TEILP: Time Prediction over Knowledge Graphs via Logical Reasoning

Siheng Xiong[1], **Yuan Yang**[1], **Ali Payani**[2], **James C Kerce**[1], **Faramarz Fekri**[1]

[1] Georgia Institute of Technology, 225 North Avenue, Atlanta, Georgia 30332 USA
[2] Cisco Systems Inc, 300 East Tasman Dr, San Jose, California 95134 USA
{sxiong45, yyang754}@gatech.edu, apayani@cisco.com,
clayton.kerce@gtri.gatech.edu, faramarz.fekri@ece.gatech.edu

Abstract

Conventional embedding-based models approach event time prediction in temporal knowledge graphs (TKGs) as a ranking problem. However, they often fall short in capturing essential temporal relationships such as order and distance. In this paper, we propose TEILP, a logical reasoning framework that naturaly integrates such temporal elements into knowledge graph predictions. We first convert TKGs into a temporal event knowledge graph (TEKG) which has a more explicit representation of time in term of nodes of the graph. The TEKG equips us to develop a differentiable random walk approach to time prediction. Finally, we introduce conditional probability density functions, associated with the logical rules involving the query interval, using which we arrive at the time prediction. We compare TEILP with state-of-the-art methods on five benchmark datasets. We show that our model achieves a significant improvement over baselines while providing interpretable explanations. In particular, we consider several scenarios where training samples are limited, event types are imbalanced, and forecasting the time of future events based on only past events is desired. In all these cases, TEILP outperforms state-of-the-art methods in terms of robustness.

Introduction

Temporal knowledge graphs (TKGs) are an important representation when dealing with dynamic and time-dependent relationship between entities. They have various applications such as healthcare and medical research, social event analysis, and recommendation systems. TKGs contain the quadruple (e_s, P, e_o, t) describing the relation P between subject entity e_s and object entity e_o at time t. Due to their large scales, real-world TKGs usually suffer from incompleteness. Thus, link prediction and time prediction, i.e., inferring missing entity and time with existing facts, are the common reasoning tasks for TKGs, proposed in either formal structured query or natural language (Saxena, Chakrabarti, and Talukdar 2021) (Chen, Liao, and Zhao 2023). Compared with link prediction, time prediction is even more challenging as a regression task (Cai et al. 2022).

Existing embedding-based methods consider time prediction as a ranking problem, e.g., HyTE (Dasgupta, Ray, and Talukdar 2018), Time-Aware Embedding (García-Durán, Dumančić, and Niepert 2018), DE-SimplE (Goel et al. 2020) and TNT-ComplEx (Lacroix, Obozinski, and Usunier 2020). The underlying principle is to calculate the score of each timestamp given the triple of subject, object and relation. To predict intervals, TimePlex (Jain et al. 2020) introduces a greedy coalescing strategy, which greedily extends the timestamp with the highest score into an interval. Although these methods give time predictions, they can neither handle unseen timestamps nor utilize the intrinsic connections between timestamps such as temporal order and distance.

Alternatively, inductive logical reasoning methods, e.g., StreamLearner (Omran, Wang, and Wang 2019), TLogic (Liu et al. 2022), and TILP (Xiong et al. 2023), have desirable features when applied to TKGs, as they provide interpretable and robust inference results, and can easily incorporate external background knowledge and domain-specific rules into the reasoning. However, these qualitative predicates based logical rules alone are not enough for time prediction. Thus, several methods built on the temporal point process, e.g., Know-Evolve (Trivedi et al. 2017), GHNN (Han et al. 2020), TLPP (Li et al. 2020) and TELLER (Li et al. 2021), have been introduced to solve the forecasting task, i.e., time prediction from previous events. Compared with the general setting, this task is restricted since all history events are required, and its target is to predict the time gap between the last known event and the next one.

In this paper, we propose TEILP which converts TKGs into a temporal event knowledge graph (TEKG) and enables a differentiable random walk approach to solve the time prediction problem. In TEILP, for each learned rule, a conditional probability density function is associated with the query interval. This density function is then used in a Gaussian mixture model to predict the time. We achieve better performance than embedding methods while providing an additional benefit of human-readable logical explanations. More specifically, our main contributions are:

- We propose TEILP, a temporal logical reasoning framework for time prediction. It is the **first** inductive approach that directly learns temporal logical rules and associated conditional probability density functions for the time prediction.
- We introduce a novel **differentiable** temporal random walk approach by converting TKGs into TEKG where multi-type nodes denote either events or timestamps, and

multi-type edges denote either entities or temporal relations.

- We achieve better performance than state-of-the-art baselines on five benchmark TKG datasets. Our model also demonstrates its robustness in several scenarios where training samples are limited, event types are imbalanced, and forecasting the time of future events based on only past events is desired.

Related Work

Time prediction over knowledge graph has been a challenging task since the timestamps come from a continuous space with intrinsic dependencies such as order and distance. Embedding-based methods, e.g., HyTE (Dasgupta, Ray, and Talukdar 2018), Time-Aware Embedding (García-Durán, Dumančić, and Niepert 2018), DE-SimplE (Goel et al. 2020) and TNT-ComplEx (Lacroix, Obozinski, and Usunier 2020), focus more on link predication, and consider time prediction as a similar ranking problem for different timestamps. HyTE projects the relation and entities on all the temporal hyperplanes, and orders the timestamps according to their plausibility scores. TNT-ComplEx creates time-dependent embeddings for entities and relations, and examines as to how the score of facts changing with time. To handle the interval prediction task, the authors of Time-Plex (Jain et al. 2020) introduces a greedy coalescing strategy, and extends embedding-based models via additional temporal constraints (relation recurrence, ordering and time gap distribution). When predicting time, both embedding-based scores and these constraints are considered to rank the timestamps. Time2box (Cai et al. 2021) considers timestamps as box filters, picking out correct answers for temporal queries. However, embedding-based timestamp ranking methods cannot be very accurate since they ignore the interdependencies (such as order of events and time distance between the events), and cannot handle unseen timestamps.

Symbolic methods for temporal knowledge graph reasoning use only qualitative predicates, e.g. StreamLearner (Omran, Wang, and Wang 2019), TLogic (Liu et al. 2022), ALRE-IR (Mei et al. 2022), TILP (Xiong et al. 2023), and thus have no capability of time prediction. The authors of NeuSTIP (Singh et al. 2023) solve this problem by introducing a Gaussian distribution for the query timestamp. However, depending on both path embedding and rule head embedding, their method is not applicable to the inductive setting. In addition, there are several inefficiency in their approach: (i) The distribution parameters are only related to the head predicate and the first predicate in the rule body, and hence many different rule patterns share the same parameters. (ii) Given a path connecting the subject and object entity, they only use one timestamp from the first body predicate, ignoring other useful information. (iii) They assume to know the temporal relation between the head predicate and the first body predicate, which is unknown in time prediction.

Another body of works looked into a related (forecasting) task of how to predict the time of next future event given all previous events. These works tend to use the temporal point process (TPP) for modelling. Embedding-based methods include Know-Evolve (Trivedi et al. 2017), GHNN (Han et al. 2020) and EvoKG (Park et al. 2022). Symbolic methods include TLPP (Li et al. 2020) and TELLER (Li et al. 2021). Compared with the general setting of Temporal Knowledge Graph Completion (TKGC), this forecasting task is more restrictive since not all history events are available in real-world applications. Due to the essence of causal sequence modelling, auto-regressive strategy, i.e., generating outputs based on past outputs and inputs, will also fail without the temporal order prior of the queries.

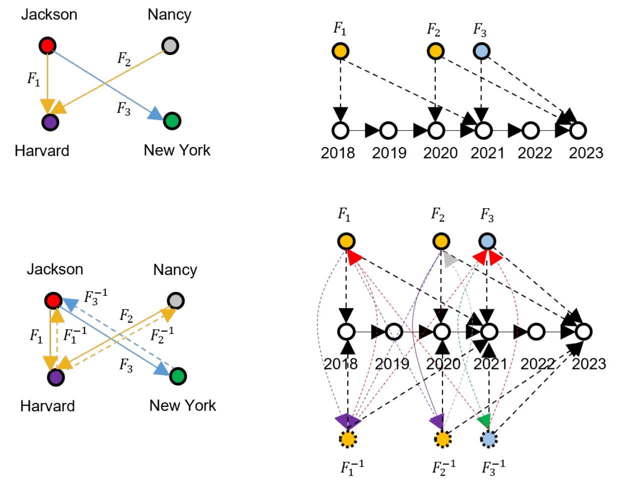

Figure 1: An example TKG (left) and the corresponding TEKG (right). The first row are the original versions, and the second row are the enhanced versions.

Our Framework

Problem Definition Let $\mathcal{E}$ be the set of entities, $\mathcal{P}$ be the set of predicates (or relations), $\mathcal{T}$ be the set of timestamps, $\mathcal{I} \subset \mathcal{T} \times \mathcal{T}$ be the set of intervals. A TKG $\mathcal{G} \subset \mathcal{E} \times \mathcal{P} \times \mathcal{E} \times \mathcal{I}$ is composed of quadruples (e_s, P, e_o, I) where $e_s, e_o \in \mathcal{E}$ denote subject and object entities, respectively, $P \in \mathcal{P}$ denotes predicate (or relation), and $I \in \mathcal{I}$ denotes the time. Let $(e_s, P, e_o, ?)$ be the query. Then, our objective is to predict the time I based on the observed facts from the same TKG $\mathcal{G}$. Compared with link predication, time prediction is more challenging as a regression (or interval estimation) task.

Temporal Event Knowledge Graph To solve the time prediction problem, given a TKG, we propose to convert the corresponding TEKG uniquely as the following. In TEKG, there exist two types of nodes: i) For each fact (e_s, P, e_o, I) in TKG, we define a corresponding event node $F \in \mathcal{F}$, where $\mathcal{F}$ is the set of facts, i.e., $F := (e_s, P, e_o, I)$. ii) For each timestamp t in TKG, a corresponding timestamp node T is defined in TEKG. Given the node definition, we further define three types of edges: i) An entity edge $E_{FF'}$ exists from F to F' iff some entity $e \in \mathcal{E}$ is the object of F as well as the subject of F'; ii) A temporal order edge $E_{TT'}$ exists between consecutive timestamp nodes T and T'; iii) A start time edge $E_{FT,s}$ or end time edge $E_{FT,e}$ exists from F to

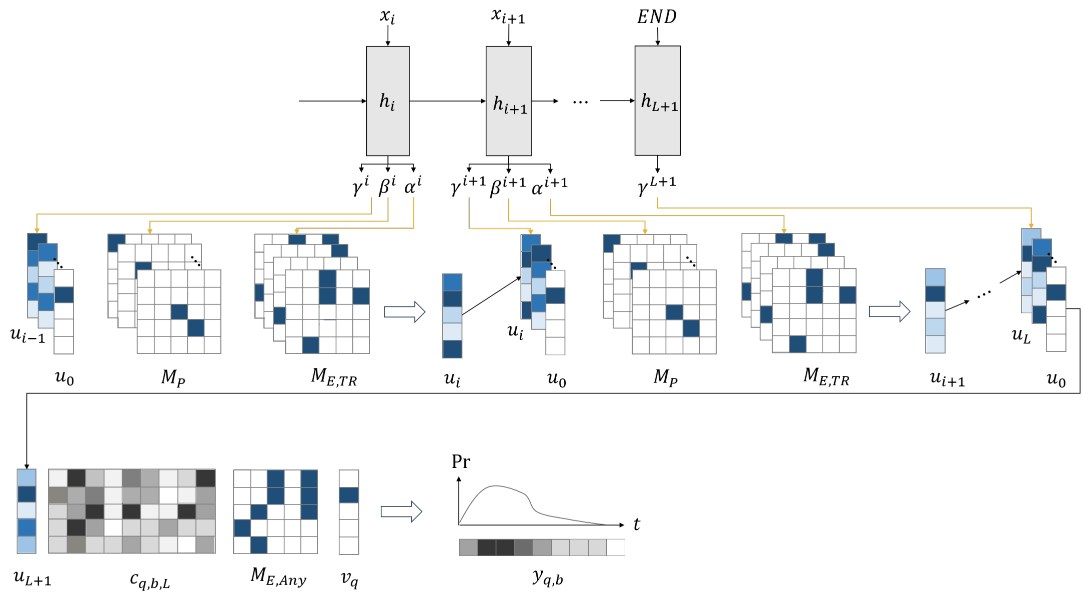

Figure 2: The illustration of rule-based time prediction.

T iff T is the start time or end time of F. Figure 1 visualizes an example TKG and the corresponding TEKG where $F_1 =$ ('Jackson', 'StudyIn', 'Harvard', $[2018, 2021]$), $F_2 =$ ('Nancy', 'StudyIn', 'Harvard', $[2020, 2023]$) and $F_3 =$ ('Jackson', 'WorkIn', 'New York', $[2021, 2023]$).

Given the above definition, we describe an important property of TEKG: if there exists a path of entity edges between two event nodes, we can always find a corresponding path in TKG. This property ensures that we can find in TKG the equivalents of logical rules learned in TEKG. Further, we consider the common enhancement strategy of adding inverse edges in TKG. These inverse edges which interchange the position of subject and object entity are introduced to allow bi-directional random walks in TKG. Similarly, in TEKG, we define mirror nodes F^{-1} to represent these inverse events. The entity edges and start time or end time edges of mirror nodes follow the same definition as original nodes. Figure 1 also visualizes the enhanced TKG and corresponding TEKG by adding inverse edges and mirror nodes, respectively. As the foundation of our approach, TEKG enables a differentiable random walk process. It allows us to better learn rule structure and confidence using gradient-based optimizer.

Temporal Logical Rules in TEKG A temporal logical rule of length $l \in \mathbb{N}$ in TEKG is defined as

$$\begin{aligned} Z_{R_P, I_1, \cdots, I_l}(I_q) \leftarrow & E(F_q, F_1) \wedge \cdots \wedge E(F_{l-1}, F_l) \\ & \wedge E(F_l, F_q) \wedge P_q(F_q) \wedge P_1(F_1) \wedge \cdots \wedge P_l(F_l) \\ & \wedge TR_1(I_1, I_2) \wedge \cdots \wedge TR_{l-1}(I_{l-1}, I_l) \end{aligned} \tag{1}$$

where F_q denotes the query event, and I_q denotes its interval. $\{F_i\}_{i \in \mathbb{N}}$ denote the variables of fact, and $\{I_i\}_{i \in \mathbb{N}}$ denote their interval. We ground these variables during inference. To allow generalization, $E(\cdot)$ denotes an entity edge related to any entity $e \in \mathcal{E}$, and $P_i(\cdot)$ denotes a grounded predicate, $TR_i(\cdot) \in \{\text{Before, Overlap, After, Any}\}$ denotes a grounded temporal relation between two intervals (timestamps), which is defined in (Xiong et al. 2023). Predicate $Z(\cdot) \in \{0, 1\}$ acts as an indicator on its arguments in relation to the rule $R_P := [P_q, P_1, \cdots, P_l, TR_1, \cdots, TR_{l-1}]$ and relevant intervals $\{I_1, \cdots, I_l\}$.

The left arrow in rule is called "implication", i.e., the rule body on the right implies the rule head on the left. The rule head $Z_{R_P, I_1, \cdots, I_l}(\cdot)$ indicates whether I_q satisfy R_P given the relevant intervals $\{I_1, \cdots, I_l\}$. The rule body contains the query predicate $P_q(\cdot)$, which is given in time prediction, and a cyclic path involving F_q, specified by predicate $P_i(\cdot)$ and temporal relation $TR_i(\cdot)$. The intuition here is to use logical rules to help us find l relevant events $\{F_1, \cdots, F_l\}$ for time prediction of the query F_q.

Logical Reasoning via Random Walk

Given a logical rule, the grounding process is to replace variables into constant terms. If the structure of rule body can be corresponded to a path in knowledge graph, the inference is equivalent to performing random walk under some constraints. In our temporal logical rules, there are three types of constraints: connectivity, predicates and temporal relations. We first design the operator for node attributes. Given the predicate $P(\cdot)$, for every node $F_m \in \mathcal{F}$, the operator $M_P \in \{0, 1\}^{|\mathcal{F}| \times |\mathcal{F}|}$ is defined such that its (m, m) entry is 1 iff $P(F_m)$ is true. We then define the operator for edge attributes. Given the connectivity $E(\cdot)$ and temporal relation $TR(\cdot)$, for every pair of events $F_m, F_n \in \mathcal{F}$, the operator $M_{E,TR} \in \{0, 1\}^{|\mathcal{F}| \times |\mathcal{F}|}$ is defined such that its (n, m) entry is 1 iff $\exists e \in \mathcal{E}$, s.t. both $E(F_m, F_n)$ and $TR(I_m, I_n)$ are true, where I_m and I_n are the time of F_m and F_n, respectively. Note that, we use a single operator for connectivity E and temporal relation TR, which implicitly implies a logical "and" in the rules. Compared with using separate operators, this strategy is more efficient in practice.

Given the operators, we define a **differentiable** temporal random walk with a recurrence formulation:

$$\mathbf{v}_{i+1} = \sum_{j=1}^{|TR|} \alpha_j^i M_{E,TR_j} \left(\sum_{k=1}^{|P|} \beta_k^i M_{P_k} \mathbf{v}_i \right) \tag{2}$$

where $\mathbf{v}_i \in [0,1]^{|\mathcal{F}|}$ is the state vector for step i, representing the probability distribution of different events. M_{P_k} with the predicate index k is the operator for P_k, and β_k^i is the weight for P_k at step i. Here we use a soft selection from different predicates: $\forall i$, $\beta_k^i \in [0,1]$ and $\sum_{k=1}^{|P|} \beta_k^i = 1$. Similarly, M_{E,TR_j} with the temporal relation index j is the operator for both E and TR_j, and α_j^i is the weight for TR_j at step i. We have $\forall i$, $\alpha_j^i \in [0,1]$ and $\sum_{j=1}^{|TR|} \alpha_j^i = 1$ as a soft selection from different temporal relations.

Time Prediction

To use inductive rules in (1) for time prediction, we introduce a conditional probability density function $G(\cdot)$ which describes the relationship between I_q and $\{I_1, \cdots, I_l\}$. There exist multiple choices for $G(\cdot)$. In this paper, we consider modelling the time gap between the query timestamp in I_q and the known timestamp in $\{I_1, \cdots, I_l\}$. The intuition is that the time gap shares the same probability distribution among different events. For example, the time gap between the same person's birth date and death date, i.e., a person's lifespan, follows a Gaussian distribution, across different persons. Similarly, the time gap between the same person's birth date and university graduation date follows another Gaussian distribution. Further, we consider these distributions evolving with time, e.g., the lifespan of modern people is significantly longer than that of ancient humans.

To be specific, the relationship between $I_q := [t_{q,s}, t_{q,e}]$ and $\{I_1 := [t_{1,s}, t_{1,e}], \cdots, I_l := [t_{l,s}, t_{l,e}]\}$ is defined as:

$$G_{R_P,b}(t_{q,b}|I_1, \cdots, I_l) = \sum_{i=1}^{l} a_{P_q,b,i} \cdot g_{R_P,b,i}(t_{q,b}|I_i) \tag{3}$$

where $G(\cdot)$ denotes the conditional probability density related to a pattern R_P and subscript $b = \{\text{'s'}, \text{'e'}\}$. Note that, (3) updates G whenever the rule in (1) is satisfied by the temporal pattern of events, i.e., $R_P : Z_{R_P, I_1, \cdots, I_l}(I_q) = 1$. The components $g(\cdot)$ denote the conditional probability density related to R_P, subscript b and index i with learnable weights $a_{P_q,b,i} \in [0,1]$ and $\sum_{i=1}^{l} a_{P_q,b,i} = 1$, where P_q denotes the query predicate.

$$\begin{aligned} g_{R_P,b,i}(t_{q,b}|I_i) = \; & w_{P_q,b,i} \cdot f_{R_P,b,i,s}\left(t_{q,b} - t_{i,s}\right) + \\ & \left(1 - w_{P_q,b,i}\right) \cdot f_{R_P,b,i,e}\left(t_{q,b} - t_{i,e}\right) \end{aligned} \tag{4}$$

where the components $f(\cdot)$ denotes the conditional probability density related to R_P, b, i and subscript 's' or 'e' with learnable weights $w_{P_q,b,i} \in [0,1]$. More details for the probability density function design are shown in the supplementary material. During training, functions $f(\cdot)$ will be fitted, and weights a and w will be learned. Further, we consider two options for modelling $t_{q,e}$: estimate it directly, or estimate duration $t_{q,d}$ and set $\widehat{t}_{q,e} := \widehat{t}_{q,s} + \widehat{t}_{q,d}$. We compare them in experiments.

Algorithm 1 : Rule Learning

Input: Temporal knowledge graph $\mathcal{G}$, query event F_q, target interval $I_q := [t_{q,s}, t_{q,e}]$.
Parameters: Maximum rule length L, flag for duration modelling $\mathbb{F}_{t_d}$.
Output: Rule patterns S_{R_P}, probability density functions $g_{R_P,b,i}(\cdot)$, attention vectors α, β, γ.

1: Convert $\mathcal{G}$ into temporal event knowledge graph $\mathcal{G}'$.
2: Sample cyclic random walks of length $l \in [1, L]$ on $\mathcal{G}'$ starting from either F_q or $F_{q^{-1}}$ to obtain rule pattern candidates S_{R_P} and local graph $\mathcal{G}'_q$.
3: For each rule pattern $R_P \in S_{R_P}$, fit the probability density functions $g_{R_P,b,i}(\cdot)$ for subscript $b \in \{\text{'s'}, \text{'e'}, \text{'d'}\}$ and index $i \in \{1, l\}$.
4: For all known events $F_m \in \mathcal{G}'_q$ and target timestamp $t_{q,b}$, calculate $g_{R_P,b,i}(t_{q,b}|I_m)$ if F_m satisfy R_P.
5: Calculate the probabilities $\Pr(t_{q,b})$ based on either (5) - (9) (event-split version) or (10) (rule-split version).
6: Learn the optimal attention vectors α, β, γ from (11).

Rule Learning & Application

The pseudocode for rule learning is described in Algorithm 1. Inspired by Neural-LP (Yang, Yang, and Cohen 2017), we use an attention mechanism to deal with varying rule length. Compared with Nerual-LP, which is developed for static link prediction, we add operators for connectivity, temporal relation and conditional probability density functions for time prediction.

$$\mathbf{u}_1 = M_{E,\text{Any}} \mathbf{u}_0 \tag{5}$$

$$\mathbf{u}_i = \sum_{j=1}^{|TR|} \alpha_j^i M_{E,TR_j} \left(\sum_{k=1}^{|P|} \beta_k^i M_{P_k} \left(\sum_{\tau=0}^{i-1} \gamma_\tau^i \mathbf{u}_\tau \right) \right) \tag{6}$$

$$\mathbf{u}_{L+1} = \sum_{\tau=0}^{L} \gamma_\tau^{L+1} \mathbf{u}_\tau \tag{7}$$

$$\mathbf{y}_{q,b} = \mathbf{v}_q^{\mathrm{T}} M_{E,\text{Any}} \left(\mathbf{c}_{q,b,L} \odot h_{\text{RPT}} \left(\mathbf{u}_{L+1}, |S_t| \right) \right) \tag{8}$$

where $\mathbf{u}_i \in [0,1]^{|\mathcal{F}|}$ denotes the partial inference result at step i. The recurrent formulation of (6) is based on (2). The difference is that we use an attention vector, i.e., $\forall i, \gamma_\tau^i \in [0,1]$ and $\sum_{\tau=0}^{i-1} \gamma_\tau^i = 1$, to softly select previous inference results. The initial result in (5) is the one-hot encoding of the query event, i.e., $\mathbf{u}_0 = \mathbf{v}_q \in \{0,1\}^{|\mathcal{F}|}$ whose q-th entry is 1 only. Further, $M_{E,\text{Any}}$ is the matrix operator for connectivity E and temporal relation 'Any'. Finally, $L \in \mathbb{N}$ denotes the maximum rule length. The inference result $\mathbf{u}_{L+1}$ in (7) is a soft selection from all the previous results $\{\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_L\}$. It represents the probability distribution we arrive at different events after at most L-step random walk. We calculate the time prediction $\mathbf{y}_{q,b} \in [0,1]^{|S_t|}$, where subscript $b \in \{\text{'s'}, \text{'e'}, \text{'b'}\}$, with (8). To allow efficient matrix operations, we quantize the timestamp range $[t_{min}, t_{max}]$ into a set of timestamps $S_t := \{t_{min}, \cdots, t_r, \cdots, t_{max}\}$. In experiments, we use a uniform discretization, and more complex quantizations can be adopted. Conditional probability matrix $\mathbf{c}_{q,b,L} \in [0,1]^{|\mathcal{F}|\times|S_t|}$ is based on the conditional probability density function $g_{R_P,b,l}(\cdot)$. The detailed calculation of $\mathbf{c}_{q,b,L}$ is given in the supplementary material. Function $h_{\text{RPT}}(\cdot)$ duplicates $\mathbf{u}_{L+1}$ along axis 1 such that $h_{\text{RPT}}(\mathbf{u}_{L+1}, |S_t|) \in [0,1]^{|\mathcal{F}|\times|S_t|}$. Operator $\odot$ denotes an element-wise multiplication, and the left part $\mathbf{v}_q^{\mathrm{T}} M_{E,\text{Any}}$ is introduced since we require the path to return to F_q. Note that, we only use the last event on the path for prediction. In experiments, we found that middle intervals $\{I_2, \cdots, I_{l-1}\}$ have a less significant impact on the performance. To involve the first event on the path, a trick here is to replace the query event F_q with its mirror node $F_{q^{-1}}$. Let $\mathbf{y}_{q,b}$ and $\mathbf{y}_{q^{-1},b}$ be the corresponding time predictions. Based on (3), the final time prediction result can be written as:

$$Y_{q,b} = a_{P_q,b} \cdot \mathbf{y}_{q,b} + (1 - a_{P_q,b}) \cdot \mathbf{y}_{q^{-1},b} \tag{9}$$

where the learnable weight $a \in [0,1]$ is related to the query predicate P_q and subscript b.

Further, (5) - (9) essentially provide an **event-split** version for time prediction. Given the rules, we first calculate the probability of arriving at different events, and then predicts the query given the interval of the events. Alternatively, the **rule-split** version is to directly predict the query given the rule confidence and the interval of the events satisfying the rule, i.e.,

$$\begin{aligned}(Y_{q,b})_r = \sum_{\kappa=1}^{|S_{R_p}|} \mathbf{s}_{R_p^\kappa}(\alpha,\beta,\gamma)\left(\left|S_{path}^\kappa\right|\right)^{-1} \sum_{\zeta_\kappa=1}^{|S_{path}^\kappa|} \Big(a_{P_q,b} \cdot \\ g_{R_P^\kappa,b,1}(t_r \mid I_1^{\zeta_\kappa}) + (1-a_{P_q,b}) \cdot g_{R_P^\kappa,b,l_\kappa}(t_r \mid I_{l_\kappa}^{\zeta_\kappa})\Big)\end{aligned} \tag{10}$$

where $Y_{q,b} \in [0,1]^{|S_t|}$ denotes the time prediction, and $(Y_{q,b})_r$ denotes its r-th entry corresponding to candidate $t_r \in S_t$. Rule R_P is indexed by κ, and S_{R_p} is the set of rules. The rule score function $\mathbf{s}(\cdot)$ is defined in (Yang, Yang, and Cohen 2017). Further, S_{path}^κ is the set of paths given F_q and R_P^κ. Learnable weight $a \in [0,1]$ is conditioned on query predicate P_q and subscript b. Finally, $I_1^{\zeta_\kappa}$, $I_{l_\kappa}^{\zeta_\kappa}$ are the corresponding intervals given the ζ_κ-th path with length l_κ.

Based on previous analysis, we know that the task of learning temporal logical rules is to learn the attention vectors α, β, γ which softly select predicates, temporal relations and rule lengths, respectively. Inspired by TILP (Xiong et al. 2023), we use an LSTM model illustrated in Figure 2 to ensure that current step's attention vectors depend on previous steps'. The calculation is given in the supplementary material. Further, to ensure the efficiency of our model on large TKGs, we adopt some acceleration strategies and analyze the time complexity in the supplementary material.

Training of the model is to minimize the log-likelihood loss:

$$\mathcal{L} = -\sum_{F_q}\left(\log \Pr\left(t_{q,s} \mid Y_{q,s}\right) + \log \Pr\left(t_{q,e} \mid Y_{q,e}\right)\right) \tag{11}$$

Algorithm 2 : Rule Application

Input: Temporal knowledge graph $\mathcal{G}$, query event F_q.
Parameters: Rule patterns S_{R_P}, probability density functions $g_{R_P,b,i}(\cdot)$, attention vectors α, β, γ, flag for duration modelling $\mathbb{F}_{t_d}$, quantized time range S_t.
Output: Predicted interval $\widehat{I}_q$.

Convert $\mathcal{G}$ into temporal event knowledge graph $\mathcal{G}'$.
Given S_{R_P}, obtain a local knowledge graph $\mathcal{G}_q'$ via cyclic walks starting from either F_q or $F_{q^{-1}}$ on $\mathcal{G}'$.
For all known events $F_m \in \mathcal{G}_q'$ and candidate timestamps $t_r \in S_t$, calculate $g_{R_P,b,i}(t_r|I_m)$ if F_m satisfy R_P.
Calculate predictions $Y_{q,b}$ with α, β, γ based on either (5) - (9) (event-split version) or (10) (rule-split version).
Estimate $\widehat{t}_{q,s}$ with $Y_{q,s}$ based on (12).
If $\mathbb{F}_{td}$ = True, estimate $\widehat{t}_{q,d}$ with $Y_{q,d}$ and set $\widehat{t}_{q,e} = \widehat{t}_{q,s} + \widehat{t}_{q,d}$, else directly estimate $\widehat{t}_{q,e}$ with $Y_{q,e}$.
Set $\widehat{I}_q = [\widehat{t}_{q,s}, \widehat{t}_{q,e}]$.

where $\Pr(t_{q,b} \,|Y_{q,b})$ denotes the probability of $t_{q,b}$ given the prediction $Y_{q,b} \in [0,1]^{|S_t|}$.

The pseudocode for rule application is described in Algorithm 2. Given the learned rule patterns S_{R_P}, probability density functions $g_{R_P,b,i}(\cdot)$ and attention vectors α, β, γ, inference of the model is to find the timestamp t_r in S_t that maximizes the probability:

$$\widehat{t}_{q,b} = \underset{t_r \in S_t}{\operatorname{argmax}} \Pr\left(t_r \mid Y_{q,b}\right) \tag{12}$$

The underlying logic of our method is to model probability distribution of the query interval. An alternative strategy is to directly perform regression. We found in experiments that the regression-based approach is essentially memorizing the answer. Their performance becomes much worse in the future event time forecasting.

Experiments

Datasets We evaluate the proposed method TEILP on five benchmark temporal knowledge graph datasets: WIKIDATA12k, YAGO11k (Dasgupta, Ray, and Talukdar 2018), ICEWS14, ICEWS05-15 (García-Durán, Dumančić, and Niepert 2018), and GDELT100 (Leetaru and Schrodt 2013). According to the type of event time, we divide them into two classes: interval-based (WIKIDATA12k, YAGO11k) and timestamp-based (ICEWS14, ICEWS05-15, GDELT100). All these datasets contain temporal facts in a quadruple form, e.g., (Iran, Express intent to meet or negotiate, China, 2014-02-02). For interval-based datasets, we know both the start and end time of an event, while for timestamp-based datasets, we only know the start time. To ensure a fair comparison, we use the split provided by (Jain et al. 2020) for WIKIDATA12k, YAGO11k, ICEWS14, ICEWS05-15 datasets and (Goel et al. 2020) for GDELT dataset. Note that, we delete the repeated edges in GDELT, and preserve the top 100 entities with the most edges. In the supplementary material, we provide a detailed introduction and dataset statistics.

Model	**YAGO11k**		**WIKIDATA12k**		**ICEWS14**	**ICEWS05-15**	**GDELT100**
	aeIOU	TAC	aeIOU	TAC	MAE	MAE	MAE
HyTE	0.0541	0.0546	0.0541	0.0722	117.71	1315.46	122.24
DE-SimplE	0.0663	0.0877	0.0484	0.0519	83.87	1348.99	110.35
TNT-Complex	0.0840	0.0975	0.2335	0.2640	120.14	1281.37	115.97
TimePlex (base)	0.1421	0.1503	0.2620	0.3057	99.58	992.04	109.76
TimePlex	0.2003	0.2253	0.2636	0.3054	87.39	1098.07	102.88
NeuSTIP (base)	0.1642	-	0.2627	-	-	-	-
NeuSTIP w/ Gadgets	0.2635	-	0.2630	-	-	-	-
NeuSTIP w/ KGE	0.2488	-	0.2735	-	-	-	-
GBDT	0.1336	0.1432	0.2923	0.2693	85.81	910.16	94.92
TEILP (event-split-td)	0.2675	0.2589	0.3086	0.2995	-	-	-
TEILP (event-split)	**0.2996**	0.2861	0.3260	0.3026	70.72	812.07	97.54
TEILP (rule-split-td)	0.2573	0.2575	0.3228	0.3120	-	-	-
TEILP (rule-split)	0.2977	**0.2877**	**0.3285**	**0.3153**	**70.06**	**774.01**	**94.45**

Table 1: Time prediction performance on the benchmark datasets.

Evaluation Metrics For interval-based datasets, we adopt a new evaluation metric aeIOU, proposed by (Jain et al. 2020). It is developed from Intersection over Union (IOU), and has desirable properties for the interval time prediction task. We also use another popular metric, TAC (Ji et al. 2011) (Surdeanu 2013) for evaluating intervals. The definitions are given as:

$$\text{aeIOU}\left(I, \widehat{I}\right) = \frac{\max\left\{1, \text{vol}\left(I \cap \widehat{I}\right)\right\}}{\text{vol}\left(\text{ConvHull}(I, \widehat{I})\right)} \tag{13}$$

$$\text{TAC}\left(I, \widehat{I}\right) = \frac{1}{2}\left[\frac{1}{1+|t_s - \widehat{t}_s|} + \frac{1}{1+|t_e - \widehat{t}_e|}\right] \tag{14}$$

where $I := [t_s, t_e]$ denotes the ground truth, $\widehat{I} := [\widehat{t}_s, \widehat{t}_e]$ denotes the prediction, $\text{vol}(I_a)$ represents the length of interval I_a, $\text{ConvHull}(I_a, I_b)$ represents the smallest single continuous interval containing both I_a and I_b, and 1 represents the smallest time granularity. We use '1 year' for both WIKIDATA12k and YAGO11k. From (13) and (14), we know that TAC focuses on the prediction accuracy of the start and end timestamp of an interval, while aeIOU focuses on the similarity between two intervals. Both of them fall into the range of $[0, 1]$, and a higher value means a better performance. For timestamp-based datasets, we follow the settings in (Trivedi et al. 2017), using Mean Absolute Error (MAE) as the evaluation metric with the smallest time granularity of '1 day'. Obviously, a lower MAE means a better model performance.

Baseline Methods We compare TEILP[1] with stat-of-the-art baselines for time prediction over knowledge graphs: HyTE (Dasgupta, Ray, and Talukdar 2018), DE-SimplE (Goel et al. 2020), TNT-Complex (Lacroix, Obozinski, and Usunier 2020), TimePlex (Jain et al. 2020), and NeuSTIP (Singh et al. 2023). As embedding-based methods, HyTE, DE-SimplE and TNT-Complex rank different timestamps given the known subject, object entity and relation. To obtain a time interval prediction, we adopt a greedy coalescing strategy proposed in (Jain et al. 2020). TimePlex is also built on embeddings, but it introduces additional temporal constraints such as relation recurrence, ordering and time gap distribution. To contrast, NeuSTIP is a temporal neuro-symbolic model which learns logical rules and Gaussian distributions from knowledge graphs. In addition, we consider gradient-boosted decision trees (GBDT), the conventional machine learning algorithm for a regression task.

[1]Code and data available at https://github.com/xiongsiheng/TEILP.

Results and Analysis

The results of the experiments are shown in Table 1, where TEILP outperforms all baselines with respect to all metrics. For our method, due to the choice of event-split or rule-split modelling, and whether to use interval duration prediction (-td), there are four versions, as noted in the table. The maximum rule length of our method is set to 5 for YAGO11k, and 3 for the others. For NeuSTIP, we use the results reported in their paper. For other baselines, we run the code on all the datasets. To deal with the incomplete events in YAGO11k and WIKIDATA12k, we remove the test queries with missing time when evaluating.

Following conclusions can be made from the results. Conventional embedding-based methods are not suitable for time prediction since they consider it as a ranking problem similar to link prediction. Different from entities, timestamps come from a continuous space and have intrinsic connections such as order and distance. TimePlex improves its performance by adding temporal constraints. However, these constraints are still not enough for accurate time prediction. NeuSTIP introduces a similar probability distribution modelling while our method involves much more timestamps enhanced by multiple distribution types and temporally-evolving parameters. Finally, the main limitation of conventional machine learning algorithms such as GBDT is the failure to capture the complex interactions between different events.

Model	**YAGO11k**		**WIKIDATA12k**		**ICEWS14**	**ICEWS05-15**	**GDELT100**
	aeIOU	TAC	aeIOU	TAC	MAE	MAE	MAE
GHNN	-	-	-	-	30.08	150.47	23.97
EvoKG	-	-	-	-	29.57	148.17	23.81
TimePlex (base)	0.0700	0.0811	0.0985	0.1031	33.42	164.05	27.25
TimePlex	0.0849	0.0924	0.1120	0.1214	32.08	157.72	25.87
GBDT	0.1258	0.1311	0.1688	0.1771	26.81	141.38	22.78
TEILP	**0.3175**	**0.3763**	**0.3971**	**0.4172**	**24.86**	**134.89**	**19.90**

Table 2: Time prediction performance in future event time forecasting.

Learned Rules and Distributions Given a query, our approach uses a chain of events, which connects subject and object entities, to reason the missing time. In particular, we focus on the events happening on either subject or object, which are of the same relation as the query or some other directly related relations. We found that the time gap between the query time and known relevant timestamps follows a certain distribution. We show some examples of learned rules and distributions here. In the YAGO11k dataset, we learn the following rule:

$$\begin{aligned} Z_{R_P,I_1,I_2,I_3}(I_q) \leftarrow\ & E\left(F_q, F_1\right) \wedge E\left(F_1, F_2\right) \wedge E\left(F_2, F_3\right) \\ & \wedge E\left(F_3, F_q\right) \wedge \text{isAffiliatedTo}\left(F_q\right) \wedge \text{isAffiliatedTo}^{-1}\left(F_1\right) \\ & \wedge \text{isAffiliatedTo}\left(F_2\right) \wedge \text{isAffiliatedTo}^{-1}\left(F_3\right) \\ & \wedge \text{Before}(I_1, I_2) \wedge \text{Overlap}(I_2, I_3) \end{aligned}$$

Given a query $F_q =$ ('David Davis (Supreme Court justice)', 'isAffiliatedTo', 'Republican Party (United States)', $[?, ?]$), we ground the rules with $F_1 =$ ('Republican Party (United States)', 'isAffiliatedTo^{-1}', 'Nathaniel P. Banks', $[1857, 1875]$), $F_2 =$ ('Nathaniel P. Banks', 'isAffiliatedTo', 'Independent politician', $[1875, 1877]$), and $F_3 =$ ('Independent politician', 'isAffiliatedTo^{-1}', 'David Davis (Supreme Court justice)', $[1872, 1886]$). The generated conditional probability distribution is shown in Figure 3 (left) where the ground truth and our answer are $[1854, 1870]$ and $[1863, 1871]$, respectively. Similarly, given another query $F_q =$ ('John Reynolds (Canadian politician)', 'isAffiliatedTo', 'Reform Party of Canada', $[?, ?]$), we ground the rules with $F_1 =$ ('Reform Party of Canada', 'isAffiliatedTo^{-1}', 'Raymond Speaker', $[1992, 2000]$), $F_2 =$ ('Raymond Speaker', 'isAffiliatedTo', 'Canadian Alliance', $[2000, 2003]$), and $F_3 =$ ('Canadian Alliance', 'isAffiliatedTo^{-1}', 'John Reynolds (Canadian politician)', $[2000, 2003]$). The generated conditional probability distribution is shown in Figure 3 (right) where the ground truth and our answer are $[1997, 2000]$ and $[1994, 2002]$, respectively. More examples are shown in the supplementary material.

More Difficult Problem Settings Inspired by (Xiong et al. 2023), we demonstrate that TEILP, which uses symbolic representations and conditional probability density functions for time prediction, is more robust than embedding-based methods. In the low-data scenario, given the same validation and test set, we change the size of the training set over a broad range. We observe a less pronounced drop in our model performance when training samples are limited. In the imbalanced scenario, given the same validation and test set, we intentionally reduce the number of training events of a certain type to investigate its effect on accuracy. We show that the learning process of TEILP is less affected by data imbalance. Further, we consider the scenario of future event forecasting which brings even more challenges to time prediction than the link prediction studied in (Xiong et al. 2023). In experiments, we re-split the datasets according to the start time of events. Our finding from Table 2 is that models based on temporal point process will fail with sparse training data. In contrast, both logical rules and time gap distribution modelling provide our method the generalization to unseen entities and timestamps. We provide more detailed results and analysis for all the settings in the supplementary material.

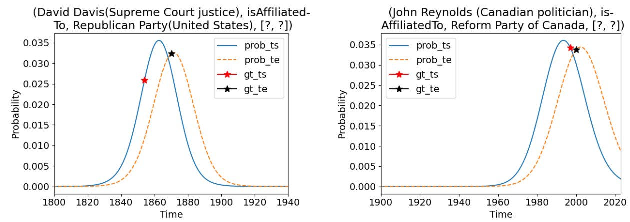

Figure 3: The conditional probability distribution for the query interval given by TEILP.

Conclusion

TEILP, an inductive logical reasoning framework, has been proposed to predict time of events in knowledge graphs. Predicting both timestamp and the interval of time can be handled by our framework. Experiments on five benchmark datasets indicate that TEILP achieves better performance than state-of-the-art methods while providing logical explanations. In addition, we consider more difficult scenarios in temporal knowledge graph reasoning, where TEILP outperforms all baseline methods. An interesting direction for future work is to predict entity attributes or event attributes which are changing with time. We need to develop efficient tools for numerical variable modelling and effectively combine them with logical rules learning, which will further extend the expressive power of neural-symbolic methods.

Acknowledgments

This work was supported by a sponsored research award by Cisco Research.

References

Cai, B.; Xiang, Y.; Gao, L.; Zhang, H.; Li, Y.; and Li, J. 2022. Temporal knowledge graph completion: A survey. *arXiv preprint arXiv:2201.08236*.

Cai, L.; Janowicz, K.; Yan, B.; Zhu, R.; and Mai, G. 2021. Time in a box: advancing knowledge graph completion with temporal scopes. In *Proceedings of the 11th on Knowledge Capture Conference*, 121–128.

Chen, Z.; Liao, J.; and Zhao, X. 2023. Multi-granularity Temporal Question Answering over Knowledge Graphs. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 11378–11392. Toronto, Canada: Association for Computational Linguistics.

Dasgupta, S. S.; Ray, S. N.; and Talukdar, P. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2001–2011.

García-Durán, A.; Dumančić, S.; and Niepert, M. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4816–4821. Brussels, Belgium: Association for Computational Linguistics.

Goel, R.; Kazemi, S. M.; Brubaker, M.; and Poupart, P. 2020. Diachronic embedding for temporal knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 3988–3995.

Han, Z.; Ma, Y.; Wang, Y.; Günnemann, S.; and Tresp, V. 2020. Graph hawkes neural network for forecasting on temporal knowledge graphs. *arXiv preprint arXiv:2003.13432*.

Jain, P.; Rathi, S.; Chakrabarti, S.; et al. 2020. Temporal knowledge base completion: New algorithms and evaluation protocols. *arXiv preprint arXiv:2005.05035*.

Ji, H.; Grishman, R.; Dang, H. T.; and Griffitt, K. 2011. Overview of the TAC2011 Knowledge Base Population (KBP) Track. In *Presentation at the Fourth Text Analysis Conference (TAC 2011), November*, 14–15.

Lacroix, T.; Obozinski, G.; and Usunier, N. 2020. Tensor Decompositions for Temporal Knowledge Base Completion. In *International Conference on Learning Representations*.

Leetaru, K.; and Schrodt, P. A. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, 1–49. Citeseer.

Li, S.; Feng, M.; Wang, L.; Essofi, A.; Cao, Y.; Yan, J.; and Song, L. 2021. Explaining point processes by learning interpretable temporal logic rules. In *International Conference on Learning Representations*.

Li, S.; Wang, L.; Zhang, R.; Chang, X.; Liu, X.; Xie, Y.; Qi, Y.; and Song, L. 2020. Temporal Logic Point Processes. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 5990–6000. PMLR.

Liu, Y.; Ma, Y.; Hildebrandt, M.; Joblin, M.; and Tresp, V. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 4120–4127.

Mei, X.; Yang, L.; Cai, X.; and Jiang, Z. 2022. An Adaptive Logical Rule Embedding Model for Inductive Reasoning over Temporal Knowledge Graphs. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 7304–7316.

Omran, P. G.; Wang, K.; and Wang, Z. 2019. Learning Temporal Rules from Knowledge Graph Streams. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*.

Park, N.; Liu, F.; Mehta, P.; Cristofor, D.; Faloutsos, C.; and Dong, Y. 2022. Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 794–803.

Saxena, A.; Chakrabarti, S.; and Talukdar, P. 2021. Question Answering Over Temporal Knowledge Graphs. arXiv:2106.01515.

Singh, I.; Kaur, N.; Gaur, G.; and Mausam. 2023. NeuSTIP: A Novel Neuro-Symbolic Model for Link and Time Prediction in Temporal Knowledge Graphs. arXiv:2305.11301.

Surdeanu, M. 2013. Overview of the TAC2013 Knowledge Base Population Evaluation: English Slot Filling and Temporal Slot Filling. *TAC*, 8: 2.

Trivedi, R.; Dai, H.; Wang, Y.; and Song, L. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *international conference on machine learning*, 3462–3471. PMLR.

Xiong, S.; Yang, Y.; Fekri, F.; and Kerce, J. C. 2023. TILP: Differentiable Learning of Temporal Logical Rules on Knowledge Graphs. In *The Eleventh International Conference on Learning Representations*.

Yang, F.; Yang, Z.; and Cohen, W. W. 2017. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30.