

Hawkes based Representation Learning for Reasoning over Scale-free Community-structured Temporal Knowledge Graphs

Yuwei Du, Xinyue Liu*, Wenxin Liang, Linlin Zong, Xianchao Zhang

School of Software, Dalian University of Technology, China

{xyliu, wxliang, llzong, xc Zhang}@dlut.edu.cn

duyuwei2023@mail.dlut.edu.cn

Abstract

Temporal knowledge graph (TKG) reasoning has become a hot topic due to its great value in many practical tasks. The key to TKG reasoning is modeling the structural information and evolutionary patterns of the TKGs. While great efforts have been devoted to TKG reasoning, the structural and evolutionary characteristics of real-world networks have not been considered. In the aspect of structure, real-world networks usually exhibit clear community structure and scale-free (long-tailed distribution) properties. In the aspect of evolution, the impact of an event decays with the time elapsing. In this paper, we propose a novel TKG reasoning model called Hawkes process-based Evolutional Representation Learning Network (HERLN), which learns structural information and evolutionary patterns of a TKG simultaneously, considering the characteristics of real-world networks: community structure, scale-free and temporal decaying. First, we find communities in the input TKG to make the encoding get more similar intra-community embeddings. Second, we design a Hawkes process-based relational graph convolutional network to cope with the event impact-decaying phenomenon. Third, we design a conditional decoding method to alleviate biases towards frequent entities caused by long-tailed distribution. Experimental results show that HERLN achieves significant improvements over the state-of-the-art models.

1 Introduction

Temporal Knowledge Graph (TKG) is a dynamic multi-relational graph used to record evolutionary events and knowledge in the real world. TKGs indicate facts as quadruples (subject, relation, object, time) and are actually sequences of temporal subgraphs divided by the time (timestamp) dimension. Reasoning over TKGs aims to infer the

missing quadruple facts, which has two settings: interpolation and extrapolation. Given a TKG with timestamps from t_0 to t_K , interpolation aims at inferring missing facts that occur at time t , where $t_0 < t < t_K$. Oppositely, extrapolation attempts to predict facts that occur at time t with $t > t_K$. In this paper, we focus on the extrapolation setting, which has gained much attention in recent years due to its great practical value in event prediction, question answering, and other areas (Wang et al., 2019b; Lan and Jiang, 2020). There are mainly two extrapolation tasks: entity prediction and relation prediction. We aim to propose a unified model that can accomplish both the entity and relation prediction tasks.

The key to TKG reasoning is modeling the structural information and evolutionary patterns of the TKGs. The prior extrapolation TKG reasoning models such as CyGNet (Zhu et al., 2021) and CENET (Yi et al., 2023), learn the evolutionary patterns by generating the historical event vocabulary to predict repetitive events. Later models such as RE-GCN (Li et al., 2021), HisMatch (Li et al., 2022) and HGLS (Zhang et al., 2023b), employ a relational graph convolutional network (RGCN) (Schlichtkrull et al., 2018) to capture the structural information from historical snapshots and use a recurrent neural network (RNN) to model the evolutionary patterns. Some recent works such as TITER (Sun et al., 2021) and DREAM (Zheng et al., 2023) introduce reinforcement learning on the TKG reasoning task.

Nevertheless, while TKGs are reflections of the real world, the structural and evolutionary characteristics of real-world networks have not been considered in previous models. In the aspect of structure, real-world networks (e.g., social-networks) usually exhibit clear community structure and scale-free (long-tailed distribution) properties (Barabási and Albert, 1999). In the aspect of evolution, the impact of an event decays with the time elapsing (Hawkes,

*Corresponding author

1971). Taking these characteristics into consideration not only can improve the reasoning performance, but also better facilitate the down-stream tasks.

In this paper, we propose a novel TKG reasoning model called Hawkes process-based Evolutional Representation Learning Network (HERLN), which learns structural information and evolutional patterns of a TKG simultaneously, considering the characteristics from real-world networks: community structure, scale-free and temporal decaying. Specifically, our model consists of three modules: an embedding initializing module, an evolution encoding module and a conditional decoding module.

In the embedding initializing module, to exploit the community structure properties of TKGs, we first find communities in the input TKG, and apply a graph convolution network to get embeddings of events within each community. The embeddings are then used as inputs in the evolution encoding module, which make the evolution encoding module output more similar intra-community embeddings. In the evolution encoding module, to cope with the event impact-decaying phenomenon, we design a Hawkes process-based relational graph convolutional network (HRGCN). The graph convolutional network contracts the structural information of the TKG, while the Hawkes process assigns different weights to the timestamps such that the impacts of events decay over time. In the conditional decoding module, to alleviate biases towards frequent entities caused by long-tailed distribution, we construct a conditional decoder which consists of a hyper network and a query-specific decoder. The hyper network adjusts the parameters according to the query events and the decoder generates conditional intensity scores for the candidate entities based on the adjusted parameters.

The main contributions of this work are summarized as follows:

- We recognize that the TKGs possess the structural and evolutional characteristics inherited from real-world networks: community structure, scale-free and temporal decaying, but they have not been considered or well exploited in previous studies.
- We propose a Hawkes process-based evolutional representation learning network (HERLN), which consists of three modules: (1) An embedding initialize module, which

extracts semantic information of community structure; (2) An evolution encoding module, which addresses the temporal decaying of event impact; (3) A conditional decode module, which alleviates the biases towards frequent entities caused by long-tailed distribution.

- Our proposed model HERLN can predict entities and relations at the same time. Experimental results on four benchmark TKG datasets show that HERLN achieves significant improvements over the state-of-the-art models.

2 Related work

2.1 Embedding-based methods

Embedding-based methods encode the whole or part of the TKGs to obtain the embeddings of entities and relations, and use the embeddings to evaluate the possibility of missing facts.

RE-NET (Jin et al., 2020) proposes an autoregressive architecture which uses a graph neural network (GNN) to capture local entity embeddings and a RNN to model interactions between entities over time. RE-GCN (Li et al., 2021) constructs a static graph to get the static attributes and presents a framework that can execute both entity and relation reasoning.

EvoKG (Park et al., 2022) uses an autoregressive architecture and captures the ever-changing structural and temporal dynamics via recurrent event modeling. HiSMATCH (Li et al., 2022) generates background graphs, entity-related graphs and relation-related graphs to jointly model the evolutional patterns. HGLS (Zhang et al., 2023b) models local snapshots or global graphs by using different GNNs and decodes them to get the predicting scores. CENET (Yi et al., 2023) combines historical and non-historical information and identifies highly related entities via contrastive learning. TARGAT (Xie et al., 2023) captures the interactions of multi-facts at different timestamps. DLGR (Xiao et al., 2024) learns the local and global perspective representations in a contrastive manner. DSTKG (Li et al., 2024) introduces two latent variables to capture the dynamic and static characteristics of entities in TKGs.

L^2 TKG (Zhang et al., 2023a) finds the missing relationships on the known KGs first and then reasons on the completed graph and the original graph jointly. RETIA (Liu et al., 2023) constructs a hyper-graph to connect different relations in a

high-dimensional space. These models use RNNs to represent the temporal information. Thus they are based on an assumption that the temporal sequences are equidistant, which is inconsistent with many real-life event sequences (Sun et al., 2022).

2.2 Path-based methods

Path-based methods find several related paths of query facts and select the most relevant one as the answer. TITer (Sun et al., 2021) adopts reinforcement learning to sample actions from query-related trajectories based on a time-shaped reward function. xERTE (Han et al., 2021a) samples and prunes the query-related subgraph according to query-dependent attention scores. TANGO (Han et al., 2021b) explores the neural ordinary differential equation to build a continuous-time model. TLogic (Liu et al., 2022) automatically mines recurrent temporal logic rules by extracting temporal random walks. DREAM (Zheng et al., 2023) use generative adversarial networks to design an adaptive reward function. However, the path-based methods focus on the local structure graph of the query, ignore the potential connection of events, and do not perform well on long-term reasoning.

In addition, it is worth noting that methods such as xERTE (Han et al., 2021a) and HISMatch (Li et al., 2022) consider the impact of temporal information on prediction results. They encode timestamps and concat timestamps with entity embeddings. However, different from our work, they actually learns that entities with different time intervals have different impacts on results, rather than considering the gradual decay of event impacts.

2.3 Hawkes process-based methods

The Hawkes process (Hawkes, 1971) is a stochastic process that models sequential discrete events occurring in continuous time. There are several works that combine the Hawkes process and neural networks for TKG reasoning. Know-Evolve (Trivedi et al., 2017) introduces a temporal point process to model facts evolved in the continuous time domain. GHNN (Han et al., 2020) proposes a graph Hawkes process to capture the potential temporal dependence across different timestamps. However, Know-Evolve and GHNN do not use the graph structural information. GHT (Sun et al., 2022) uses a temporal Transformer to capture long-term and short-term information jointly. However, none of the previous work has considered problem of temporal decaying of events' impacts. Our proposed

module uses the Hawkes process to assign different weights to the timestamps during message passing, thus the information of the event impact-decaying is encoded and utilized.

3 Problem Formulation

A temporal knowledge graph (TKG) $G = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F}\}$ is a directed multi-relational graph, where \mathcal{E} , \mathcal{R} , \mathcal{T} and \mathcal{F} denote the sets of entities, relations, timestamps and facts, respectively. A node in G represents an entity $i \in \mathcal{E}$, and an edge e_{ij} represents the interaction between node i and node j with relation $r \in \mathcal{R}$ at timestamp $t \in \mathcal{T}$. A fact in G is a quadruple $q = (s, r, o, t)$ that represents a real-world event consisting of the relation r between a subject entity s and an object entity o at timestamp t .

Given a TKG $G_{[t_1:t_k]} = \{\mathcal{E}, \mathcal{R}, \mathcal{T}, \mathcal{F} \mid \mathcal{T} = [t_1, t_k]\}$, the extrapolation reasoning task is to predict object o_q in a query like $(s_q, r_q, ?, t_q)$ where $t_q > t_k$, or predict relation r_q in a query like $(s_q, ?, o_q, t_q)$ where $t_q > t_k$.

4 Method

The proposed Hawkes process-based evolutionary representation learning network (HERLN) model is shown in Fig. 1, it consists of three modules, an embedding initializing module, an evolution encoding module and a conditional decoding module. The embedding initializing module detects communities in the input TKG and embeds the interaction frequencies between entities within each community into the initialized entity embeddings. With the initialized entity embeddings as input, the evolution encoding module updates the entity embeddings by learning the structural information from historical graph. The conditional decoding module uses representations of query quadruple to adjust the parameters and generate scores for candidates.

4.1 Embedding Initializing Module

To initialize the embeddings of entities, we first identify the communities, and then extract semantic information in the communities to get the initialized entity embedding matrix H_C .

4.1.1 Community Detection

The interactions between entities in real world exhibit distinct community structures. Exploiting of the community structural properties contributes to the improvement of reasoning performance. For

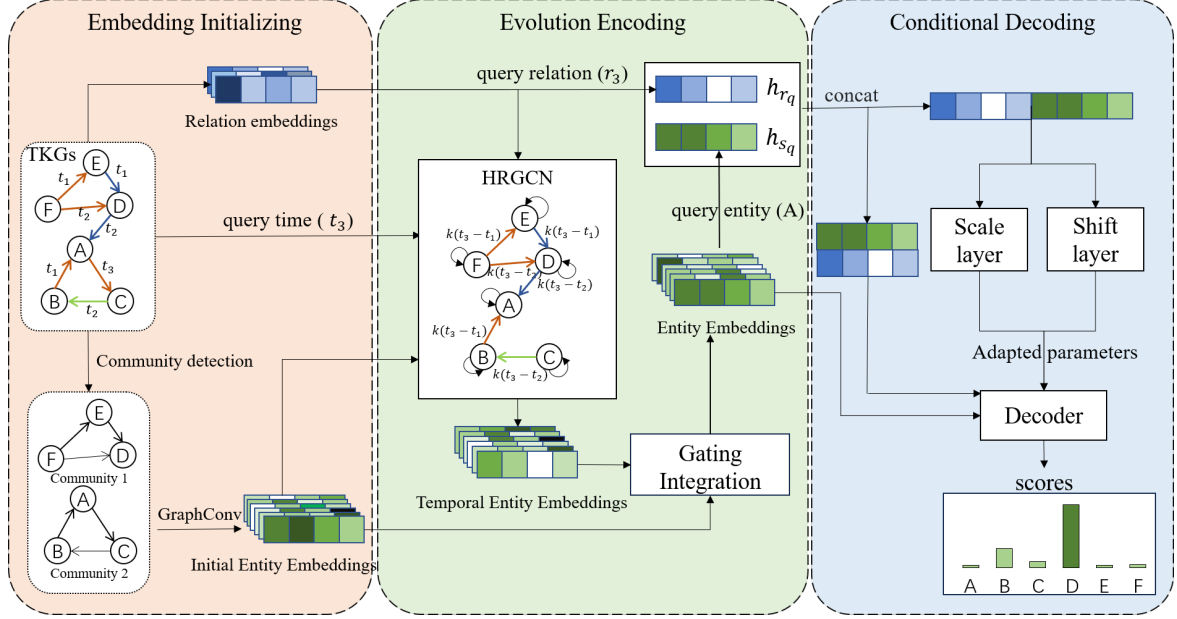


Figure 1: Overall framework of the proposed HERLN model. HERLN consists of three modules: an embedding initializing module, an evolution encoding module and a conditional decoding module. First, the embedding initializing module extracts the community structural information in input TKGs to get the initialized embeddings. Then the evolution encoding module updates the embeddings with the candidate-related historical structure to learn the evolutionary patterns of events. Finally, the conditional decoding module reasons according to the embeddings and gets scores of the candidate entities, then selects the entity with the highest score as the results. The input TKG contains 6 nodes (marked as A to F), three different types of edges (indicated in red, blue and green respectively) and three timestamps (from t_1 to t_3). Additionally, the model receives an incomplete quadruple as query. The query given in the figure is $(A, r_3, ?, t_3)$.

example, an entity that cooperates with a country like America is more likely to be a government or an organization rather than the citizens of a country. This information could be used to reduce the scores of entities that are not consistent with the facts.

We use a community detection algorithm on the entire TKG, which divides the entities set into different communities according to interaction frequencies, and obtains a graph that only contains inner-community links. The algorithm assigns each entity i to its community c_i , and there are a total of K communities in the TKG.

In TKGs, there are no inherent community labels. Therefore, we require an unsupervised and reliable method to detect possible communities in the TKGs. And since a TKG is a multi-relational graph, we extend the Louvain algorithm (Blondel et al., 2008) to handle with the multi-relational links. Specifically, we calculate modularity Q_r for different relation r by Eq. 1, which is an indicator that measures the quality of community detection.

$$Q_r = \sum_c \left[\frac{\sum in}{2m} - \left(\frac{\sum tot}{2m} \right)^2 \right] = \sum_c [e_c - a_c^2] \quad (1)$$

where $\sum in$ is the sum of weights of inner-community edges with relation r ; $\sum tot$ is the sum of weights of all the edges with relation r and m is the total weight of edges on the whole graph.

During the optimization process, when a community is merged into another community, the algorithm will calculate the modularity of the new entire graph, compare it with the modularity before merging to get ΔQ as described in Eq. 2.

$$\begin{aligned} \Delta Q &= \sum_r \Delta Q_r \\ &= \sum_r \left(\left[\frac{\sum in + k_{i,in}}{2m} - \left(\frac{\sum tot + k_i}{2m} \right)^2 \right] \right. \\ &\quad \left. - \left[\frac{\sum in}{2m} - \left(\frac{\sum tot}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \right) \\ &= \frac{1}{2m} \left(k_{i,in} - \frac{\sum_{tot} k_i}{m} \right) \end{aligned} \quad (2)$$

where $k_{i,in}$ is the sum of the edges' weights between node i and the new community in ; k_i is the

sum of the edges' weights between node i and all the nodes in the graph. The algorithm ends when ΔQ no longer changes.

4.1.2 Embedding Initialization

In order to import the information contained in the communities into the embeddings, we use a GCN to generate embeddings on the community subgraphs, which is formalized as:

$$h_i = \sigma \left(\sum_{j \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|} W h_j^{init} \delta(c_i, c_j) + W_0 h_i^{init} \right) \quad (3)$$

where h_i^{init} and h_j^{init} are randomly initialized embeddings of nodes i, j ; W is the parameter of message passing between nodes; W_0 is the parameter of self updating of a node; $\delta(c_i, c_j)$ is an indicator which is set to 1 if i and j belong to a same community and 0 otherwise; $\sigma(\cdot)$ is an activate function; \mathcal{N}_i is the set of neighbors of node i .

4.2 Evolution Encoding

After getting the initialized embedding matrix, the next move is to encode the candidate-related historical structure to learn the evolutionary patterns of events. In real-world events, the entity's status changes over time. Another common real-world phenomenon is that the impact of an event decays over time. The entity embeddings should be able to cope with the variations. This module achieves these points by updating the entity embeddings with historical information.

4.2.1 The Hawkes Process on TKGs

The Hawkes process is a stochastic process that models a sequential discrete events that occur chronologically, which is typically modeled by a conditional intensity function. The intensity function $\lambda(t)$ represents the probability that events happen at t , it is defined as follows:

$$\lambda_{(s,r,o)}(t) = \sum_{(s',r',o,t') \in \mathcal{F}_o} \gamma_{s',r'}(t') \kappa(t-t') + \mu_{s,r,o}(t) \quad (4)$$

where $\mu_{s,r,o}(t)$ is the base intensity at time t ; \mathcal{F}_o is the set of historical events related to node o ; $\gamma(\cdot)$ represents the amount of excitement induced by the corresponding events on results and $\kappa(\cdot)$ is a kernel function to model the effect of historical events on the current event.

To integrate the Hawkes Process into the TKG, we treat the Hawkes Process as an encoder-decoder network. For encoding, we use a function $En(\cdot)$ to get representations h_s, h_r and h_o derived from their historical neighbors' information, as shown in Eq. 5. For decoding, we transfer the representations into a certain intensity value with an appropriate decoding function $De(\cdot)$, as shown in Eq. 6.

$$h_s, h_r, h_o = En(G) \quad (5)$$

$$\lambda_{(s,r,o)}(t) = De(h_s, h_r, h_o) \quad (6)$$

We present detailed implementations of $En(\cdot)$ and $De(\cdot)$ in the following sections.

4.2.2 Encoding with Hawkes process-based RGCN

To get the historical representation of entities described in Eq.5, we design a Hawkes process-based relation graph convolutional network (HRGCN) as $En(\cdot)$ to pass message and update entity embeddings on the TKG.

As shown in Fig. 2, the improvement of HRGCN over traditional RGCN is that HRGCN can effectively learn the structural information from neighbors and assign weights of the messages which represents the temporal decaying of these messages, i.e.

$$h_o^t = \sigma \left(W_1 h_o + \sum_{s,r,t' \in \mathcal{F}_o^t} \frac{1}{|\mathcal{F}_o^t|} W_r (h_s + h_r) \tilde{\kappa}(t-t') \right) \quad (7)$$

where h_s, h_o and h_r are input embeddings of nodes s, o and edge r got from H_C and R , respectively; \mathcal{F}_o^t represents historical neighbors of node o , which contains all quadruples $= (s, r, o, t')$ where $t' < t$; W_r is the parameter of message passing between s and o ; W_1 is the parameter of self updating of node o ; $\tilde{\kappa}(t-t') = \frac{\kappa(t-t')}{\sum_{s,r,o,t'' \in \mathcal{F}_o^t} \kappa(t-t'')}$ is the temporal decaying function, where $\kappa(t-t') = \exp(-\delta(t-t'))$.

This module outputs the updated embedding matrix H_T^t that contains historical information.

4.2.3 Gating Integration

The learned embeddings by HRGCN get the historical information of entities, while some useful original information directly from the input TKG may be overwritten. We use a control gate to balance the contribution of the two kinds of information.

Specifically, we employ a fully connected layer to generate a graph embedding h_g according to the

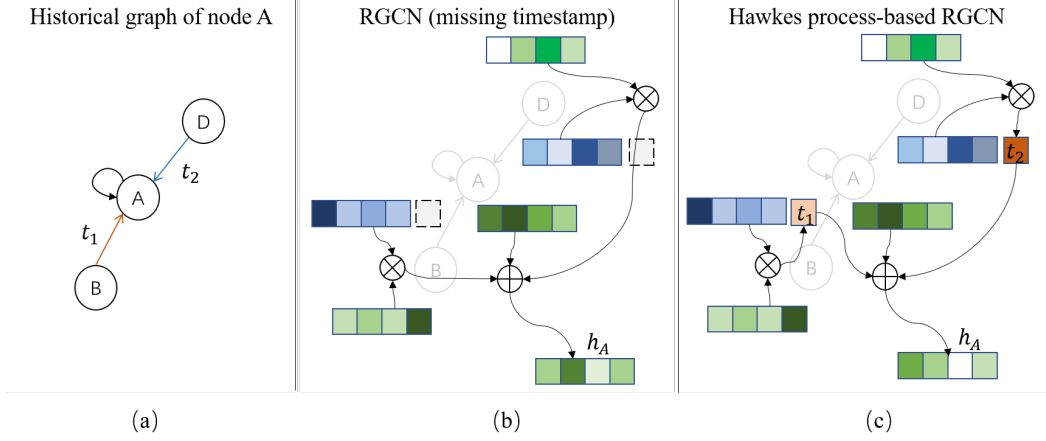


Figure 2: The update process of HRGCN. (a) is a historical structure of node A, (b) shows the update process of a traditional RGCN, which does not use timestamp information and (c) shows the update process of HRGCN, which takes the time interval as decaying weight to represent the event declines over time.

embeddings H_T^t from HRGCN. Then we use another fully connected layer to calculate the weight of H_T^t according to graph embedding h_g . The final embedding matrix is a weighted sum of H_T^t and the initialized embedding matrix H_C .

$$h_g = \sigma(W_{graph}H_T^t + b_{graph}) \quad (8)$$

$$\gamma = \sigma(W_{gate}h_g + b_{gate}) \quad (9)$$

$$H^t = \gamma * H_T^t + (1 - \gamma)H_C \quad (10)$$

where h_g is the generated graph embedding, W_{graph} , W_{gate} , b_{graph} and b_{gate} are learnable parameters.

The output of the encoding module is the balanced evolutionary embedding matrix H^t that encodes the evolutionary information on graph $G_{[t_1, t]}$.

4.3 Conditional Decoding

The last step of our model is to do reasoning according to the embedding matrix and get the conditional intensity scores of the candidate entities, then select the entity with the highest score as the result, as described in Eq.6.

4.3.1 Previous RGCN Decoding

Existing works (Li et al., 2021, 2022) use ConvTransE as the decoding function $De()$ for traditional RGCN to calculate the certain intensity values λ_o of entity o by Eq. 11.

$$\lambda_o^{ori} = \text{ConvTransE}(\text{concat}(h_{s_q}^{t_q}, h_{r_q}); h_{o_q}^{t_q}) \quad (11)$$

Note that the parameters of ConvTransE are fixed over different queries, which leads the model to tend to reason using the few evolutionary patterns

of the most common events, causing bias against other evolutionary patterns of the non-common events. We adjust ConvTransE to avoid the bias caused by long-tailed distribution in the following.

4.3.2 Avoiding Long-tailed Distribution Bias

We use feature linear modulation (FiLM) to construct a hyper-network, which adjusts parameters of decoder according to different queries, so that it can choose the appropriate query-specific evolutionary pattern for reasoning.

Specifically, given a query quadruple $= (s_q, r_q, o_q, t_q)$, the hyper-network generates a shifting factor $\alpha^{(s_q, r_q, t_q)}$ and a scaling factor $\beta^{(s_q, r_q, t_q)}$ according to the vector of the query quadruple to scale and shift the decoding parameters.

$$\alpha^{(s_q, r_q, t_q)} = \sigma((h_{s_q}^{t_q} || h_{r_q})W_\alpha + b_\alpha) \quad (12)$$

$$\beta^{(s_q, r_q, t_q)} = \sigma((h_{s_q}^{t_q} || h_{r_q})W_\beta + b_\beta) \quad (13)$$

$$\theta^q = (\alpha^{(s_q, r_q, t_q)} + 1) \odot \theta + \beta^{(s_q, r_q, t_q)} \quad (14)$$

where $h_{s_q}^{t_q}$ and h_{r_q} are embeddings of subject s_q at time t_q and relation r_q respectively; W_α , W_β , b_α and b_β are learnable parameters; θ is the original parameters from decoder and θ^q is query-specific parameters; \odot is Harmard product.

4.3.3 Adjusted Decoding

The adjusted decoder extracts the multi-dimensional features of the query quadruple through one-dimensional convolution and gets the conditional intensity scores of candidate entities via the inner product with the embedding matrix.

$$\lambda_o = \text{ConvTransE}([h_{s_q}^{t_q}, h_{r_q}]; h_{o_q}^{t_q}; \theta^q) \quad (15)$$

λ_o is the conditional intensity of candidate entity o ; $[\cdot]$ means the concat function; θ^q is the adjusted parameters got from the feature transform unit. The decoder chooses the entity which has the highest score as the missing part of the query quadruple.

4.4 Learning Objective

The task to predict the missing entity of a given quadruple could be seen as a multi-classification task and each entity in the candidate set belongs to one class. The optimization objective of entity prediction task is to maximum the scores of the ground truth entities, which can be convert to a cross-entropy loss \mathcal{L}_e :

$$\mathcal{L}_e = - \sum_{t_q \in \mathcal{T}} \sum_{\mathcal{F}_{t_q}} \sum_{k=1}^K y_k \log p(o_k | s_q, r_q, t_q) \quad (16)$$

where \mathcal{T} is the timestamp set; \mathcal{F}_{t_q} is the quadruple set with timestamp t_q ; K is the number of entities; $y_k = 1$ if entity o_k equals to ground truth o_q , otherwise 0; $p(o_k | s_q, r_q, t_q)$ is the probability of o_k , normalized by the scores of all the candidate entities in the candidate sets.

5 Experiments

5.1 Experimental Setup

5.1.1 Datasets

We use four benchmark datasets which are generally used in TKG reasoning task to evaluate the effectiveness of HERLN, ICEWS14 (Li et al., 2021), ICEWS18 (Li et al., 2021), WIKI (Jin et al., 2020) and YAGO (Jin et al., 2020). ICEWS is a database got from more than 100 data sources over more than 250 countries and regions. ICEWS14 and ICEWS18 datasets contain events occurred in 2014 and 2018 respectively. WIKI and YAGO are subsets of the Wikipedia history and YAGO3 respectively. We list the statistics of these datasets in Appendix A.

5.1.2 Baselines

We compare HERLN with 10 TKG reasoning models, which can be categorized into three classes. (1) Embedding-based models, RE-NET (Jin et al., 2020), REGCN (Li et al., 2021), EvoKG (Park et al., 2022), CENET (Yi et al., 2023), HGAT (Shao et al., 2023) and TiPNN (Dong et al., 2024); (2) Path-based models, TG-Tucker (Han et al., 2021b) and TLogic (Liu et al., 2022); (3) Hawkes process-based models, GHNN (Han et al., 2020) and GHT (Sun et al., 2022).

5.1.3 Evaluation Metrics

We report MRR, which is the mean of the reciprocal values of the actual missing entities' ranks averaged by all the queries, and Hits@1/3/10, i.e., the proportion of correct test cases that are ranked within top 1/3/10.

5.1.4 Implementation Details

We implement our model in Pytorch (Paszke et al., 2019) and DGL Library (Wang et al., 2019a). The experiments are conducted on a Nvidia GeForce Titan GPU. To be consistent with the baselines, we set the embedding dimension of entities d_e and relations d_r to 200. The number of HRGCN layers is set to 2 and the dropout rate for each layer is set to 0.2. We set all weights of edges to 1 in the embedding initializing module. We use the same hyperparameter settings of ConvTransE given by Li et al. (2021), the decode unit has 50 convolutional kernels with a size of 2×3 for each kernel. Adam (Kingma and Ba, 2015) is adopted for parameter learning with the learning rate of 0.001 on all the datasets. We report the average experimental results on three random seeds. Our code is available at <https://github.com/WisdomMLlab/HERLN>.

5.2 Experimental Results

5.2.1 Entity Prediction

Table 1 shows the entity prediction results on the benchmark datasets. The best results are marked in bold and the second best ones are underlined. It can be seen that our proposed HERLN performs the best nearly on all the settings, an only exception is that it achieves the second best Hits@3 score on ICEWS14. GHNN and GHT combines Hawkes point process with neural networks. But they do not consider the temporal decaying of event impact, which limits their performance. RE-GCN uses RNN on static graphs and loses temporal information during long-term evolution. CENET measures the probability of different entities by constructing the historical vocabulary. It does not exploit the graph structural information. TiPNN focuses on query-aware temporal path to capture the feature and doesn't take the potential relations between entities into account.

Our model can capture the historical structure and the event evolutionary patterns at the same time, considering the community structure, long-tailed distribution, and temporal decaying characteristics, thus it makes significant improvements over the baselines on all the datasets.

Method	ICEWS18				ICEWS14				WIKI				YAGO			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
GHNN	27.93	18.77	31.55	45.80	37.54	26.65	41.83	52.66	59.69	57.25	60.93	63.99	63.17	58.41	65.45	72.18
GHT	27.40	18.08	30.76	45.76	37.40	27.77	41.66	56.19	60.02	59.43	61.52	63.16	68.26	60.78	70.37	79.93
RE-NET	28.81	19.05	32.44	47.51	38.28	28.68	41.34	54.52	49.66	46.88	51.19	53.48	58.02	53.06	61.08	66.29
RE-GCN	30.55	20.00	34.73	<u>51.46</u>	41.50	30.86	46.60	<u>62.47</u>	51.53	-	58.29	69.53	63.07	-	71.17	82.07
TG-Tucker	28.68	19.35	32.17	47.04	26.25	17.30	29.07	44.18	50.43	48.52	51.47	53.58	57.83	53.05	60.78	65.85
TLogic	29.82	20.54	33.95	48.53	43.04	33.56	48.27	61.23	-	-	-	-	-	-	-	-
EvoKG	27.20	17.61	31.14	45.82	35.78	26.82	39.75	52.90	67.68	55.02	<u>79.48</u>	<u>84.03</u>	68.87	54.47	81.22	<u>89.81</u>
CENET	27.75	18.92	32.08	46.19	38.24	28.82	42.14	56.82	62.67	59.18	65.31	67.90	65.15	56.96	68.13	70.35
HGAT	28.50	19.60	32.70	46.60	38.90	29.70	42.40	56.40	56.10	52.90	58.10	61.80	63.60	59.80	66.00	71.50
TIPNN	30.32	21.55	35.06	50.08	41.20	32.75	46.23	59.60	73.99	<u>71.57</u>	76.82	80.67	<u>80.30</u>	<u>78.85</u>	<u>82.10</u>	89.04
OurModel	31.33	21.93	35.59	52.01	43.94	34.62	49.48	63.44	79.10	74.92	82.47	85.31	84.47	80.31	87.56	91.06

Table 1: Entity prediction results. The best results are marked in bold and the second best ones are underlined.

And among these four datasets, our method exhibits significant improvements compared to the baseline on the WIKI and YAGO dataset, while the enhancement on the ICEWS dataset is not as pronounced. This discrepancy is due to the fact that the WIKI and YAGO data both have a temporal span of one year, whereas ICEWS has a span of 24 hours. This aligns with real-world phenomena because the impact of events diminishes more noticeably over a longer time span, and the decay of short-term events is limited. Consequently, HERLN demonstrates substantial improvement in TKGs with longer time spans and distinct community structures.

5.2.2 Relation Prediction

The results of the relation prediction task in terms of the MRR metric are shown in Table 2. In the relation prediction task experiment, we do not include TG and RE-NET as baseline because they conduct only entity prediction. It can be seen from Table 2 that our proposed HERLN outperforms all the baselines and receives a boost of up to 10% in the MRR metric. The reasons of the performances of both our model and the baselines are similar to those for the entity prediction task.

Method	ICEWS18	ICEWS14s	WIKI	YAGO
ConvE	37.73	38.80	78.23	91.33
ConvTransE	38.00	38.40	86.64	90.98
R-GCRN	37.14	38.04	88.88	90.18
RE-GCN	40.53	41.06	<u>97.92</u>	<u>97.74</u>
EvoKG	<u>41.11</u>	<u>42.47</u>	90.63	90.26
CENET	38.24	40.50	87.51	91.44
OurModel	51.47	50.55	98.50	98.54

Table 2: Relation prediction results by MRR metric.

5.2.3 Ablation Study

We conduct ablation experiments on the ICEWS14 dataset. (1) OurModel without (w.o) ConvTransE: we replace the ConvTransE in the decoder unit

with a simple MLP layer. (2) OurModel without (w.o) FiLM: Instead of FiLM, we directly use ConvTransE. (3) OurModel without (w.o) HRGCN: we replace HRGCN with a RGCN to aggregate snapshots. (4) OurModel without (w.o) Community: we omit the embedding initialize module.

As shown in Table 3, replacing any component of our model degrades the performance, which demonstrates that each component of the model has a positive gain on the result. The variant (1) has an almost 6% drop in MRR, suggesting that an appropriate decoder can learn event evolution patterns effectively. The variant (2) shows 2% decreasing in Hits@1, indicating that the hypernetwork helps the model distinguish different event evolution patterns. The variant (3) has 7.89%, 6.89%, 9.31%, 11.90% drops in MRR, Hits@1, Hits@3, respectively, emphasizing the importance of incorporating the temporal information of events through HRGCN. The variant (4) has 3% decreasing in MRR, confirming the helpfulness of extracting community structures.

Method	MRR	Hits@1	Hits@3	Hits@10
w.o ConvTransE	37.98	29.59	41.72	54.19
w.o FiLM	41.48	31.39	46.47	60.14
w.o HRGCN	36.05	27.73	40.17	51.54
w.o Community	41.05	31.86	45.96	58.25
OurModel	43.94	34.62	49.48	63.44

Table 3: Effect of main components

6 Conclusions

In this paper, we propose a Hawkes-based evolutionary representation learning network (HERLN) to tackle the TKG reasoning tasks. We exploit the structural and evolutionary characteristics of TKGs inherited from real-world networks to learn structural information and evolutionary patterns. We initialize the embeddings with a community detection algorithm and a graph convolution network to make use of community structure. We design a Hawkes

process-based relational graph convolutional network to tackle the temporal decaying phenomenon. We construct a conditional decoder to alleviate the biases caused by scale-free property (long-tailed distribution). Experimental results show the superiority of our proposed model.

7 Limitations

Note that there are many entities appear only in the test set (named unseen entities), which will continue to appear as the size of TKG increases. Our model could not get sufficient information to assign proper embeddings for these entities. For further work, we plan to find a method to deal with the unseen entities.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No. 62476040) and the Fundamental Research Funds for the Central Universities.

References

- Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science*, 286:509–512.
- Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008.
- Hao Dong, Pengyang Wang, Meng Xiao, Zhiyuan Ning, Pengfei Wang, and Yuanchun Zhou. 2024. Temporal inductive path neural network for temporal knowledge graph reasoning. *Artificial Intelligence*, 329:104085.
- Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2021a. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *ICLR 2021, 9th International Conference on Learning Representations*, Virtual Event, Austria. OpenReview.net.
- Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021b. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *EMNLP 2021, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364, Virtual Event / Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020. Graph hawkes neural network for forecasting on temporal knowledge graphs. In *AKBC 2020, Conference on Automated Knowledge Base Construction*, Virtual. OpenReview.net.
- Alan G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58:83–90.
- Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *EMNLP 2020, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6669–6683, Online. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015, 3rd International Conference on Learning Representations*, San Diego, CA. OpenReview.net.
- Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 969–974, Online. Association for Computational Linguistics.
- Pengfei Li, Guangyou Zhou, Zhiwen Xie, Penghui Xie, and Jimmy Xiangji Huang. 2024. Learning dynamic and static representations for extrapolation-based temporal knowledge graph reasoning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:4741–4754.
- Zixuan Li, Zhongni Hou, Saiping Guan, Xiaolong Jin, Weihua Peng, Long Bai, Yajuan Lyu, Wei Li, Jiafeng Guo, and Xueqi Cheng. 2022. Hismatch: Historical structure matching based temporal knowledge graph reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7328–7338, Abu Dhabi, United Arab Emirate. Association for Computational Linguistics.
- Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutionary representation learning. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 408–417, Virtual Event, Canada. ACM.
- Kangzheng Liu, Feng Zhao, Guandong Xu, Xianzhi Wang, and Hai Jin. 2023. RETIA: relation-entity twin-interact aggregation for temporal knowledge graph extrapolation. In *39th IEEE International Conference on Data Engineering, ICDE 2023*, pages 1761–1774, Anaheim, CA, USA. IEEE.
- Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *AAAI 2022, Thirty-Sixth AAAI Conference on Artificial Intelligence*, pages 4120–4127, Virtual Event. AAAI Press.

- Namyong Park, Fuchen Liu, Purvanshi Mehta, Dana Cristofor, Christos Faloutsos, and Yuxiao Dong. 2022. Evokg: Jointly modeling event time and network structure for reasoning over temporal knowledge graphs. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining*, pages 794–803, Virtual Event / Tempe, AZ, USA. ACM.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.
- M. Schlichtkrull, Thomas Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC 2018, The Semantic Web - 15th International Conference*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607, Heraklion, Crete, Greece. Springer.
- Pengpeng Shao, Jiayi He, Guanjun Li, Dawei Zhang, and Jianhua Tao. 2023. Hierarchical graph attention network for temporal knowledge graph reasoning. *Neurocomputing*, 550:126390.
- Haohai Sun, Shangyi Geng, Jialun Zhong, Han Hu, and Kun He. 2022. Graph hawkes transformer for extrapolated reasoning on temporal knowledge graphs. In *EMNLP 2022: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7481–7493, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Haohai Sun, Jialu Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. In *EMNLP 2021, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8306–8319, Virtual Event / Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Rakshit S. Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *ICML 2017, Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3462–3471, Sydney, NSW, Australia. PMLR.
- Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yujie Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Haotong Zhang, Haibin Lin, J. Zhao, Jinyang Li, Alex Smola, and Zheng Zhang. 2019a. Deep graph library: towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*.
- Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019b. KGAT: knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019*, pages 950–958, Anchorage, Alaska, USA. ACM.
- Yao Xiao, Guangyou Zhou, Zhiwen Xie, Jin Liu, and Jimmy Xiangji Huang. 2024. Learning dual disentangled representation with self-supervision for temporal knowledge graph reasoning. *Inf. Process. Manage.*, 61(3).
- Zhiwen Xie, Runjie Zhu, Jin Liu, Guangyou Zhou, and Jimmy Xiangji Huang. 2023. Targat: A time-aware relational graph attention model for temporal knowledge graph embedding. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 31:2246–2258.
- Xu Yi, Ou Junjie, Xu Hui, and Fu Luoyi. 2023. Temporal knowledge graph reasoning with historical contrastive learning. In *AAAI*, pages 4765–4773, Washington, DC, USA. AAAI Press.
- Mengqi Zhang, Yuwei Xia, Qiang Liu, Shu Wu, and Liang Wang. 2023a. Learning latent relations for temporal knowledge graph reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023*, pages 12617–12631, Toronto, Canada. Association for Computational Linguistics.
- Mengqi Zhang, Yuwei Xia, Qiang Liu, Shu Wu, and Liang Wang. 2023b. Learning long- and short-term representations for temporal knowledge graph reasoning. In *WWW 2023, Proceedings of the ACM Web Conference 2023*, pages 2412–2422, Austin, TX, USA. ACM.
- Shangfei Zheng, Hongzhi Yin, Tong Chen, Quoc Viet Hung Nguyen, Wei Chen, and Lei Zhao. 2023. DREAM: adaptive reinforcement learning based on attention mechanism for temporal knowledge graph reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023*, pages 1578–1588, Taipei, Taiwan. ACM.
- Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhan. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *AAAI 2021, Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 4732–4740, Virtual Event. AAAI Press.

Appendix

A Statistics of the Datasets

The statistics of the datasets used in our experiment are summarized in Table 4. We divide

ICEWS14, ICEWS18, WIKI and YAGO into training, validation and test sets with a proportion of 80%, 10% and 10% in the chronological order, i.e., $t_{train} < t_{valid} < t_{test}$.

B Case Study

In Table 5, we present three cases obtained from the ICEWS14s test set, each of which validates the effectiveness of one contribution within HERLN. In all three cases, HERLN achieves the highest score and successfully completes the predictions. In the first case, two related events share the same subject entity and relation, with a different object entity. HERLN leverages the community structure of Criminal (Venezuela) and Citizen (Venezuela) to complete the reasoning. In the second case, multiple events occur between three entities, but the influence of earlier events on the reasoning outcome 10 months later is evidently smaller than that of more recent events. The Hawkes process helps our model to distinguish the useful information and make the correct choice. In the last case, *Demand settling of dispute* is a rare relation type, occurring less than 10 times in the dataset. The model is prone to be influenced by more common evolution patterns during reasoning. HERLN still successfully learns this evolution pattern by constructing a hyper network, completing the reasoning.

C Time Cost Analysis

To evaluate the efficiency of our model, we compare HERLN with several TKG reasoning methods including RE-NET, RE-GCN and EvoKG. We select these baseline methods for the time analysis because these methods are similar to ours, being embedding-based methods. Comparing the time efficiency on these methods can better illustrate the efficiency improvement of our approach. CENET is an exception as it does not provide official code. Although we conduct the experiments, the computational time is significantly longer than other methods (exceeding 1 day on ICEWS14s). Therefore, we exclude CENET from the results. As Fig. 3 shows, our model runs faster than RE-NET and has a similar time consumption with RE-GCN. On the one hand, RE-NET uses multiple RNN structures to fit multi-level conditional probability distributions of events while ours relies on a Hawkes process for message passing and aggregation on one single graph. On the other hand, we use query-specific version of ConvTransE as the decoder. The

structure of ConvTransE allows it to predict multiple events at the same time, this high parallelism saves much time.

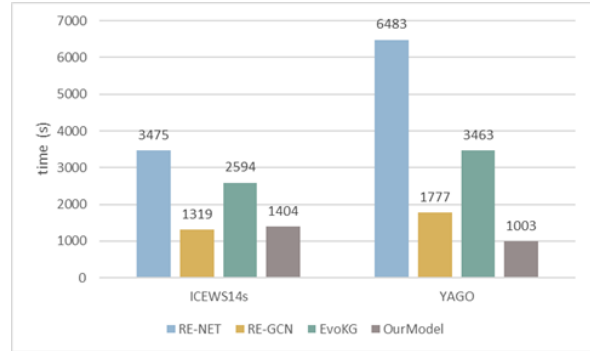


Figure 3: Time cost analysis on ICEWS14s and YAGO.

Dataset	\mathcal{E}	\mathcal{R}	\mathcal{F}	\mathcal{T}	\mathcal{F}_{train}	\mathcal{F}_{valid}	\mathcal{F}_{test}	Time interval
ICEWS18	23033	256	468558	304	373019	45996	49546	24 hours
ICEWS14	7128	230	90730	365	74846	8515	7372	24 hours
WIKI	12554	24	669934	231	539287	67539	63111	1 year
YAGO	10623	10	201089	187	161541	19524	20027	1 year

Table 4: Statistics of the datasets

Query	Answer	History
Criminal (Venezuela), fight with small arms and light weapons, ?, 2014/11/20	Citizen (Venezuela)	- Criminal (Venezuela), Use unconventional violence, Citizen (Venezuela), 2014/3/17 - Criminal (Venezuela), Use unconventional violence, Businessperson (Germany), 2014/6/19 - Jason C. Hu, Praise or endorse, Lin Chia-lung, 2014/11/25 - Jason C. Hu, consult, Ma Ying Jeou, 2014/1/16
Jason C. Hu, Yield, ?, 2014/11/29	Lin Chia-lung	- Jason C. Hu, consult, Ma Ying Jeou, 2014/1/16 - John Kerry, Make statement, Iran, 2014/11/9
John Kerry, Demand settling of dispute, ?, 2014/11/12	Iran	- John Kerry, Make statement, Iran, 2014/11/9

Table 5: Case Study