Tackling Long-Tail Entities for Temporal Knowledge Graph Completion

Mehrnoosh Mirtaheri
USC Information Sciences Institute
Los Angeles, CA, USA

Ryan A. Rossi
Adobe Research
San Jose, CA, USA

Sungchul Kim
Adobe Research
San Jose, CA, USA

Kanak Mahadik
Adobe Research
San Jose, CA, USA

Tong Yu
Adobe Research
San Jose, CA, USA

Xiang Chen
Adobe Research
San Jose, CA, USA

Mohammad Rostami
USC Information Sciences Institute
Los Angeles, CA, USA

ABSTRACT

Most Temporal Knowledge Graphs (TKGs) exhibit a long-tail entity distribution, where the majority of entities have sparse connections. Existing TKG completion methods struggle with managing new or unseen entities that often lack sufficient connections. In this paper, we introduce a model-agnostic enhancement layer that can be integrated with any existing TKG completion method to improve its performance. This enhancement layer employs a broader, global definition of entity similarity, transcending the limitations of local neighborhood proximity found in Graph Neural Network (GNN) based methods. Additionally, we conduct our evaluations in a novel, realistic setup that treats the TKG as a stream of evolving data. Evaluations on two benchmark datasets demonstrate that our framework surpasses existing methods in overall link prediction, inductive link prediction, and in addressing long-tail entities. Notably, our approach achieves a 10% improvement in MRR on one dataset and a 15% increase on another.

CCS CONCEPTS

• **Computing methodologies** → **Learning latent representations**; **Knowledge representation and reasoning**.

KEYWORDS

Temporal Knowledge Graph, Graph Representation Learning

ACM Reference Format:
Mehrnoosh Mirtaheri, Ryan A. Rossi, Sungchul Kim, Kanak Mahadik, Tong Yu, Xiang Chen, and Mohammad Rostami. 2024. Tackling Long-Tail Entities for Temporal Knowledge Graph Completion. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651565

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05
https://doi.org/10.1145/3589335.3651565

1 INTRODUCTION

Knowledge graphs (KGs) have become a fundamental tool for studying the underlying structure of multi-relational data in the real world [12]. KGs encapsulate factual information as a set of triplets, each consisting of a subject entity, a relation, and an object entity. This facilitates the analysis of complex relations and interactions within the data, establishing KGs as essential components that can enhance various other NLP tasks such as question answering, information retrieval, and LLMs [15]. However, KGs often grapple with a problem of incompleteness. Despite advancements in extraction methods, the occurrence of missing facts remains prevalent, significantly impacting downstream applications. As a result, the task of KG completion, i.e. predicting these missing facts has become of paramount importance [16].

Most real-world KGs are often derived from dynamic, ever-evolving data streams, such as daily news articles. New facts, regularly added to the data, may introduce new entities and relationships and are often accompanied by temporal information. In semantic KGs like Yago [9], facts are associated with time intervals, e.g., *(Obama, President, United States, 2009-2017)*. In contrast, Temporal event-centric knowledge graphs (TKGs) like ICEWS [1] link facts to specific timestamps, capturing precise interaction times. Thus, *(Obama, meet, Merkel)* in a TKG might recur multiple times between 2009 and 2017, reflecting distinct events. Therefore, event-centric KGs exhibit greater dynamism and present more complex challenges for completion.

TKG completion has recently emerged as a prominent research area, leading to the development of methods that derive temporal representations from historical interactions [7, 14, 17, 21]. Translation-based approaches, such as those introduced by [2, 6, 10], employ an embedding representation like a vector [6, 10] or a hyperplane [2, 18] to encode event timestamps and establish a function to transition an initial embedding to one that is time-aware. Others model the temporal information using shallow encoders [3, 20] or by employing sequential neural architectures [4, 5, 8, 11, 19, 21]. For example [8, 19] employs a recurrent structure to aggregate past timestamp entity neighborhoods. Traditional approaches typically initialize an embedding table for all entities [7, 19], which is then learned during training. However, for entities unseen during training, their lack of connections in the graph hinders the update of their embeddings, rendering them effectively untrained. While Graph Neural Network (GNN) based methods aim to infer embeddings for these unseen entities using their temporal neighborhoods, they often struggle when these neighborhoods are sparse. Furthermore, only a few works evaluate their methods under the inductive link prediction setup [4, 17], where edges during inference involve previously unseen entities. Even fewer, such as the method by [17], explicitly address these entities to derive robust representations.

To address the aforementioned shortcomings, we propose a model-agnostic enhancement layer that can be augmented to any TKG completion model and seeks to enhance the representations of entities with limited local connections. It leverages a broader, global definition of entity similarity, moving beyond mere local neighborhood proximity. Moreover, we present a more realistic incremental training and evaluation framework tailored for TKGs, effectively integrating unseen or sparsely connected entities into the model. For validation, we introduce two benchmark datasets designed for incremental learning, assessing our model across various scenarios. In addition to overall link prediction, we also assess our model's capability for inductive link prediction and its effectiveness in addressing long-tail entities.

2 TEMPORAL KNOWLEDGE GRAPH COMPLETION

In this section, we provide formal definitions for incremental TKG completion. A TKG $G = \langle \mathcal{Q}, \mathcal{E}, \mathcal{R} \rangle$ encapsulates a series of events as a set of quadruples $\mathcal{Q} = \{(s, r, o, \tau) | s, o \in \mathcal{E}, r \in \mathcal{R}\}$. Here, $\mathcal{E}$ and $\mathcal{R}$ represent the sets of entities and relations respectively, and τ denotes the timestamp of the event's occurrence. These events signify one-time interactions between entities at a specific point in time. The objective of TKG completion is to predict potential interactions between two entities at a given time, which can be achieved by predicting the object entity, given the subject and relation at a certain time, or by predicting the relation between two entities, given the subject and object at a specific time. For a given TKG with observations made up to time T, predictions can be made for events timestamped before T (interpolation) or beyond T (extrapolation). Our study focuses on extrapolation.

Streaming Temporal Knowledge Graphs. A TKG G can also be represented as a stream of graph snapshots $\langle G_1, G_2, \ldots, G_T \rangle$ arriving over time. Each snapshot $G_t = \langle \mathcal{Q}_t, \mathcal{E}_t, \mathcal{R}_t \rangle$ comprises $\mathcal{Q}_t = \{(s, r, o, \tau) | s, o \in \mathcal{E}_t, r \in \mathcal{R}_t, \tau \in [\tau_t, \tau_{t+1})\}$, which represents the set of quadruples occurring within the time interval $[\tau_t, \tau_{t+1})$. Here $\mathcal{E}_t$ and $\mathcal{R}_t$ denote the sets of entities and relations at time t respectively. Incremental training of a TKG completion method involves updating the model parameters $\mathcal{M}$ as new graph snapshots with a set of events becomes available over time. This process strives to retain previously learned information while assimilating new patterns. Formally, we define a set of tasks $\langle \mathcal{T}_1, \ldots, \mathcal{T}_T \rangle$, where each task $\mathcal{T}_t = \left(D_t^{train}, D_t^{test}, D_t^{val}\right)$ consists of disjoint subsets of the G_t events. The trained model $\mathcal{M}$ is thus represented as a stream of models $\mathcal{M} = \langle \mathcal{M}_1, \ldots, \mathcal{M}_T \rangle$, with corresponding parameter sets $\theta = \langle \theta_1, \theta_2, ..., \theta_T \rangle$, trained incrementally as a stream of tasks $\mathcal{T} = \langle \mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T \rangle$ arrives. Training starts on the model $\mathcal{M}$ using D_1^{train}, with D_1^{val} employed for hyper-parameter tuning. At each time step t, the model $\mathcal{M}_t$ with parameter set θ_t is initialized with parameters from the preceding time step θ_{t-1}. Subsequently, $\mathcal{M}_t$'s parameters are updated by training on D_t^{train}. Figure 1 depicts the training and evaluation setup for temporal extrapolation.

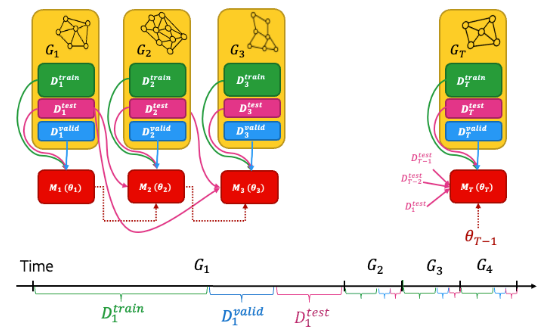

Figure 1: Incremental Extrapolation Setup. The arrow depicts the time. For extrapolation, the edge timestamps in the validation and test set do not have any overlap.

3 ENHANCEMENT LAYER

TKGs are dynamic structures in which entities and their relations evolve over time. A significant challenge in TKG completion is the long-tailed distribution of entities, characterized by many entities appearing infrequently and leading to sparse connections. This sparsity is further accentuated in growing TKGs, where the distribution of entities shifts over time, resulting in entities having dense connections at one time point and sparse connections at another. Such dynamics causes the model parameters to become biased towards entities with dense connections in recent times, and compromises the representation quality of sparsely connected entities. We propose a model-agnostic enhancement layer that extends beyond the local neighborhood of entities. This layer can be integrated with any TKG completion method. It enhances the representation of an entity by incorporating a temporal global representation from other similar entities, even if they aren't in the immediate vicinity. The enhanced entity representation e_s is formulated as:

$$e_s = \lambda f(s) + \phi(d_s)(1 - \lambda)g(s) \tag{1}$$

Where $\phi(d_s)$ is a decreasing function of d_s, which represents the degree of s, and is defined here as $\frac{1}{1+\exp(d_s)}$. The intuition behind this is to assign higher weights to nodes with lower degrees. Here, $f(s)$ denotes the embeddings of entity s from the underlying TKG completion model or any intermediate layer of a GNN-based method. The enhancement function, $g(s)$, derives a representation from a set of entities similar to s at time t, represented as $S_t(s)$. This introduces a general enhancement framework that provides flexibility in defining both the enhancement function and the criteria for entity similarity, which can vary depending on the application. In this work, we adopt a relation-based criterion for entity similarity. Two entities s_1 and s_2 are deemed similar if each has a connection of type r with any other entity. Formally, the set of similar entities at time t for a given entity s and relation r is:

$$S_t(s, r) = \{s_i | (s_i, r, o_i, t_i) \in G, t_i < t\} \tag{2}$$

The enhancement function, given a query $(s, r, ?, t)$, is defined as:

$$g(s, r, t) = \frac{\sum_{s_i \in S_t(s,r)} w_i e_{s_i}}{\sum w_i}, \quad w_i = \frac{1}{1 + \exp(t - t_i)} \tag{3}$$

For efficiency, we consider only a subset of $S_t(s, r)$, focusing on the most recent events and their corresponding entities.

4 EXPERIMENTS

Our experiments are focused on entity prediction for TKG completion. Through these experiments, our goal is to assess the effectiveness of the enhancement layer when it is integrated into a TKG completion model. We conduct evaluations for both inductive and general link prediction (LP) tasks. Furthermore, to better understand the impact of streaming data, we design our experiments to simulate data arriving incrementally. In the following sections, we detail the dataset construction, evaluation setup, base model, and performance comparison across various scenarios.

4.1 Evaluation Setup

This section provides detailed information on the dataset construction, training framework, and the baseline models.

Dataset Construction. We partition the quadruples of the TKG into sequential temporal snapshots, where each snapshot introduces new facts and potentially new entities. We use the Integrated Crisis Early Warning System (ICEWS) dataset, which records interactions among geopolitical actors with daily event timestamps. To create the evaluation benchmarks, we select a specific period from the ICEWS dataset and generate temporal snapshots by dividing this period into separate time intervals, thus partitioning the data into sets of quadruples with timestamps corresponding to each interval. The first snapshot contains 50% of the selected period, and the remaining data is segmented into non-overlapping periods of size w. Each snapshot is further divided into training, validation, and test sets based on their timestamps. In alignment with the extrapolation setup, the timestamps of the training set are earlier than those of the validation set, which are, in turn, earlier than those of the test set, as depicted in Figure 1. For both ICEWS18 and ICEWS14 the first seven months constitute the first snapshot, while subsequent snapshots each cover a span of seven days.

Training Procedure. As described in Section 2, we train the model incrementally using a stream of arriving data. As illustrated in Figure 1, in the extrapolation setup, the training, validation, and testing datasets are constructed to ensure there is no temporal overlap. Specifically, event timestamps in the training data are earlier than those in both the validation and test datasets. This design ensures that the model is trained on historical data and evaluated on future quadruples with unseen timestamps. However, this approach may omit data segments containing pivotal information for incremental learning. Therefore, at training step t, we maintain two checkpoints: one after training on the training set D_t^{train} for evaluation purposes, and another after additional training on both the validation and test sets for a few epochs, before proceeding to D_{t+1}^{train}.

Evaluation. For our base model, we employ TiTer [17], which is the state-of-the-art in inductive TKG completion. TiTer utilizes reinforcement learning to enable path-based reasoning. The enhancement layer (EL) is integrated with the TiTer model and fine-tuned according to the previously explained training procedure. As a baseline, we use the original TiTer model trained only on the first snapshot, in addition to the original TiTer model that is incrementally trained. Similar to other KG completion studies [12], we evaluate the models using the Mean Reciprocal Rank (MRR) and Hit@k metrics for $k = 1, 3, 10$. Following [13], we report the average model performance across all test data from previous training steps, as well as the current step t.

Table 1: Link prediction performance at the final step: current test dataset vs. average across all tests, comparing the base model (initial snapshot), Fine-Tuning (FT), and Enhancement Layer (EL)

		Current				Average			
Dataset	Model	H@10	H@3	H@1	MRR	H@10	H@3	H@1	MRR
ICEWS18	Titer	0.443	0.317	0.206	0.287	0.451	0.336	0.224	0.303
	+ FT	0.444	0.324	0.214	0.295	0.464	0.350	0.229	0.312
	+ EL (Ours)	**0.490**	**0.367**	**0.246**	**0.330**	**0.484**	**0.365**	**0.239**	**0.325**
ICEWS14	Titer	0.571	0.466	0.364	0.436	0.555	0.428	0.286	0.380
	+ FT	0.582	0.494	0.393	0.464	0.572	0.456	0.326	0.413
	+ EL (Ours)	**0.621**	**0.517**	0.393	**0.488**	**0.587**	**0.468**	**0.343**	**0.425**

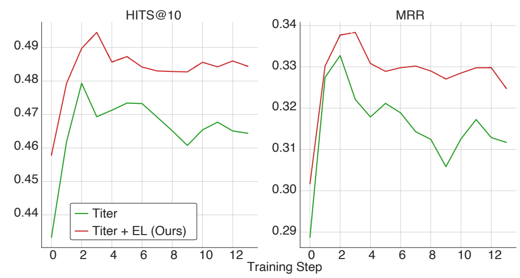

Figure 2: Titer performance over time for ICEWS18

4.2 Overall Performance

Our evaluation focuses on link prediction performance across models, with results summarized in Table 1. This table showcases the performance of the final incrementally trained model, $\mathcal{M}_T$, on the last snapshot and its average performance across all test datasets. When trained solely on the first snapshot, the base model, TiTer, demonstrated limited performance both for the latest snapshot and on average across all test datasets, highlighting the model's difficulty in effectively transferring knowledge. Fine-tuning (FT) offered a modest improvement over the model trained only on D_1^{train}. Incorporating the enhancement layer (EL) into TiTer led to superior performance over the base model, with more significant gains in MRR and Hit@10 metrics. Notably, our approach resulted in a 10% improvement in MRR for ICEWS14 and a 15% enhancement for ICEWS18, underscoring the effectiveness of our method in improving link prediction, especially within an incremental training framework. To further analyze the contribution of the enhancement layer for incremental training, we visualize the models' performances over time in Figure 2. Specifically, at time t, we compute

Table 2: Inductive Link prediction performance comparison. Performance is evaluated at the first test set.

	Model	H@10	H@3	H@1	MRR
ICEWS18	Titer	0.448	0.334	0.219	0.298
	Titer + EL (Ours)	**0.562**	**0.435**	**0.292**	**0.385**
ICEWS14	Titer	0.547	0.421	0.285	0.376
	Titer + EL (Ours)	**0.564**	**0.439**	**0.295**	**0.388**

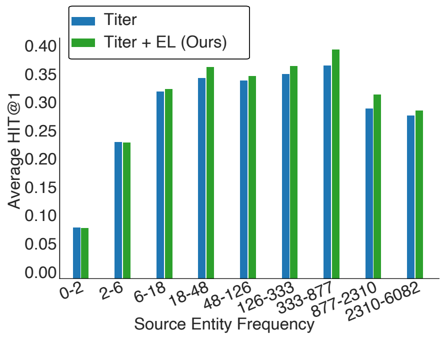

Figure 3: Hit@1 Performance over the union of test datasets grouped by source entity frequency

the average performance of $\mathcal{M}_t$ over the current test set and all preceding ones, i.e., $D_1^{test}, \dots, D_t^{test}$. We define the performance at time t as $P_t = \frac{1}{t}\sum_{j=1}^{t} p_{t,j}$, where $p_{t,j}$ represents the performance of $\mathcal{M}_t$ on D_j^{test}, measured by MRR and Hit@10. As illustrated in Figure 2, incorporating the enhancement layer into the base model significantly improves the model's performance over time.

4.3 Inductive Link Prediction

The inductive link prediction performance of various models, incrementally trained on the ICEWS18 and ICEWS14 benchmarks, is presented in Table 2. This evaluation assesses each model's ability to predict links involving entities not observed during training. We present results for the model $\mathcal{M}_1$, trained on D_1^{train} and evaluated against the corresponding test dataset D_1^{test}. This snapshot is particularly significant as it is the most extensive and contains the largest number of quadruples. Integrating our method with the original TiTer model yielded substantial improvements across all metrics. Additionally, we explore the enhancement layer's effectiveness in addressing long-tail entities. Figure 3 illustrates each model's performance across the union of test sets $D_1^{test}, \dots, D_T^{test}$, with test set quadruples aggregated by the incremental frequency of their source entity. This incremental frequency, which differs from overall frequency, reflects the entity's occurrence rate in the graph at the time the specific query quadruple was observed. The results indicate that for entities with frequencies below 18, none of the models exhibit a significant difference in the Hit@1 score. However, for frequencies above 18, our model consistently outperforms the fine-tuned variant, particularly for frequencies within the [18-48] range and above.

5 CONCLUSION

We address the challenge of sparse connections in TKGs by introducing a model-agnostic enhancement layer that enriches any TKG completion method. Adopting a global view of entity similarity significantly improves link prediction accuracy, especially for long-tail entities. Our evaluations, set in a realistic evolving data framework, demonstrate notable improvements in MRR for link prediction. Future research includes integrating richer features and continuously extracting facts from LLMs to enhance TKG representations.

REFERENCES

[1] E Boschee, J Lautenschlager, S O'Brien, S Shellman, J Starz, and M Ward. 2015. Integrated Crisis Early Warning System (ICEWS) Coded Event Data. *URL: https://dataverse. harvard. edu/dataverse/icews* (2015).
[2] Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. Hyte: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of EMNLP*. 2001–2011.
[3] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Dyernie: Dynamic evolution of riemannian manifold embeddings for temporal knowledge graph completion. *arXiv preprint arXiv:2011.03984* (2020).
[4] Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. xERTE: Explainable Reasoning on Temporal Knowledge Graphs for Forecasting Future Links. *arXiv preprint arXiv:2012.15537* (2020).
[5] Zhen Han, Yunpu Ma, Yuyi Wang, Stephan Günnemann, and Volker Tresp. 2020. Graph Hawkes Neural Network for Forecasting on Temporal Knowledge Graphs. In *Automated Knowledge Base Construction*.
[6] Prachi Jain, Sushant Rathi, Soumen Chakrabarti, et al. 2020. Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols. *arXiv preprint arXiv:2005.05035* (2020).
[7] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2019. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. *arXiv preprint arXiv:1904.05530* (2019).
[8] Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent Event Network: Autoregressive Structure Inference over Temporal Knowledge Graphs. In *Proceedings of EMNLP*. 6669–6683.
[9] Gjergji Kasneci, Maya Ramanath, Fabian Suchanek, and Gerhard Weikum. 2009. The YAGO-NAGA approach to knowledge discovery. *ACM SIGMOD Record* 37, 4 (2009), 41–47.
[10] Julien Leblay and Melisachew Wudage Chekol. 2018. Deriving validity time in knowledge graph. In *Companion Proc. of the The Web Conference*. 1771–1776.
[11] Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Virtual Event, Canada 1,2. 2021. Temporal Knowl-edge Graph Reasoning Based on Evolutional Representation Learning. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada* 1 (2021), 10. https://doi.org/10.1145/3404835.3462963
[12] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. *arXiv preprint arXiv:2212.05767* (2022).
[13] Mehrnoosh Mirtaheri, Mohammad Rostami, and Aram Galstyan. 2023. History Repeats: Overcoming Catastrophic Forgetting For Event-Centric Temporal Knowledge Graph Completion. *arXiv preprint arXiv:2305.18675* (2023).
[14] Mehrnoosh Mirtaheri, Mohammad Rostami, Xiang Ren, Fred Morstatter, and Aram Galstyan. 2020. One-shot learning for temporal knowledge graphs. *arXiv preprint arXiv:2010.12144* (2020).
[15] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2023. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *arXiv preprint arXiv:2306.08302* (2023).
[16] Tong Shen, Fu Zhang, and Jingwei Cheng. 2022. A comprehensive overview of knowledge graph completion. *Knowledge-Based Systems* (2022), 109597.
[17] Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. TimeTraveler: Reinforcement learning for temporal knowledge graph forecasting. *arXiv preprint arXiv:2109.04101* (2021).
[18] Zhihao Wang and Xin Li. 2019. Hybrid-TE: Hybrid Translation-Based Temporal Knowledge Graph Embedding. In *IEEE ICTAI*. IEEE, 1446–1451.
[19] Jiapeng Wu, Meng Cao, Jackie Chi Kit Cheung, and William L Hamilton. 2020. Temp: Temporal message passing for temporal knowledge graph completion. *arXiv preprint arXiv:2010.03526* (2020).
[20] Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2019. Temporal knowledge graph embedding model based on additive time series decomposition. *arXiv preprint arXiv:1911.07893* (2019).
[21] Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhan. 2020. Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks. *arXiv preprint arXiv:2012.08492* (2020).