Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)

doi:10.3233/FAIA230878

Short-Term Sequential Pattern for Temporal Knowledge Graph Reasoning

Hongyang LI[1], Cui ZHU[2], Wenjun ZHU[3], Yunfeng ZHOU[4] and Xinyu ZHANG[5]
School of Computer Science, Faculty of Information Technology,
Beijing University of Technology, Beijing, China

Abstract. Temporal knowledge graphs (TKGs) have been widely used in various fields, and predicting missing knowledge graph inference has been widely explored. However, the task of reasoning about potential future facts on TKGs is more challenging and has attracted the attention of researchers. As the unknowability of future events complicates inference, a thorough study of the characteristics of historical facts becomes crucial. The study of the concurrency of historical events and the underlying common patterns of relation- ships facilitates reasoning about future facts. In this paper, we propose a novel representation learning model based on Short-Term Sequential Patterns for TKG reasoning, namely STSP. By modeling TKG sequences recurrently and learning representations of entities and relations. Specifically, the STSP encoder uses three main modules. Concurrent facts for each timestamp are modeled using a convolution-based relation-aware GCN. The entity-aware attention module is used to integrate the entity representation of the current timestamp and the previous timestamp. The sliding window mechanism is used to learn different relations sequentially. The entity and relation representations are then handed over to a translation-based decoder for final reasoning. We use four benchmark datasets to evaluate the proposed approach. The experimental results show that STSP out-performs state-of-the-art TKG reasoning methods and obtains substantial performance.

Keywords. Temporal Knowledge Graph, Knowledge Reasoning, Attention Mechanisms, Graph Convolutional Network, Sliding Window

1. Introduction

Knowledge graph has been used in search engine, information recommendation, intelligent question and answer, and other fields. However, they are usually incomplete, which limits the implementation of downstream tasks. For reasoning and completion in static knowledge graphs, there is a wealth of research at this stage. However, events undergo complex dynamic changes over time, and it becomes a challenge to describe and implement a reasoning approach to temporal knowledge graphs (TKGs).

TKGs can be seen as KG sequences with time, where the facts are represented by the quadruple (subject entity, relation, object entity, timestamp), and each fact will have

---

[1] Hongyang LI, Beijing University of Technology; e-mail: lihyang@emails.bjut.edu.cn
[2] Corresponding Author: Cui ZHU, Beijing University of Technology; e-mail: cuizhu@bjut.edu.cn
[3] Wenjun ZHU, Beijing University of Technology; e-mail: zhuwenjun@bjut.edu.cn
[4] Yunfeng ZHOU, Beijing University of Technology; e-mail: yfzhou@emails.bjut.edu.cn
[5] Xinyu ZHANG, Beijing University of Technology; e-mail: zxybjut@emails.bjut.edu.cn

a timestamp indicating when the event occurred. For example, (Anna, enroll, Tsinghua University, 2014-09-01) indicates Anna enrolled in Tsinghua University on 2014-09-01.

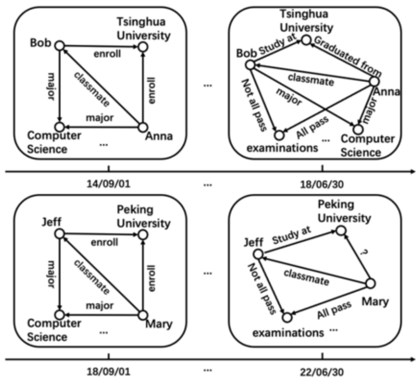


Figure 1. An illustration of temporal reasoning over a TKG.

Reasoning on TKGs is mainly divided into two types: interpolation (refers to inferring missing facts for *t* satisfied $t_0 \leq t \leq t_T$) and extrapolation (focus on predicting future events for *t* beyond $t_T (t > t_T)$). This paper addresses the problem of extrapolation of entities and relations in TKG reasoning.

Events will develop over time. Historical facts carry a certain sequential nature, and events that are close in time will have a stronger influence on current events, and relations change over time with certain common patterns. These characteristics are driving for the development of events and can be captured to predict the events that will follow.

As shown in Figure 1, target is (Mary,?, Peking University, 2022-06-30) . According to the historical information, we can learn a common feature about the school system, the university is a four-year system. So the result can be obtained as (Mary,Graduated from,Peking University,2022-06-30). Therefore, windowing historical facts that are close in time to get the common features of the relation can limit the prediction range and improve the accuracy of the final result.

In recent years, RE-NET[2] can only capture and encode directly involved historical facts in sequence, with a limited timestamp. RE-GCN[4] and RE-GAT[5] only model historical facts sequentially, using the same nonlinear scale for learning the entity representations of neighboring timestamps, without considering the factor of different influence of entities at different moments. None considers the common pattern of relations in the short term.

We therefore propose a model for capturing short-term sequential pattern features, called short-term sequential pattern (STSP), which models entities and relations in each timestamp cyclically through KG sequences. The entities and relations are encoded separately using encoders. use the temporal subgraphs as sequences to capture the common patterns of relations in short-term time using sliding windows. Learning of entities using attention mechanism. Finally, reasonings are implemented using decoders.

In general, this paper makes the following contributions:

- We propose a short-term sequential learning model STSP for temporal reasoning in TKGs, which enhances the representation of patterns of sequential influence between different relations by learning commonalities between them through sliding windows while considering the sequential nature of historical facts.

- We characterize the TKG through the lens of KG sequences, using an attention mechanism to encode timestamped neighboring subgraph sequences to achieve a trade-off between the influence of the current moment and the historical moment.
- We conducted entity reasoning and relation reasoning experiments on four real datasets, and the results show that STSP model can achieve better performance.

2. Related Works

Reasoning on TKGs is mainly divided into two types: interpolation and extrapolation.

- Interpolation attempts to infer missing facts at historical timestamps. TTransE[10], TATransE[11] are variants of TransE[6] which treat relations and time as translations between entities. HyTE[12],Hybrid-TE[13] associate each timestamped entity and relation with the corresponding hyperplane. Most importantly, neither of them can predict the facts of future timestamps, nor do they apply to the extrapolation setup.
- The extrapolation focused on in this paper attempts to infer new facts about future timestamps based on historical facts. Know Evolve[1] and DyREP[15] use point processes to model the occurrence of events and determine the likelihood of future events by estimating conditional probabilities through point processes.

RE-NET[2] first models historical facts as subgraph sequences for learning, but it ignores the structural dependence of KGs at different points in time.CyGNet[3] uses a sequential replication network to model repeated facts, and its reasoning results focus on facts that occur repeatedly multiple times. The factor that the influence of facts varies from moment to moment is ignored.

The most relevant work is RE-GCN[4] and RE-GAT[5], which learns entity and relational representations for each timestamp by sequentiallycovering the local history structure. The learning of entities is performed using a gating mechanism of nonlinear functions, without differentiating entity influence weights. However, if the learning of the relation order is too long, it will lead to inaccurate grasping of the common pattern between the relations.

3. The Proposed Method

3.1. Task Definition

We first describe the notations for the temporal knowledge graph (TKG), let $\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$ *and* $\mathcal{F}$ denote the sets of entities, relations, timestamps, and facts. We formalize a TKG $\mathcal{G}$ as a sequence of subgraphs, $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \cdots\cdots, \mathcal{G}_{\mathrm{T}}\}$. The subgraph $\mathcal{G}_{\mathrm{t}} = (\mathcal{E}_{\mathrm{t}}, \mathcal{R}, \mathcal{F}_{\mathrm{t}})$ at t is a directed multi-relational graph, where $\mathcal{E}_{\mathrm{t}}$ is the set of entities, $\mathcal{R}$ is the set of relations, and $\mathcal{F}_{\mathrm{t}}$ is the set of facts at t, so a graphical snapshot represents the TKG over time. A fact in $\mathcal{F}_{\mathrm{t}}$ can be formalized as a quadruple (s, r, o, t), where $s, o \in \mathcal{E}_{\mathrm{t}}$ *and* $r \in \mathcal{R}$. Represents there is a relation r between the entity s and the entity o at t.

TKG reasoning can be divided into two parts: entity reasoning aims to predict the missing entity of a query $(s, r, ?, t+1)$ or $(?, r, o, t+1)$, relation reasoning task attempts to predict the missing relation of a query $(s, ?, o, t+1)$.

For all historical event sets before a given time stamp t, under the assumption that the reasoning of the facts at a future timestamp $t+1$ depends on the KGs at the latest k timestamps (i.e., $\{\mathcal{G}_{t-k,}, \mathcal{G}_{t-k+1}, \cdots, \mathcal{G}_t\}$), and the information of the historical KG sequence is modeled in the evolutional embedding matrices of the entities $\mathcal{E}_t \in \mathbb{R}^{|\mathcal{E}| \times d_{\mathcal{E}}}$ and the relations $\mathcal{R}_t \in \mathbb{R}^{|\mathcal{R}| \times d_{\mathcal{R}}}$ at timestamp t, where $d_{\mathcal{E}}$ and d_R is the dimension of the event entity vector representations and event relational type vector representations, the two temporal reasoning tasks can be formulated as follows:

Entity reasoning task, use the subject entities s, the relation r, and the historical KG sequences $\mathcal{G}_{t-k+1:t}$ to calculate the conditional probability for all objects entities:

$$\vec{p}(o \mid \mathcal{G}_{t-k+1:t}, s, r) = \vec{p}(o \mid \mathcal{E}_t, \mathcal{R}_t, s, r) \quad (1)$$

Similarly, the problem of relation reasoning can be defined as follows:

$$\vec{p}(r \mid \mathcal{G}_{t-k+1:t}, s, o) = \vec{p}(r \mid \mathcal{E}_t, \mathcal{R}_t, s, o) \quad (2)$$

3.2. Model Overview

STSP recursively takes the structural features of each timestamp in KG and learns them, encodes neighboring subgraph sequences using an attention mechanism, learns the commonality of temporally similar facts and the sequential nature of historical facts using sliding windows. Based on the learned embeddings, the decoder can reason about the events on future timestamps. As shown in Figure 2, STSP takes the graph at time t as input, encodes it through an encoder structure, and then uses the decoder to complete the reasoning at time t+1. The encoder acts on entity encoding and relation encoding for learning historical KG sequences and obtaining evolutionary representations of entities and relations. The decoder takes as input the embedding generated by the final timestamp of the encoder and uses a translation-based scoring function to score the corresponding reasoning tasks. The following are detailed in turn.

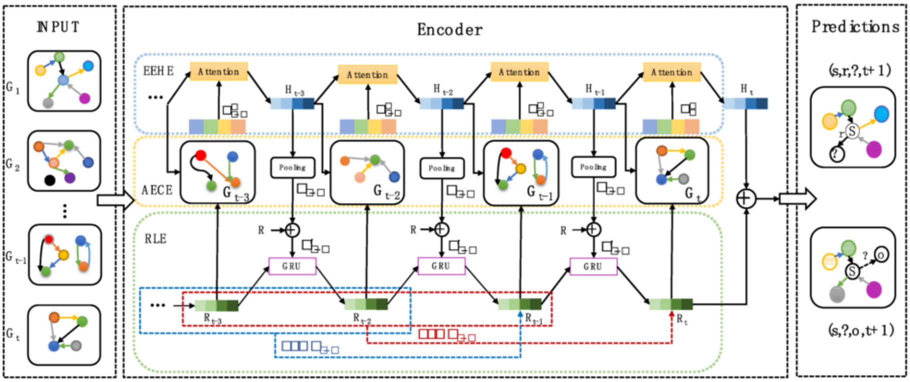


Figure 2. An illustrative diagram of the temporal reasoning of STSP at timestamp t+1. Take the sliding window length of 3 as an example.

3.3. Loop Embedding Encoder Module

The cyclic embedding encoder module consists of a convolution-based relation-aware GCN, an entity-aware attention component, a sliding window, a gated flow unit (GRU).

3.3.1. Aggregation Embedding of Current Entities (AECE)

At each timestamp, we model concurrent facts by capturing the association of relations between entities and between shared entities. GCN[17] and its refinements [7][8][9] have been widely used as a modeling model for multi-relational graph structures, so we use a one-dimensional convolution-based ω-layer relation-aware GCN to aggregate relations and multi-hop neighbor information at each timestamp. Formally, for a KG at timestamp t, the embedding of object entity o under layer l messaging and the fused embedding operation of the entity s with the relation under layer l messaging can obtain the embedding of an object entity o at layer $l+1$, so the aggregator is defined as:

$$\vec{h}_{o,t}^{l+1} = \sigma\left(\frac{1}{c_o}\sum\nolimits_{(s,r),\exists(s,r,o)\epsilon\mathcal{F}_t} W_r^l \,\mathrm{Conv}\left(\vec{h}_{s,t}^l, \vec{r}_t\right) + W_o^l \vec{h}_{o,t}^l\right) \tag{3}$$

where $\vec{h}_{o,t}^{l+1}$ denotes the $l+1^{th}$ layer embedding of entity o at timestamp t; $\vec{h}_{s,t}^l$, $\vec{h}_{o,t}^l$, $\vec{r}_t$ denote the l^{th} layer embedding of entity s, o, and relation r at timestamp t, respectively; $\mathrm{Conv}\left(\vec{h}_{s,t}^l, \vec{r}_t\right)$ denotes the merging of relation r with entity s by convolution operation; W_r^l and W_o^l are the aggregation features and self-loop parameters of the l^{th} layer, c_o is a normalizing factor, and $\sigma(\cdot)$ denotes the RReLU activation function.

3.3.2. Evolution Embedding of Historical Entities (EEHE)

Sequential learning of historical facts can help to better learn the behavioral trends and preferences of entities. To better learn entity o in a subgraph sequence, we use the entity attention-aware module to incorporate the entity information of the previous timestamp into the operation.

Then the final embedding H_t of the entities at timestamp t is affected by two parts: the embedded expression H_t^ω of entities under the current timestamp t, and the final embedding of the entity at the previous timestamp t-1 is obtained as H_{t-1}. The Attention values of two components are computed after performing activation operations:

$$\begin{aligned} \alpha_{H_t^\omega} &= \frac{\exp(\tanh(PH_t^\omega))}{\exp(\tanh(PH_t^\omega))+\exp(\tanh(QH_{t-1}))} \\ \alpha_{H_{t-1}} &= \frac{\exp(\tanh(QH_{t-1}))}{\exp(\tanh(PH_t^\omega))+\exp(\tanh(QH_{t-1}))} \end{aligned} \tag{4}$$

where $\alpha_{H_t^\omega}$ denote the attention score of the entity embedding expression under the current timestamp t (H_t^ω), and $\alpha_{H_{t-1}}$ donate the attention score of the final embedding of the entity obtained at the previous timestamp $t-1$ (H_{t-1}); tanh $(\cdot)$ denotes the activation function, and P, Q are the learnable parameters. Then the final embedding H_t of the entity at timestamp t can be expressed as:

$$H_t = \mathrm{prob}[0] \otimes H_t^\omega + prob[1] \otimes H_{t-1} \tag{5}$$

where $\otimes$ denotes the dot product operation; prob$[0]$, $prob[1]$ are the fraction of attention that varies linear variation between 0 and 1 after normalization by the Softmax function:

$$\text{prob} = \text{Softmax}\left(\alpha_{H_t^{\omega}} \oplus \alpha_{H_{t-1}}\right) \tag{6}$$

3.3.3. Relation Loop Embedding (RLE)

The module aims to model the historical patterns of relations, and potential historical events containing potential relation characteristics and patterns that can also represent historical trends and regularity. To cover as many patterns of relations in historical facts as possible, a sliding window model of short-term sequences is used to learn relations under multiple timestamps, and the whole process is modeled using the GRU component.

First, the relation at the current timestamp t needs to be learned first. We consider that the entity $\mathcal{E}_{r,t} = \{\, i \mid (i,r,o,t) \text{ or} (s,r,i,t) \in \mathcal{E}_t \,\}$ associated with r at the previous timestamp t-1 will have an impact on the embedding expression of r at the current timestamp t. The embedding matrix of the entity associated with r at timestamp t-1 is averaged pooling operation and spliced with the relational embedding vector:

$$\vec{r}_t^{\,\prime} = \left[\text{pooling}\left(H_{t-1}, \mathcal{E}_{r,t}\right) \oplus \vec{r}\right] \tag{7}$$

where $\oplus$ represents the concatenation operation; $\vec{r}$ is the randomly initializes the embedding of relation r in the relation matrix R.

Then, the previous timestamp relational embedding matrix $new\mathcal{R}_{t-1}$ is obtained by combining the relational embedding vectors of the previous m timestamps through the sliding window mechanism, and finally, the relational embedding matrix is updated from $\mathcal{R}_{t-1}$ to $\mathcal{R}_t$ through GRU:

$$new\mathcal{R}_{t-1} = \textstyle\sum_{x=t-m}^{t-1} \mathcal{R}_x \tag{8}$$

$$\mathcal{R}_t = GRU(new\mathcal{R}_{t-1}, \mathcal{R}_t') \tag{9}$$

where $new\mathcal{R}_{t-1}$ is the sum of the relational embedding matrices for the previous m moments; $\mathcal{R}_t'$ consists of $\vec{r}_t^{\,\prime}$ for all relations, and $\mathcal{R}_\mathrm{t} \in \mathbb{R}^{|R| \times d}$ is the relational embedding matrix at timestamp t.

3.4. Translation-Based Decoder Module

KG reasoning tasks use a scoring function to measure the plausibility of a given triple (s, r, o), a decoder is used to simulate the conditional probability to obtain the probability score of a candidate triple. We choose ConvTransE as the decoder model. It contains a one-dimensional convolutional layer and a fully connected layer. We use ConvTransE$(\cdot)$ to represent these two layers. The probability vectors of all entities is:

$$\vec{p}(\, o \mid H_t, \mathcal{R}_t, s, r\,) = \sigma\left(H_t \, \text{ConvTrans}\,E\left(\vec{s}_t, \vec{r}_t\right)\right) \tag{10}$$

In the same way, the probability vector of all the relations is:

$$\vec{p}(r \mid H_t, \mathcal{R}_t, s, o) = \sigma\left(\mathcal{R}_t \operatorname{ConvTrans} E\left(\vec{s_t}, \vec{o_t}\right)\right) \quad (11)$$

where $\sigma(\cdot)$ denotes Sigmoid function, $\vec{s_t}, \vec{r_t}, \vec{o_t}$ denote the vector representation of subject entity s, relation r and object entity o in H_t and $\mathcal{R}_t$ at timestamp t, respectively.

3.5. Parameter Learning

We consider entity reasoning and relation reasoning as multi-label, multi-task learning problems. Denote by $y_{t+1}^e \in \mathbb{R}^{\mathcal{E}}$, $y_{t+1}^r \in \mathbb{R}^{\mathcal{R}}$ the label vector representation of the event entity reasoning task and the relation reasoning task at timestamp $t+1$. Thus, the total loss contains the entity reasoning loss L^e and the relation reasoning loss L^r is formalized as:

$$L^e = \sum_{t=0}^{T-1} \sum_{(s,r,o,t)\in F_t} \sum_{i=0}^{|\mathcal{E}|-1} y_{t+1,i}^e \log p_i(o \mid H_t, \mathcal{R}_t, s, r)) \quad (12)$$

$$L^r = \sum_{t=0}^{T-1} \sum_{(s,r,o,t)\in F_t} \sum_{i=0}^{|\mathcal{R}|-1} y_{t+1,i}^r log\ p_i(r \mid H_t, \mathcal{R}_t, s, o) \quad (13)$$

$$L = \lambda_1 L^e + \lambda_2 L^r \quad (14)$$

where T denotes the total number of timestamps in the training dataset; $y_{t+1,i}^e$ and $y_{t+1,i}^r$ represent the i^{th} vector element of y_{t+1}^e and y_{t+1}^r; $\log p_i(o \mid H_t, \mathcal{R}_t, s, r)$ and $\log\ p_i(r \mid H_t, \mathcal{R}_t, s, o)$ are the probability scores of entity i and relation i.

4. Experiments

4.1. Setup

4.1.1. Datasets

There are four typical TKGs commonly used in previous studies, ICEWS14, ICEWS18, WIKI, and YAGO. ICEWS14 and ICEWS18 are from the Integrated Crisis Early Warning System[18](ICEWS). We evaluated STSP on all these datasets. We preprocessed the four datasets for the extrapolation reasoning task as per previous works: we divided them into training, valid and test sets by timestamps. The timestamp ratio is 80%, 10%, and 10%. The datasets details are shown in Table 1.

Table 1. Statistics of the datasets. $|E_{train}|$, $|E_{valid}|$ and $|E_{test}|$ are the numbers of facts in training, valid, and test sets. Time intervals represent the temporal granularity between temporally neighboring facts.

Datasets	$\|E\|$	$\|R\|$	$\|E_{train}\|$	$\|E_{valid}\|$	$\|E_{test}\|$	*|Time interval|*
ICEWS14	7128	230	368,868	46,302	46,159	24hours
ICEWS18	23,033	256	373,018	45,995	49,545	24hours
WIKI	12,554	24	539,286	67,538	63,110	1year
YAGO	10,623	10	161,540	19,523	20,026	1year

4.1.2. Evaluation Metrics

We use two metrics widely used on TKGs reasoning tasks MRR and Hit@k to evaluate the performance of the model. MRR is the inverse mean of the ranking of real entity candidates for all queries and Hit@k denotes the proportion of times that real entity candidates appear in the top k positions of ranked candidates.

Many previous works use filtering settings during the valuation process. As described in[19], all event quartets that appear in the training set, valid set, or test set are removed from the ranking results, it is not suitable for the TKG extrapolation task[20]. Therefore, we use the experimental results under the original settings.

4.1.3. Baselines

STSP model is compared with three categories of models：for static KG reasoning and for TKG reasoning under the interpolation setting and extrapolation setting. The typical static models DistMult[21], R-GCN[7], ConvE[22] and RotaE[16] are selected with the temporal information of facts ignored. HyTE[12], TTransE[10] and TA-DistMult[11] are selected as the temporal models under the interpolation setting. The representative RE-NET[2], CyGNet [3], RE-GCN[4], rGalT[14] and RE-GAT[5] are selected as the temporal models under the extrapolation setting.

4.1.4. Implementation Details

For all datasets, the embedded dimension d is set to 200, the number of layers of the convolution-based relation-aware GCN is set to 2, and the dropout rate of each layer is set to 0.2. The best history length k for ICEWS14, ICEWS18, WIKI, and YAGO is 4,8,1,1. The best length m for sliding window is 7,6,3,4 for entity reasoning tasks and 5,4,3,5 for relation reasoning tasks. Static graph constraints have been added to ICEWS14 and ICEWS18. Adam is used for parameter learning, the learning rate is set to 0.001. For ConvTransE, the number of channels is set to 50, the kernel size is set to 2×3 and the dropout rate is set to 0.2. For the joint learning of the entity reasoning task and the relation reasoning task, λ_1 and λ_2 are experimentally set to 0.7 and 0.3.

4.2. Experimental Results

4.2.1. Results on Entity Reasoning

The results of the entity reasoning task are shown in Tables 2 and 3. STSP outperformed all baselines on the four benchmark datasets. The results convincingly verify its effectiveness. STSP significantly outperforms the static model because it captures some important historical temporal information. STSP works better than the temporal model in the interpolation setting, because STSP learns the sequential nature between timestamps, captures the temporal order pattern of things, and can get a more accurate representation of evolution for unlearned timestamps.

STSP achieves performance over all other temporal models in the extrapolation setting. RE-NET and CyGNet, which mainly focus on modeling all single-hop neighbors and repeated patterns, show stronger performance. STSP also outperforms them on all datasets because it captures structural correlations in KG better than the former; compared to the latter, STSP considers repeated fact patterns in histories in addition to the recent relation development on events, with more valid information.

rGalT only learns events sequentially by introducing Transformer-like structures into the inference task of temporal knowledge graphs. RE-GCN has a similar structure and effect to RE-GAT, but still does not learn events deeply in the short term so the effects of STSP are both improved under comparison.

4.2.2. Results on Relation Reasoning

Since some models are not designed for the relational reasoning task, we select models from the baseline that can be used for relational reasoning and present the experimental results in Table 4 based on MRR only. In more detail, we chose ConvE from the static model, RGCRN and RE-GCN from the temporal model. RE-NET and CyGNet are not used because they cannot be directly applied to the relation reasoning task. It can be observed that STSP outperforms all baselines. The performance gap between STSP and other baselines on the relation reasoning task is smaller than that on the entity reasoning task, because the number of relations is much smaller than the number of entities. The superior performance of STSP shows that our short-term sequential model can obtain a more accurate representation by capturing more of the relations that have a deeper impact in the near future after comprehensive modeling of the history. This validates the observations mentioned in the entity reasonings.

4.3. Auxiliary Experiments

4.3.1. Ablation Studies

To better understand the effectiveness of the STSP model components, we conducted an ablation study. As shown in Tables 5 and 6, -ST indicates that the short-term sequential mode is not used, -ATT indicates that the entity-aware attention module is not used, and -ST, -ATT indicates that both components are not used. The -ST mode only learns the relations sequentially and does not learn the evolution of the recent event relations, which leads to performance degradation and thus illustrates the effectiveness of this module. The -ATT mode directly uses the expression of the last timestamp as the input for the evolution of the current timestamp, and the process does not involve the entity-aware attention module, and the performance drops sharply, which fully illustrates the necessity of learning the entities under adjacent timestamps through entity-aware attention. From the above two aspects, the performance degradation in -ST, -ATT mode is inevitable, indicating that sequential learning of sequence information and using these two parts can largely improve the experimental performance, i.e., proving the effectiveness of the two parts, which is consistent with our observation of the results of entity reasoning and relation reasoning in the experimental results module.

Table 2. Performance for the entity reasoning task on ICEWS14 and ICEWS18 with raw metrics (in percentage).

Model	***ICEWS14***				***ICEWS18***			
	MRR	*H@1*	*H@3*	*H@10*	*MRR*	*H@1*	*H@3*	*H@10*
DistMult	20.32	6.13	27.59	46.61	13.86	5.61	15.22	31.26
R-GCN	28.03	19.42	31.95	44.83	15.05	8.13	16.49	29.00
ConvE	30.30	21.30	34.42	47.89	22.81	13.63	25.83	41.43
RotaE	25.71	16.41	29.01	45.16	14.53	6.47	15.78	31.86
HyTE	16.78	2.13	24.84	43.94	7.41	3.10	7.33	16.01
TTransE	12.86	3.14	15.72	33.65	8.44	1.85	8.95	22.38
TA-DistMult	26.22	16.83	29.72	45.23	16.42	8.60	18.13	32.51
RGCRN	33.31	24.08	36.55	51.54	23.46	14.24	26.62	41.96
RE-NET	35.77	25.99	40.41	54.87	26.17	16.43	29.89	44.37

CyGNet	34.68	25.35	38.88	53.16	24.98	15.54	28.58	43.54
RE-GCN	40.83	30.07	45.96	61.58	30.31	19.65	34.68	51.30
rGalT	38.33	28.57	42.86	58.13	27.88	18.01	31.59	47.02
RE-GAT	40.69	29.78	45.88	62.09	29.79	19.31	33.85	50.45
STSP	**41.38**	**30.62**	**46.96**	**62.19**	**30.93**	**20.30**	**35.20**	**51.84**

Table 3. Performance for the entity reasoning task on WIKI and YAGO with raw metrics (in percentage).

Model	***WIKI***			***YAGO***		
	MRR	*H@3*	*H@10*	*MRR*	*H@3*	*H@10*
DistMult	27.96	32.45	39.51	44.05	49.70	59.94
R-GCN	13.96	15.75	22.05	20.25	24.01	37.30
ConvE	26.03	30.51	39.18	41.22	47.03	59.90
RotaE	26.08	31.63	38.51	42.08	46.77	59.39
HyTE	25.40	29.16	37.54	14.42	39.73	46.98
TTransE	20.66	23.88	33.04	26.10	36.28	47.73
TA-DistMult	26.44	31.36	38.97	44.98	50.64	61.11
RGCRN	28.68	31.44	38.58	43.71	48.53	56.98
RE-NET	30.87	33.55	41.27	46.81	52.71	61.93
CyGNet	30.77	33.83	41.19	46.72	52.48	61.52
RE-GCN	51.09	57.60	68.82	62.89	71.02	79.98
rGalT	-	-	-	51.45	57.76	68.31
STSP	**52.26**	**59.17**	**69.80**	**63.66**	**72.18**	**83.53**

Table 4. Performance on the relation reasoning task with MRR.

Model	***ICEWS14***	***ICEWS18***	***WIKI***	***YAGO***
ConvE	38.80	37.73	78.23	91.33
R-GCN	38.04	37.14	88.88	90.18
RE-GCN	39.86	40.34	97.65	93.72
STSP	**40.98**	**40.92**	**98.16**	**93.99**

Table 5. Ablation studies on entity reasoning with MRR.

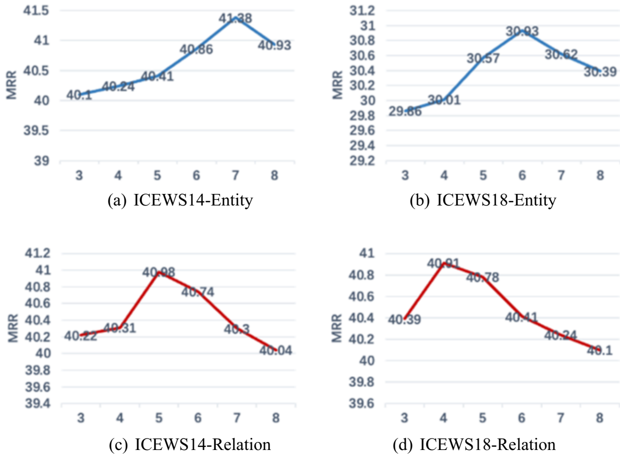
Model	***ICEWS14***	***ICEWS18***	***WIKI***	***YAGO***
STSP	**41.38**	**30.93**	**52.26**	**63.66**
-ST	40.53	27.97	51.88	63.43
-ATT	37.48	28.01	49.39	63.15
-ST,-ATT	34.68	27.15	31.01	48.55

Table 6. Ablation studies on relation reasoning with MRR.

Model	***ICEWS14***	***ICEWS18***	***WIKI***	***YAGO***
STSP	**40.98**	**40.92**	**98.16**	**93.99**
-ST	40.27	40.06	97.78	93.87
-ATT	39.01	39.89	87.26	93.76
-ST,-ATT	34.26	37.22	75.35	90.19

4.3.2. Influence of Sliding Window Length

To investigate the influence of sliding window length change on the reasoning results, we analyzed this factor. As shown in Figure 3, as the length of the sliding window changes, (a) and (b) denote the MRR results for ICEWS14 and ICEWS18 on the entity reasoning task, and (c) and (d) denote the MRR results for ICEWS14 and ICEWS18 on the relation reasoning task, which can be seen to change, further demonstrating the need to incorporate short-term relational change patterns in STSP. The experimental results show that entity reasoning and relation reasoning do not achieve the best performance at the same length setting. The entity reasoning performance is better when the length setting is 6-7, and the relation reasoning performance is better when the length setting is 3-4. This phenomenon suggests the necessity of learning recent historical facts, while entity reasoning requires indirect learning of relations from short-term sequential patterns, so the entity reasoning task needs more recent historical facts to help it.



(a) ICEWS14-Entity (b) ICEWS18-Entity

(c) ICEWS14-Relation (d) ICEWS18-Relation

Figure 3. Reasoning results for the influence of sliding window length.

5. Conclusion

In this paper, we propose a TKG reasoning model called STSP, which learns the representation of entities and relations by capturing features of historical facts. We combine an attention module that balances the influence of entities at different timestamps with a sliding window module that captures the common patterns within a segment of relations, and the final timestamped representations are reasoned through scoring functions using a decoder. Experimental results on four benchmark datasets demonstrate the significant advantages and effectiveness of STSP for both entity and relation aspects of the temporal reasoning task. Moreover, the ablation experiments show that these features of historical facts play an active role in TKG reasoning.

Acknowledgment

The work is supported by the National Natural Science Foundation of China under grant 62276011.

References

[1] Trivedi, Rakshit S. et al. “Know-Evolve: Deep Temporal Reasoning for Dynamic Knowledge Graphs.” *International Conference on Machine Learning* (2017).
[2] Jin, Woojeong et al. “Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs.” *Conference on Empirical Methods in Natural Language Processing* (2019).
[3] Zhu, Cunchao et al. “Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks.” *ArXiv* abs/2012.08492 (2020): n. pag.

[4] Li, Zixuan et al. "Temporal Knowledge Graph Reasoning Based on Evolutional Representation Learning." *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021): n. pag.
[5] Li, Zhipeng et al. "Future Event Prediction Based on Temporal Knowledge Graph Embedding." *Comput. Syst. Sci. Eng.* 44 (2022): 2411-2423.
[6] Bordes, Antoine et al. "Translating Embeddings for Modeling Multi-relational Data." *NIPS*(2013).
[7] Schlichtkrull, M. et al. "Modeling Relational Data with Graph Convolutional Networks." *Extended Semantic Web Conference* (2017).
[8] Ye, Rui et al. "A Vectorized Relational Graph Convolutional Network for Multi-Relational Network Alignment." *International Joint Conference on Artificial Intelligence* (2019).
[9] Vashishth, Shikhar et al. "Composition-based Multi-Relational Graph Convolutional Networks." *ArXiv* abs/1911.03082 (2019): n. pag.
[10] Leblay, Julien and Melisachew Wudage Chekol. "Deriving Validity Time in Knowledge Graph." *Companion Proceedings of the The Web Conference 2018* (2018): n. pag.
[11] García-Durán, Alberto et al. "Learning Sequence Encoders for Temporal Knowledge Graph Completion." *Conference on Empirical Methods in Natural Language Processing* (2018).
[12] Dasgupta, Shib Sankar et al. "HyTE: Hyperplane-based Temporally aware Knowledge Graph Embedding." *Conference on Empirical Methods in Natural Language Processing* (2018).
[13] Wang, Zhihao and Xin Li. "Hybrid-TE: Hybrid Translation-Based Temporal Knowledge Graph Embedding." *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)* (2019): 1446-1451.
[14] Gao, Yifu et al. "Modeling Precursors for Temporal Knowledge Graph Reasoning via Auto-encoder Structure." *International Joint Conference on Artificial Intelligence* (2022).
[15] Trivedi, Rakshit S. et al. "DyRep: Learning Representations over Dynamic Graphs." *International Conference on Learning Representations* (2019).
[16] Sun, Zhiqing et al. "RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space." *ArXiv* abs/1902.10197 (2018): n. pag.
[17] Kipf, Thomas and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks." *ArXiv* abs/1609.02907 (2016): n. pag.
[18] O'Brien, Sean P.. "Crisis Early Warning and Decision Support: Contemporary Approaches and Thoughts on Future Research." *International Studies Review* 12 (2010): 87-104.
[19] Ding, Zifeng et al. "Temporal Knowledge Graph Forecasting with Neural ODE." *ArXiv*abs/2101.05151 (2021): n. pag.
[20] Han, Zhen et al. "Explainable Subgraph Reasoning for Forecasting on Temporal Knowledge Graphs." *International Conference on Learning Representations* (2021).
[21] Yang, Bishan et al. "Embedding Entities and Relations for Learning and Inference in Knowledge Bases." *CoRR* abs/1412.6575 (2014): n. pag.
[22] Dettmers, Tim et al. "Convolutional 2D Knowledge Graph Embeddings." *AAAI Conference on Artificial Intelligence* (2017).