Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks

Cunchao Zhu[1], **Muhao Chen**[2], **Changjun Fan**[1], **Guangquan Cheng**[1], **Yan Zhang**[3]

[1] College of Systems Engineering, National University of Defense Technology, China
[2] Viterbi School of Engineering, University of Southern California, USA
[3] École Pour l'Informatique et les Techniques Avancées, France
{zhucunchao19,fanchangjun,cgq299}@nudt.edu.cn, muhaoche@usc.edu, yan.zhang@epita.fr

Abstract

Large knowledge graphs often grow to store temporal facts that model the dynamic relations or interactions of entities along the timeline. Since such temporal knowledge graphs often suffer from incompleteness, it is important to develop time-aware representation learning models that help to infer the missing temporal facts. While the temporal facts are typically evolving, it is observed that many facts often show a repeated pattern along the timeline, such as economic crises and diplomatic activities. This observation indicates that a model could potentially learn much from the known facts appeared in history. To this end, we propose a new representation learning model for temporal knowledge graphs, namely `CyGNet`, based on a novel time-aware copy-generation mechanism. `CyGNet` is not only able to predict future facts from the whole entity vocabulary, but also capable of identifying facts with repetition and accordingly predicting such future facts with reference to the known facts in the past. We evaluate the proposed method on the knowledge graph completion task using five benchmark datasets. Extensive experiments demonstrate the effectiveness of `CyGNet` for predicting future facts with repetition as well as *de novo* fact prediction.

1 Introduction

Knowledge Graphs (KGs) are widely used resources for knowledge representations of real-world facts (or events), inasmuch as it supports countless knowledge-driven tasks in areas such as information retrieval (Liu et al. 2018), natural language understanding (Chen, Chen, and Yu 2019; He et al. 2017), recommender systems (Koren, Bell, and Volinsky 2009) and healthcare (Hao et al. 2020). Traditionally, a KG only possesses facts in a static snapshot, while currently the rapid growing data often exhibit complex temporal dynamics. This calls for new approaches to model such dynamics of facts by assigning the interactions of entities with temporal properties (i.e., known as *temporal knowledge graphs*, or TKGs.). Representative TKGs include Global Database of Events, Language, and Tone (GDELT) (Leetaru and Schrodt 2013) and Integrated Crisis Early Warning System (ICEWS)

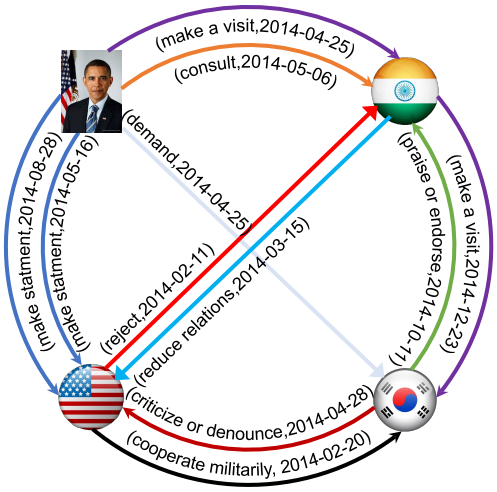

Figure 1: A snippet of ICEWS showing several records of diplomatic activities.

(Boschee et al. 2015), which are two well-known event-based data repository that store evolving knowledge about entity interactions across the globe. Figure 1 shows a snippet of ICEWS, which shows several records of diplomatic activities during different time.

Recently, many research efforts have been put into representation learning for TKGs. Relevant methods typically encode the temporally evolving facts of entity relations with time-specific embeddings. This provides a versatile and efficient tool to complete future facts of TKGs based on the embedding representations of the past facts. Moreover, it benefits a wide range of downstream applications, e.g. transaction recommendation (Lortscher Jr 2010), event process induction (Bouchard and Andreoli 2012; Du et al. 2016) and social relation prediction (Zhou et al. 2018a). Several temporal knowledge graph embedding (TKGE) methods only focus on calculating latent representations for each snapshot separately, and thus are not capable of capturing the long-term dependency of facts in consecutive time snapshots (Jiang et al. 2016; Dasgupta, Ray, and Talukdar 2018; Wang and Li 2019; Ma, Tresp, and Daxberger 2019; Lacroix, Obozinski, and Usunier 2019). Some other recent

attempts encode the temporally evolving facts of entity relations by incorporating past information from previous snapshots (Goel et al. 2020; Trivedi et al. 2017; Jin et al. 2019). However, the complex evolution of temporal facts, including repetition and trending patterns, can cause the aforementioned methods to fall short.

In fact, many facts occur repeatedly along the history. For instance, global economic crises happen periodically in about every seven to ten years (Korotayev and Tsirel 2010); the diplomatic activities take place regularly between two countries with established relationships (Feltham 2004); East African animals undergo annual vast migration in every June (Musiega and Kazadi 2004). More specifically, we found that over 80% of all the events throughout the 24 years of ICEWS data (i.e. 1995 to 2019) have already appeared during the previous time period. This phenomenon highlights the importance of leveraging the known facts in predicting the future ones. However, most existing methods do not incorporate the awareness of such evolution patterns in modeling a TKG. On the contrary, to conform with the nature of evolving facts in TKG events, we believe more precise temporal fact inference should make full use of known facts in history. Accordingly, a TKGE model would benefit from learning the temporal repetition patterns while characterizing the TKGs.

To this end, we propose a new representation learning method for TKGs based on a novel time-aware copy mechanism. The proposed CyGNet (Temporal **C**opy-**G**eneration **Net**work) is not only able to predict future facts from the whole entity vocabulary, but also capable of identifying facts with repetition, and accordingly selecting such facts based on the historical vocabulary of entities that form facts only appeared in the past. This behaves similarly to the copy mechanism in the abstractive summarization (Gu et al. 2016) in natural language generation (NLG), which allows a language generator to choose to copy subsequences from the source text, so as to help generate summaries that preserve salient information in the source text. Inspired by this mechanism, when predicting a future fact of quadruple $(s_i, p_j, ?, t_T)$, we can treat the known facts $\{(s_i, p_j, o_a, t_0), (s_i, p_j, o_b, t_0), ..., (s_i, p_j, o_k, t_{T-1})\}$ appeared in the previous snapshots as the source text in abstractive summarization, and predict the future facts based solely on the known facts from the historical vocabulary $\{o_a, o_b, ..., o_k\}$. As shown in Figure 2, CyGNet consists of two modes of inference, namely Copy mode and Generation mode. We illustrate CyGNet's two inference modes with an example prediction of the 2018 NBA championship. Accordingly, all 18 NBA champion teams before 2018 are collected as the historical entity vocabulary, and the total 30 NBA teams are considered as the whole entity vocabulary. To complete the query (*NBA*,*champion*,*?*,2018), CyGNet utilizes the Copy mode to predict the entity probabilities from the known entity vocabulary, and adopts the Generation mode to infer the entity probabilities from the whole entity vocabulary. Then, both probability distributions are combined as the final prediction.

We conduct extensive experiments on five benchmark TKG datasets, which demonstrate that CyGNet can effectively model TKG data via the combination of the Copy mode and the Generation mode in both learning and inference. Accordingly, CyGNet achieves more precise future fact predictions than state-of-the-art (SOTA) methods that do not take special consideration of temporal facts with repetition patterns.

The main contributions of this paper are as follows:

1. We investigate the underlying phenomena of temporal facts with repetition, and propose to make reference to known facts in history when learning to infer future facts in TKGs;
2. We propose a novel TKGE model CyGNet via the time-aware copy-generation mechanism, which combines two modes of inference to make predictions based on either the historical vocabulary or the whole entity vocabulary, hence being more consistent to the aforementioned evolution pattern of TKG facts;
3. We conduct extensive experiments on five public TKG benchmarks datasets, and demonstrate its superiority over previous SOTA TKG models on the task of future fact (link) prediction.

2 Related Work

We discuss three lines of relevant research. Each has a large body of work, of which we can only provide a highly selected summary.

Static KG Embeddings. A large number of methods are developped to model static KGs without temporally dynamic facts, which have been summarize in recent surveys (Wang et al. 2017; Ji et al. 2020; Dai et al. 2020). A class of these methods are the translational models (Bordes et al. 2013; Wang et al. 2014; Ji et al. 2015), which models a relation between two entity vectors as a translation vector. Another class of models are the semantic matching models that measures plausibility of facts using a triangular norm (Yang et al. 2015; Trouillon et al. 2016; Sun et al. 2019). Some other models are based on deep neural network approaches using feed-forward or convolutional layers on top of the embeddings (Schlichtkrull et al. 2018; Dettmers et al. 2018; Schlichtkrull et al. 2018). However, these methods do not capture temporally dynamic facts.

Temporal KG Embeddings. More recent attempts have been made to model the evolving facts in TKGs. TTransE (Jiang et al. 2016) is an extension of TransE (Bordes et al. 2013) by embedding temporal information into the score function. HyTE (Dasgupta, Ray, and Talukdar 2018) replaces the unit normal vector of the hyperplane projection in TransH (Wang et al. 2014) with a time-specific normal vector. Know-Evolve (Trivedi et al. 2017) learns non-linearly evolving entity representations over time which models the occurrence of a fact as a temporal point process. DE-SimplE (Goel et al. 2020) leverages diachronic embeddings to represent entities at different timestamps and employs the same score function as SimplE (Kazemi and Poole 2018) to score the plausibility of a quadruple. Based on the Tucker decomposition (Balazevic, Allen, and Hospedales 2019), ConT (Ma, Tresp, and Daxberger 2019) learns a new core tensor

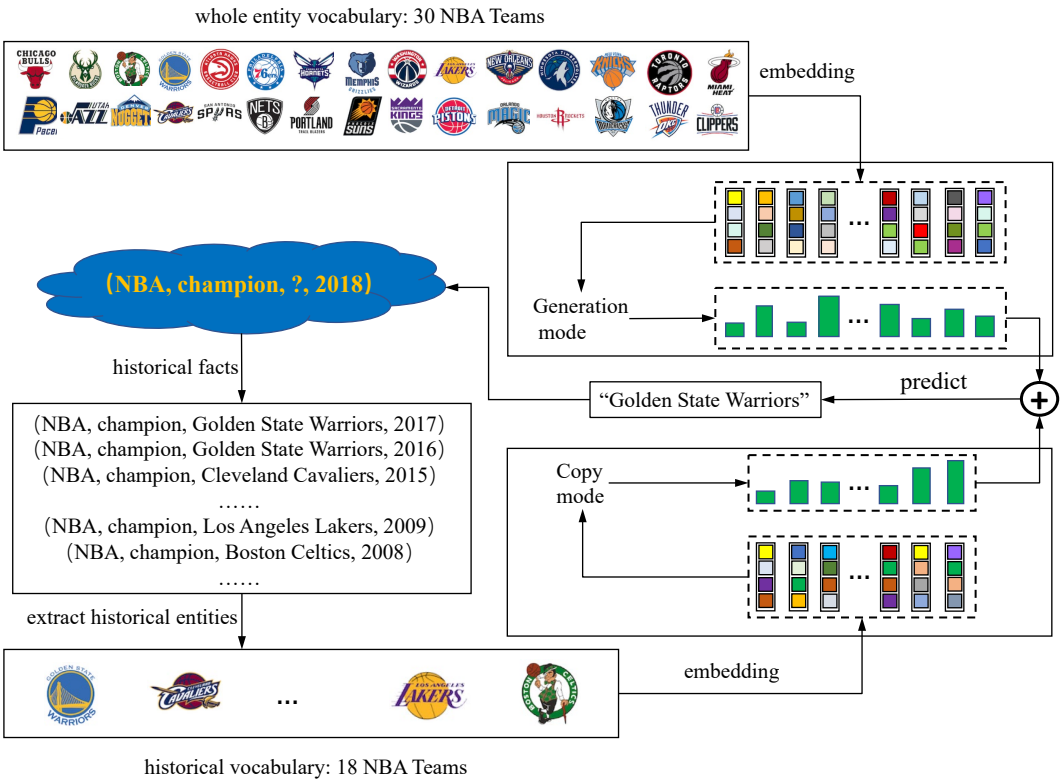

Figure 2: Illustration of CyGNet's copy-generation mechanism. In this figure, when predicting which team was the champion in NBA in 2018. CyGNet first obtains each entity's embedding vector (shown as a color bar). It then utilizes both inference modes to infer entity probabilities (shown as green bars, with heights proportional to the probability values) from the historical or the whole entity vocabulary. "*Golden State Warriors*" is predicted as the 2018 NBA champion by combining probabilities from both Copy and Generation modes.

for each timestamp. However, these models do not provide a mechanism to capture the long-term dependency of facts in consecutive time snapshots.

Some other methods are designed to model graph sequences, which can be applied to capture long-term dependency of TKG facts. TA-DistMult (García-Durán, Dumancic, and Niepert 2018) utilizes a recurrent neural network to learn time-aware representations of relations and uses standard scoring functions from DistMult (Yang et al. 2015). GCRN (Seo et al. 2018) combines GCN (Kipf and Welling 2017) for graph-structured data and RNN to identify simultaneously meaningful spatial structures and dynamic patterns. DyREP (Trivedi et al. 2018) divides the dynamic graph network into two separate processes of global and local topological evolution, and proposes a two-time scale deep temporal point process to jointly model the two processes. Know-Evolve, DyREP and GCRN have also been combined with MLP decoders to predict future facts, as presented by Jin et al. (2019). The previous SOTA method in this line of research, i.e. RE-NET (Jin et al. 2019) models event (fact) sequences jointly with an RNN-based event encoder and an RGCN-based (Schlichtkrull et al. 2018) snapshot graph encoder.

Copy Mechanism. The copy mechanism has been previously applied to NLG tasks, particularly for abstractive summarization. Vinyals, Fortunato, and Jaitly (2015) proposed the Pointer Networks in which a pointer mechanism is used to select the output sequence directly from the input sequence. However, the Pointer Network cannot make prediction using lexemes that are external to the input sequence. COPYNET (Gu et al. 2016) solves this problem in a hybrid end-to-end model by combining the pointer network (or termed as copy mechanism in its context) with a generation mechanism that yields lexemes that do not appear in the input. SeqCopyNet (Zhou et al. 2018b) improves the copy mechanism to not only copy single lexemes, but also copies subsequences from the input text. Our work is inspired to use the copy mechanism for the characteristics of TKGs, which to the best of our knowledge, is the first work that incorporates the copy mechanism into modeling TKGs. This is consistent to the nature that temporal knowledge may contain repetition patterns along the trending timeline.

3 Method

In this section, we introduce the proposed model, named CyGNet, for fact/link prediction in TKGs. We start with the notations, and then introduce the model architecture as well as its training and inference procedures in detail.

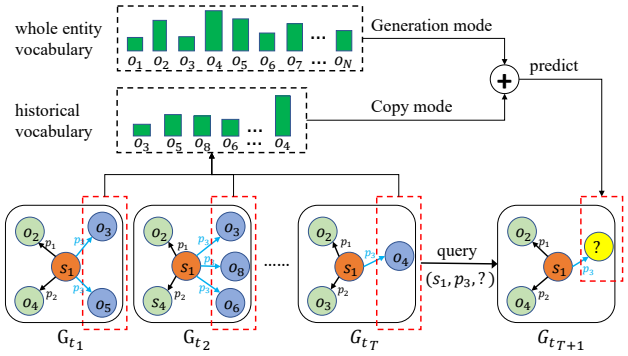

Figure 3: The overall architecture of CyGNet. Dark blue nodes are the candidate object entities in the historical vocabulary for query $(s_1, p_3, ?, t_{T+1})$. Green bars (with heights proportional) indicate the probability values predicted by the Copy mode and the Generation mode.

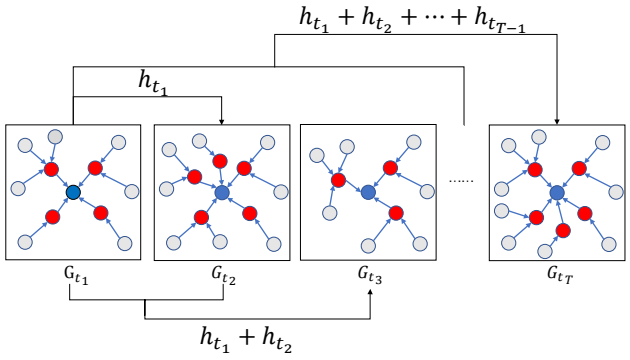

Figure 4: Illustration of the training process. Each time we train on a new TKG snapshot, we will extend the historical vocabulary based on that of the previous snapshot. h_{t_k} contains the historical vocabulary of all subject entity and predicate pair in the snapshot G_{t_k}.

3.1 Notations

TKGs incorporate temporal information in traditional KGs. In a TKG, each of the fact captures the relation (or predicate) $p \in \mathcal{R}$ for subject and object entity $s \in \mathcal{E}$ and $o \in \mathcal{E}$ at time step $t \in \mathcal{T}$, where $\mathcal{E}$, $\mathcal{R}$ denote the corresponding vocabularies of entities and relations respectively, and $\mathcal{T}$ is the set of timestamps. Boldfaced $\mathbf{s}$, $\mathbf{p}$, $\mathbf{o}$, $\mathbf{t}$ represent the embedding vectors of subject entity h, predicate p, object entity o and time step t in a temporal fact. $\mathcal{G}_t$ is a snapshot of the TKG at a time t, and $g = (s, p, o, t)$ denotes a quadruple (fact) in $\mathcal{G}_t$. A TKG is built upon a set of fact quadruples ordered ascendingly based on their timestamps, i.e., $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T\}$, where the same quadruple is removed for redundancy. For each subject entity and predicate pair at time step t, we define a delimited subset of E that is specific to (s, p, t) (namely a *historical vocabulary* for (s, p, t)) as $\mathbf{H}_{t_k}^{(s,p)}$, which contains all the entities that have served as object entities in facts with the subject entity s and the predicate p along the known snapshots $\mathcal{G}_{(t_1,t_{k-1})} = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_{k-1}\}$ before t_k, where the historical vocabulary $\mathbf{H}_{t_k}^{(s,p)}$ is an N-dimensional multi-hot indicator vector and N is the cardinality of $\mathcal{E}$, the value of entities in the historical vocabulary are marked 1 while others are 0. Prediction of a missing temporal fact aims to infer the missing object entity given $(s, p, ?, t)$ (or the subject entity given $(?, p, o, t)$, or the predicate with subject entity and object entity given $(s, ?, o, t)$). Without loss of generality, we describe our model as predicting the missing object entity in a temporal fact, although the model can be easily extended to predicting other elements including the subject entity and the predicate.

3.2 Model Components

As shown in Figure 3, our model combines two modes of inference, namely *Copy mode* and *Generation mode*, where the former seeks to select entities from a specific historical vocabulary that forms repeated facts in history while the latter predicts entities from the whole entity vocabulary. When predicting a quadruple $(s_1, p_3, ?, T + 1)$ (as shown in Figure 3), the Copy mode should infer the probability of entities in the historical vocabulary $\{s_3, s_4, s_5, ..., s_m\}$ which have served as the object entities in facts of the subject entity s_1 and the predicate p_3 along the known snapshots $\mathcal{G}_{(t_1,t_T)}$. On the other hand, the Generation mode estimates the probability of every entity in the whole entity vocabulary to answer a query. Then CyGNet combines the probabilistic predictions from both Copy mode and Generation mode to output the final prediction.

We first process the training set to obtain the historical vocabulary for each subject entity and predicate combination (s, p, t) on every t in training snapshots, i.e. $\{\mathbf{h}_{t_1}^{(s,p)}, \mathbf{h}_{t_2}^{(s,p)}, ..., \mathbf{h}_{t_T}^{(s,p)}\}$, where $\mathbf{h}_{t_k}^{(s,p)}$ is an N-dimensional multi-hot indicator vector that includes all object entities served as object entities in facts with the subject s and the predicate p in snapshot $\mathcal{G}_{t_k}$. As shown in Figure 4, we train the model sequentially on each snapshot, similar to the idea of recursion, trained by incrementally maintain the historical vocabulary for all the previous snapshots. When we evaluate CyGNet's performances in the validation set and test set, the maximum historical vocabulary from the whole training set will be used.

Specifically, for each query quadruple $(s, p, ?, t_k)$ on time t_k, the training process extends the historical vocabulary that specific to (s, p, t_k) from that of the previous snapshot, as formalized below:

$$\mathbf{H}_{t_k}^{(s,p)} = \mathbf{h}_{t_1}^{(s,p)} + \mathbf{h}_{t_2}^{(s,p)} + ... + \mathbf{h}_{t_{k-1}}^{(s,p)}, \quad (1)$$

where $\mathbf{H}_{t_k}^{(s,p)}$ is an N-dimensional multi-hot indicator vector where 1 is marked for all entities in the current historical vocabulary. We now introduce the two inference modes in the following.

Copy mode The Copy mode is designed to identify facts with repetition, and accordingly predict the future facts by copying from known facts in history.

If query $(s, p, ?, t_k)$ has the historical vocabulary $\mathbf{H}_{t_k}^{(s,p)}$

specific to the subject entity s and the predicate p at time step t_k, CyGNet will increase the probability estimated for the object entities that are selected in the historical vocabulary. In detail, the Copy mode first generates an index vector v_q with an MLP:

$$\mathbf{v}_q = \tanh(\mathbf{W}_c[\mathbf{s}, \mathbf{p}] + \mathbf{b}_c), \tag{2}$$

where $\mathbf{W}_c \in \mathbb{R}^{2d \times N}$ and $\mathbf{b}_c \in \mathbb{R}^N$ are trainable parameters. The index vector $\mathbf{v}_q$ is an N-dimensional vector, where N is the cardinality of the whole entity vocabulary $\mathcal{E}$.

To minimize the probability of some entities that do not form known facts with s and p in history (i.e. those being *uninterested* to the Copy mode), we first make modifications to $\mathbf{H}_{t_k}^{(s,p)}$. $\dot{\mathbf{H}}_{t_k}^{(s,p)}$ changes the index value for an uninterested entity in $\mathbf{H}_{t_k}^{(s,p)}$ to a small negative number. Therefore, CyGNet can delimit the candidate space by adding the index vector $\mathbf{v}_q$ and the changed multi-hot indicator vector $\dot{\mathbf{H}}_{t_k}^{(s,p)}$, minimize the probability of the uninterested entities, and then estimate the probability of the object entities in the historical vocabulary with a softmax function:

$$\mathbf{c}_q = \mathbf{v}_q + \dot{\mathbf{H}}_{t_k}^{(s,p)}, \tag{3}$$

$$\mathbf{p}(c) = \mathrm{softmax}(\mathbf{c}_q), \tag{4}$$

where $\mathbf{c}_q$ is an N-dimensional index vector, such that the values corresponding to *uninterested* entities in $\mathbf{c}_q$ are assigned close to zero. $\mathbf{p}(c)$ is a vector whose size equals to that of the whole entity vocabulary, and is to represent the prediction probabilities on the historical vocabulary. Eventually, the largest dimension of $\mathbf{p}(c)$ indicates the object entity to be copied from the historical vocabulary. The merit of the Copy mode is that it is able to learn to predict from a much delimited candidate space than the entire entity vocabulary. However, facts can also be newly appearing in a upcoming snapshot. And we therefore need a Generation mode to predict such facts.

Generation mode Given the same aforementioned query $(s, p, ?, t_k)$, the Generation mode is responsible for predicting facts by selecting the object entity from the whole entity vocabulary $\mathcal{E}$. The prediction made by the generation mode regards the predicted fact as an entirely new fact without any references to history. Similar to the Copy mode, the Generation mode also generates an index vector $\mathbf{g}_q$ whose size equals to the size of the candidate space $\mathcal{E}$, and is normalized with a softmax function to make predictions:

$$\mathbf{g}_q = \mathbf{W}_g[\mathbf{s}, \mathbf{p}, \mathbf{t_k}] + \mathbf{b}_g, \tag{5}$$

$$\mathbf{p}(g) = \mathrm{softmax}(\mathbf{g}_q), \tag{6}$$

where $\mathbf{W}_g \in \mathbb{R}^{3d \times N}$ and $\mathbf{b}_g \in \mathbb{R}^N$ are trainable parameters. Similar to $\mathbf{p}(c)$ in the Copy mode, $\mathbf{p}(g)$ represents the predicted probability on the whole entity vocabulary. The maximum value in $\mathbf{p}(g)$ indicates the object entity we predict in the whole entity vocabulary by the Generation mode. The Generation mode complements the Copy mode with the ability for *de novo* fact prediction.

3.3 Learning Objective

Predicting the (object) entity when given a query $(s, p, ?, t)$ can be viewed as a multi-class classification task, where each class corresponds to one object entity. The learning objective is to minimize the following cross-entropy loss $\mathcal{L}$ on all facts of the TKG snapshots that exist during training:

$$\mathcal{L} = -\sum_{t \in T} \sum_{i \in \mathcal{E}} \sum_{k=1}^{K} o_{it} \ln \mathbf{p}(y_{ik}|s, p, t) \tag{7}$$

where o_{it} is the i-th ground truth object entity in the snapshot $\mathcal{G}_t$, $\mathbf{p}(y_{ik}|s, p, t)$ is the combined probability value of the k-th object entity in the snapshot $\mathcal{G}_t$ when the i-th ground truth object entity is o_i.

3.4 Inference

Without loss of generality, we describe the inference process as predicting the missing object in a temporal fact, although this process can be easily extended to predicting the subject and the relation as well. To make prediction regarding a query $(s, p, ?, t_k)$, both Copy and Generation modes give a predicted object entity with the highest probability within their candidate spaces, whereof the Copy mode may make predictions from a much smaller candidate space than the entire entity vocabulary. To ensure that the sum of the probability equals 1 for all entities in $\mathcal{E}$, a coefficient α is incorporated to adjust the weight between the Copy mode and the Generation mode. CyGNet combines the probabilistic predictions from both the Copy mode and the Generation mode by adding the probability of each entity given by these two modes. The final prediction o_t will be the entity that receives the highest combined probability, as defined below:

$$\mathbf{p}(o|s, p, t) = \alpha * \mathbf{p}(c) + (1 - \alpha) * \mathbf{p}(g), \tag{8}$$

$$o_t = \mathrm{argmax}_{o \in \mathcal{E}} \mathbf{p}(o|s, p, t), \tag{9}$$

where $\alpha \in [0, 1]$, $\mathbf{p}(o|s, p, t)$ is the whole entity vocabulary size vector which contains the probability of all entities.

4 Experiments

In this section, we demonstrate the effectiveness of CyGNet with five public TKG datasets. We first explain experimental settings in detail, including details about baselines and datasets. After that, we discuss the experimental results. We also conduct an ablation study to evaluate the importance of different components of CyGNet[1].

4.1 Experimental Setup

Datasets. We evaluate CyGNet on the task of link prediction using five benchmark datasets, namely ICEWS18, ICEWS14, GDELT, WIKI and YAGO specifically. ICEWS records political facts (events) with timestamps, e.g., (*Donald Trump*, *visit*, *France*, 2018-04-10), and there are two benchmark datasets extracted from two time periods in this knowledge base i.e., ICEWS18 ((Boschee et al. 2015); from 1/1/2018 to 10/31/2018) and ICEWS14 ((Trivedi et al.

[1]The released source code and documentation are available at https://github.com/CunchaoZ/CyGNet.

#Data	#Entities	#Relation	#Training	#Validation	#Test	#Granularity	#Time Granules
ICEWS18	23,033	256	373,018	45,995	49,545	24 hours	304
ICEWS14	12,498	260	323,895	-	341,409	24 hours	365
GDELT	7,691	240	1,734,399	238,765	305,241	15 mins	2,751
WIKI	12,554	24	539,286	67,538	63,110	1 year	232
YAGO	10,623	10	161,540	19,523	20,026	1 year	189

Table 1: Statistics of the datasets.

Method	ICEWS18				ICEWS14				GDELT			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	17.56	2.48	26.95	43.87	18.65	1.12	31.34	47.07	16.05	0.00	26.10	42.29
DistMult	22.16	12.13	26.00	42.18	19.06	10.09	22.00	36.41	18.71	11.59	20.05	32.55
ComplEx	30.09	21.88	34.15	45.96	24.47	16.13	27.49	41.09	22.77	15.77	24.05	36.33
R-GCN	23.19	16.36	25.34	36.48	26.31	18.23	30.43	45.34	23.31	17.24	24.96	34.36
ConvE	36.67	28.51	39.80	50.69	40.73	33.20	43.92	54.35	35.99	27.05	39.32	49.44
RotatE	23.10	14.33	27.61	38.72	29.56	22.14	32.92	42.68	22.33	16.68	23.89	32.29
HyTE	7.31	3.10	7.50	14.95	11.48	5.64	13.04	22.51	6.37	0.00	6.72	18.63
TTransE	8.36	1.94	8.71	21.93	6.35	1.23	5.80	16.65	5.52	0.47	5.01	15.27
TA-DistMult	28.53	20.30	31.57	44.96	20.78	13.43	22.80	35.26	29.35	22.11	31.56	41.39
Know-Evolve+MLP	9.29	5.11	9.62	17.18	22.89	14.31	26.68	38.57	22.78	15.40	25.49	35.41
DyRep+MLP	9.86	5.14	10.66	18.66	24.61	15.88	28.87	39.34	23.94	15.57	27.88	36.58
R-GCRN+MLP	35.12	27.19	38.26	50.49	36.77	28.63	40.15	52.33	37.29	29.00	41.08	51.88
RE-NET	42.93	36.19	45.47	55.80	45.71	38.42	49.06	59.12	40.12	32.43	43.40	53.80
CyGNet	**47.83**	**42.02**	**50.71**	**57.72**	**49.89**	**43.15**	**53.68**	**61.18**	**51.06**	**44.66**	**54.74**	**61.32**

Table 2: Results (in percentage) on ICEWS18, ICESW14 and GDELT. Best results are boldfaced.

2017); from 1/1/2014 to 12/31/2014). GDELT (Leetaru and Schrodt 2013) is a catalog of human societal-scale behavior and beliefs extracted from news media, and the experimental dataset is collected from 1/1/2018 to 1/31/2018 with a time granularity of 15 mins. The WIKI and YAGO datasets are subsets of the Wikipedia history and YAGO3(Mahdisoltani, Biega, and Suchanek 2013), respectively. Since the WIKI and YAGO datasets contain temporal facts with a time span in the form as $(s, p, o, [T_s, T_e])$, where T_s is the starting time and T_e is the ending time, we follow prior work (Jin et al. 2019) to discretize these temporal facts into snapshots with the time granularity of a year. Table 1 summarizes the statistics of these datasets.

Baseline methods. We compare our method with a wide selection of static KGE and TKGE models. Static KGE methods include TransE (Bordes et al. 2013), DistMult (Yang et al. 2015), ComplEX (García-Durán, Dumancic, and Niepert 2018), R-GCN (Schlichtkrull et al. 2018), ConvE (Dettmers et al. 2018) and RotatE (Sun et al. 2019). Temporal methods include TTransE (Jiang et al. 2016), HyTE (Dasgupta, Ray, and Talukdar 2018), TA-DistMult (García-Durán, Dumancic, and Niepert 2018), Know-Evolve+MLP (Trivedi et al. 2017), DyRep+MLP (Trivedi et al. 2018), R-GCRN+MLP (Seo et al. 2018), and RE-NET (Jin et al. 2019), whereof RE-NET has offered SOTA performance on all of the benchmarks. Know-Evolve+MLP, DyRep+MLP, R-GCRN+MLP are the former methods in combination with MLP decoder. More detailed descriptions of baseline methods are given in Appendix B (Zhu et al. 2020).

Evaluation Setting and Metrics. Following prior work (Jin et al. 2019), we split each dataset except for ICEWS14 into training set, validation set and testing set into $80\%/10\%/10\%$ splits in the chronological order, whereas ICEWS14 is not provided with a validation set. We report Mean Reciprocal Ranks (MRR) and Hits@1/3/10 (the proportion of correct test cases that are ranked within top 1/3/10). We also enforce the *filtered* evaluation constraint that has been widely adopted in prior work (Jin et al. 2019; Dasgupta, Ray, and Talukdar 2018).

Model Configurations. The values of hyper-parameters are determined according to the MRR performance on each validation set. Specifically, since ICEWS14 does not come with a validation set, we directly carry forward the same hyper-parameter settings from ICEWS18 to ICEWS14. The coefficient α is tuned from 0.1 to 0.9 with a step of 0.1. It is set to 0.8 for ICEWS18 and ICEWS14, and 0.7 for GDELT, WIKI and YAGO. The model parameters are first initialized with Xavier initialization (Glorot and Bengio 2010), and are optimized using an AMSGrad optimizer with a learning rate of 0.001. The batch size is set to 1024. The training epoch is limited to 30, which is enough for convergence in most cases. The embedding dimension is set to 200 to be consistent with the baseline methods set by Jin et al. (2019). The baseline results are also adopted from Jin et al. (2019).

4.2 Results

Tabels 2 and 3 report the link prediction results by CyGNet and baseline methods on the five TKG datasets. As shown, CyGNet achieves the best performances in all cases. Static KGE methods generally show adequate results, while

Method	WIKI			YAGO		
	MRR	Hits@3	Hits@10	MRR	Hits@3	Hits@10
TransE	46.68	49.71	51.71	48.97	62.45	66.05
DistMult	46.12	49.81	51.38	59.47	60.91	65.26
ComplEx	47.84	50.08	51.39	61.29	62.28	66.82
R-GCN	37.57	39.66	41.90	41.30	44.44	52.68
ConvE	47.57	50.10	50.53	62.32	63.97	65.60
RotatE	50.67	50.71	50.88	65.09	65.67	66.16
HyTE	43.02	45.12	49.49	23.16	45.74	51.94
TTransE	31.74	36.25	43.45	32.57	43.39	53.37
TA-DistMult	48.09	49.51	51.70	61.72	65.32	67.19
Know-Evolve+MLP	12.64	14.33	21.57	6.19	6.59	11.48
DyRep+MLP	11.60	12.74	21.65	5.87	6.54	11.98
R-GCRN+MLP	47.71	48.14	49.66	53.89	56.06	61.19
RE-NET	51.97	52.07	53.91	65.16	65.63	68.08
CyGNet	**52.60**	**53.26**	**55.82**	**66.58**	**67.98**	**70.16**

Table 3: Results (in percentage) on WIKI and YAGO. Best results are boldfaced. Hits@1 was not reported by prior work (Jin et al. 2019) on these datasets. CyGNet achieved 50.68% in Hits@1 on WIKI and 64.09% in Hits@1 on YAGO.

Method	ICEWS18			
	MRR	Hits@1	Hits@3	Hits@10
CyGNet-Copy-only	42.12	38.82	44.32	46.71
CyGNet-Generation-only	40.17	32.14	44.10	55.03
CyGNet-Generation-new	45.84	39.28	49.10	57.61
CyGNet	**47.83**	**42.02**	**50.71**	**57.72**

Table 4: Results (in percentage) by different variants of our model on ICEWS18.

largely fall behind the best performing TKGE method as they do not capture the temporal dynamics. It can also be observed that all static KGE methods generally perform better than TTransE and HyTE. We believe this is due to that TTransE and HyTE learn representations independently for each snapshot, instead of capturing long-term dependency.

Table 2 shows that CyGNet drastically outperforms other baseline methods on ICEWS18, ICEWS14, GDELT. Specifically on GDELT, CyGNet leads to improvements of 10.94% in MRR, 12.23% in Hits@1, 11.34% in Hits@3, and 7.52% in Hits@10 over the best baseline method. Note that GDELT has denser training facts in each snapshot than other datasets, and has more complete historical information. The results in Table 3 show that static KGE baselines perform better than most TKGEs. On the other hand, CyGNet consistently surpasses the static KGE and TKGE methods on these two benchmarks as well. This implies that CyGNet effectively predict future facts via learning from history, and identifying and predicting new facts from scratch.

4.3 Ablation Study

To help understand the contribution of different model components of CyGNet, we present an ablation study. To do so, we create variants of CyGNet by adjusting the use of its model components, and compare the performance on the ICEWS18 dataset. Besides, more analyses are given in Appendix A (Zhu et al. 2020) due to space limitation.

From the results in Tables 4, we observe that the Copy mode and the Generation mode are both important. Removing the Copy mode can lead to a drop of 7.66% in MRR, as well as drastic drops of other metrics, which shows that learning to predict future facts by referring to the known facts in the past can be helpful. On the other hand, the removal of the Generation mode leads to a drop of 5.71% in MRR, which counts for the loss of the model's ability for *de novo* fact prediction. These results further explain that the promising performance by CyGNet is due to both the capability of learning from history, and identifying and predicting new facts from scratch.

CyGNet-Generation-new is another variant of CyGNet. The difference between CyGNet-Generation-new and the complete CyGNet is that the former utilizes the Generation mode to predict new facts in the whole entity vocabulary modulo the historical vocabulary. The performance of CyGNet is noticeably better than CyGNet-Generation-new. We believe this is because the original Generation mode in CyGNet can also strengthen the prediction in cases where a future fact is repeated. This merit is however discarded in the modified Generation mode of CyGNet-Generation-new.

5 Conclusion

Characterizing and inferring temporal knowledge is a challenging problem. In this paper, we for the first time leverage the copy mechanism to tackle this problem, based on the hypothesis that a future fact can be predicted from the facts in history. The proposed CyGNet is not only able to predict facts from the whole open world, it is also capable of identifying facts with repetitions, and accordingly selecting such future facts based on the known facts appeared in the past. The presented results on five benchmark datasets demonstrate CyGNet's promising performance for predicting future facts in TKGs. For future work, we plan to improve the sequential copy mechanism by identifying globally salient entities and events (Fan et al. 2019) in TKGs. Grounding dated documents to TKGs with the obtained embedding representations is also a meaningful direction. Besides, we are interested in leveraging the proposed techniques to help understand dynamic event processes in natural language text (Chen et al. 2020; Zhang et al. 2020).

Acknowledgment

We appreciate the anonymous reviewers for their insightful comments. This work is partially supported by NSFC-71701205, NSFC-62073333. Muhao Chen's work is supported by Air Force Research Laboratory under agreement number FA8750-20-2-10002, and by the DARPA MCS program under Contract No. N660011924033.

References

Balazevic, I.; Allen, C.; and Hospedales, T. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5188–5197.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2787–2795.

Boschee, E.; Lautenschlager, J.; O'Brien, S.; Shellman, S.; Starz, J.; and Ward, M. 2015. ICEWS coded event data. *Harvard Dataverse* 12.

Bouchard, G.; and Andreoli, J.-M. 2012. Temporal events analysis employing tree induction. US Patent 8,204,843.

Chen, J.; Chen, J.; and Yu, Z. 2019. Incorporating structured commonsense knowledge in story completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6244–6251.

Chen, M.; Zhang, H.; Wang, H.; and Roth, D. 2020. "What Are You Trying To Do?" Semantic Typing of Event Processes. In *Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL 2020)*. ACL.

Dai, Y.; Wang, S.; Xiong, N. N.; and Guo, W. 2020. A Survey on Knowledge Graph Embedding: Approaches, Applications and Benchmarks. *Electronics* 9(5): 750.

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

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2D knowledge graph embeddings. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, volume 32, 1811–1818. AAI Publications.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD*, 1555–1564.

Fan, C.; Zeng, L.; Ding, Y.; Chen, M.; Sun, Y.; and Liu, Z. 2019. Learning to identify high betweenness centrality nodes from scratch: A novel graph neural network approach. In *CIKM*, 559–568.

Feltham, R. 2004. *Diplomatic handbook*. Martinus Nijhoff Publishers.

García-Durán, A.; Dumancic, S.; and Niepert, M. 2018. Learning Sequence Encoders for Temporal Knowledge Graph Completion. In *EMNLP*.

Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Goel, R.; Kazemi, S. M.; Brubaker, M.; and Poupart, P. 2020. Diachronic embedding for temporal knowledge graph completion. In *AAAI*.

Gu, J.; Lu, Z.; Li, H.; and Li, V. O. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1631–1640.

Hao, J.; Ju, C. J.-T.; Chen, M.; Sun, Y.; Zaniolo, C.; and Wang, W. 2020. Bio-joie: Joint representation learning of biological knowledge bases. In *ACM BCB*, 1–10.

He, H.; Balakrishnan, A.; Eric, M.; and Liang, P. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, 1766–1776. Association for Computational Linguistics (ACL).

Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 687–696.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388* .

Jiang, T.; Liu, T.; Ge, T.; Sha, L.; Chang, B.; Li, S.; and Sui, Z. 2016. Towards time-aware knowledge graph completion. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 1715–1724.

Jin, W.; Jiang, H.; Qu, M.; Chen, T.; Zhang, C.; Szekely, P.; and Ren, X. 2019. Recurrent Event Network: Global Structure Inference over Temporal Knowledge Graph. *arXiv preprint arxiv:1904.05530v3* .

Kazemi, S. M.; and Poole, D. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in neural information processing systems*, 4284–4295.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer* 42(8): 30–37.

Korotayev, A. V.; and Tsirel, S. V. 2010. A spectral analysis of world GDP dynamics: Kondratieff waves, Kuznets

swings, Juglar and Kitchin cycles in global economic development, and the 2008–2009 economic crisis. *Structure and Dynamics* 4(1).

Lacroix, T.; Obozinski, G.; and Usunier, N. 2019. Tensor Decompositions for Temporal Knowledge Base Completion. In *International Conference on Learning Representations*.

Leetaru, K.; and Schrodt, P. A. 2013. GDELT: Global data on events, location, and tone. In *ISA Annual Convention*. Citeseer.

Liu, Z.; Xiong, C.; Sun, M.; and Liu, Z. 2018. Entity-Duet Neural Ranking: Understanding the Role of Knowledge Graph Semantics in Neural Information Retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2395–2405.

Lortscher Jr, F. D. 2010. System and method for generating transaction based recommendations. US Patent 7,720,732.

Ma, Y.; Tresp, V.; and Daxberger, E. A. 2019. Embedding models for episodic knowledge graphs. *Journal of Web Semantics* 59: 100490.

Mahdisoltani, F.; Biega, J.; and Suchanek, F. M. 2013. Yago3: A knowledge base from multilingual wikipedias.

Musiega, D. E.; and Kazadi, S.-N. 2004. Simulating the East African wildebeest migration patterns using GIS and remote sensing. *African Journal of Ecology* 42(4): 355–362.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, 593–607.

Seo, Y.; Defferrard, M.; Vandergheynst, P.; and Bresson, X. 2018. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, 362–373. Springer.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*. URL `https://openreview.net/forum?id=HkgEQnRqYQ`.

Trivedi, R.; Dai, H.; Wang, Y.; and Song, L. 2017. Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3462–3471.

Trivedi, R.; Farajtabar, M.; Biswal, P.; and Zha, H. 2018. Dyrep: Learning representations over dynamic graphs. In *International Conference on Learning Representations*.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, E.; and Bouchard, G. 2016. Complex Embeddings for Simple Link Prediction. In *International Conference on Machine Learning*, 2071–2080.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *Advances in neural information processing systems*, 2692–2700.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29(12): 2724–2743.

Wang, Z.; and Li, X. 2019. Hybrid-TE: Hybrid Translation-Based Temporal Knowledge Graph Embedding. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 1446–1451. IEEE.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 1112–1119.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*.

Zhang, H.; Chen, M.; Wang, H.; Song, Y.; and Roth, D. 2020. Analogous Process Structure Induction for Sub-event Sequence Prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. ACL.

Zhou, L.-k.; Yang, Y.; Ren, X.; Wu, F.; and Zhuang, Y. 2018a. Dynamic Network Embedding by Modeling Triadic Closure Process. In *AAAI*, 571–578.

Zhou, Q.; Yang, N.; Wei, F.; and Zhou, M. 2018b. Sequential copying networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhu, C.; Chen, M.; Fan, C.; Cheng, G.; and Zhang, Y. 2020. Learning from History: Modeling Temporal Knowledge Graphs with Sequential Copy-Generation Networks. *arXiv preprint* .

Supplementary Material

A Hyper-parameter Analysis

To help understand the contribution of different model components in CyGNet, we adjust the coefficient α to change the weight between the Copy mode and the Generation mode. The results are presented in Figure 5.

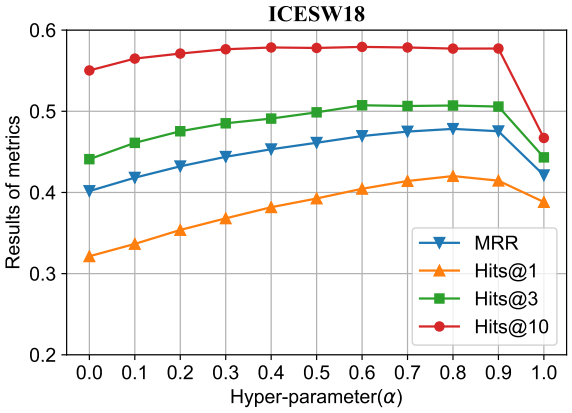

Figure 5: Results for different hyper-parameter α of CyGNet on ICEWS18.

The hyper-parameter α is 0 means that CyGNet only uses the Generation mode and α is 1 means that CyGNet only uses the Copy mode. As we can observe that, without considering the known facts which occurred in the past ($\alpha = 0$), the performance of CyGNet that only uses the Generation mode is not sufficiently effective. And within a certain range, with the increase of α, the performance of CyGNet increases. Excessive consideration of the known facts can lead to decrease the performance of CyGNet. And the most extreme is that CyGNet only uses the Copy mode that ignores the *de nove* facts, i.e., $\alpha = 1$. Thus we assume that CyGNet has the capability to balance the utilization between the historical vocabulary and the whole entity vocabulary by adjusting the coefficient α.

B Descriptions of Baseline Methods

We provide descriptions of baseline methods. In accord with Section 4.1, we separate the description in two groups.

TransE (Bordes et al. 2013) represents entities and relations in d-dimension vector space and makes the added embedding of the subject entity s and the predicate p be close to the embedding of the object entity o, i.e., $s + p \approx o$. DistMult (Yang et al. 2015) a simplified bilinear model by restricting relation matrix to be diagonal matrix for multi-relational representation learning. ComplEx (Trouillon et al. 2016) firstly introduces complex vector space which can capture both symmetric and antisymmetric relations. Relational Graph Convolutional Networks (RGCN;Schlichtkrull et al. (2018)) applies the previous Graph Convolutional Networks (GCN;Kipf and Welling (2017)) works to relational data modeling. ConvE (Dettmers et al. 2018) uses $2D$ convolution over embeddings and multiple layers of nonlinear features, and reshapes subject entity and predicate into $2D$ matrix to model the interactions between entities and relations. RotatE (Sun et al. 2019) proposes a rotational model taking predicate as a rotation from subject entity to object entity in complex sapce as $o = s \circ p$ where $\circ$ denotes the element-wise Hadmard product. However, these static KGE methods do not capture temporal facts.

More recent attempts have been made to model the evolving facts in TKGs. TTransE (Jiang et al. 2016) is an extension of TransE by embedding temporal information into the score function. HyTE (Dasgupta, Ray, and Talukdar 2018) replaces the projection normal vector in TransH (Wang et al. 2014) with a time-related normal vector. Know-Evolve (Trivedi et al. 2017) learns non-linearly evolving entity representations over time which models the occurrence of a fact as a temporal point process. TA-DistMult (García-Durán, Dumancic, and Niepert 2018) utilize recurrent neural networks to learn time-aware representations of relations and uses standard scoring functions from TransE and DistMult. RE-NET (Jin et al. 2019) models event (fact) sequences via RNN-based event encoder and neighborhood aggregator. DyREP (Trivedi et al. 2018) divides the dynamic graph network into two processes, and uses representation learning as a potential bridge connecting the two processes to learn the temporal structure information in the network. GCRN (Seo et al. 2018) merges CNN for graph-structured data and RNN to identify simultaneously meaningful spatial structures and dynamic patterns. Know-evolve, DyREP and GCRN combined with MLP decoder to predict future facts, which are called Know-evolve+MLP, DyRep+MLP and R-GCNR+MLP in (Jin et al. 2019).