge graph embedding. In: IJCAI, pp. 4396–4402 (2018)
19. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In: ICML, vol. 70, pp. 3462–3471 (2017)
20. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI, pp. 1112–1119 (2014)
21. Xiong, C., Callan, J.: Query expansion with freebase. In: ICTIR, pp. 111–120 (2015)
22. Zhou, X., Zhu, Q., Liu, P., Guo, L.: Learning knowledge embeddings by combining limit-based scoring loss. In: CIKM, pp. 1009–1018 (2017)