Evolving Knowledge Graphs

Jiaqi Liu, Qin Zhang, Luoyi Fu, Xinbing Wang and Songwu Lu
{13-liujiaqi, 1374633989, yiluofu, xwang8}@sjtu.edu.cn, slu@cs.ucla.edu

***Abstract*—Many practical applications have observed knowledge evolution, i.e., continuous born of new knowledge, with its formation influenced by the structure of historical knowledge. This observation gives rise to evolving knowledge graphs whose structure temporally grows over time. However, both the modal characterization and the algorithmic implementation of evolving knowledge graphs remain unexplored. To this end, we propose EvolveKG, a framework that reveals cross-time knowledge interaction with desirable performance of storage and computation. The novelty of EvolveKG lies in Derivative Graph – a static weighted snapshot of evolution at a certain time. Particularly, each weight quantifies knowledge effectiveness with a temporarily decaying function of consistency and attenuation, two proposed factors depicting whether or not the effectiveness of a fact fades away with time. Thanks to the cross-time interaction, EvolveKG allows future knowledge prediction by virtue of the influence from the historical ones. Empirically tested under two real datasets, the superiority of EvolveKG is confirmed via its prediction accuracy.**

I. Introduction

Knowledge graph has been proven as an effective model for characterizing and studying complex multi-relational settings in real world [1]–[4]. In recent years, many knowledge graphs, such as Freebase [5], DBpedia [6] and YAGO [7], have been established and utilized in various real applications, including question answering [8], [9], information extraction [10], [11], named entity disambiguation [12], [13], semantic parsing [14], [15], to name a few. Traditionally, a knowledge graph consists of entities (nodes) and relations (edges). Each of its edge is represented as a triplet *(subject entity, relation, object entity)*, that indicates the fact *subjectEntity_relation_objectEntity*. For example, a triplet *(Jack, visits, France)* indicates the fact *Jack _visits_France*. However, traditional knowledge graphs merely provide a static snapshot of knowledge structure, which overlooks the evolving nature of knowledge.

Hence traditional knowledge graphs need to be augmented into evolving knowledge graphs, where knowledge temporarily expands over time with the continuously generated new facts. Figure 1 illustrates an example of evolving knowledge graphs, in which each edge is associated with a timestamp that records the fact's generation time. From the example we have the two observations summarized as below: i) the evolving knowledge graph expands over time. As we can see, the size of evolving knowledge graph before year 2000 is limited to merely five facts, which, however, surprisingly expands to a greater one with over ten facts in year 2018. ii) the generation of new facts is influenced by the historical ones. For example, Bob visited U.S. and Canada in year 2017 and year 2018, respectively. It is reasonable to infer that he established some business with companies in North America in his early visit to U.S., which consequently leads to his later visit to Canada. The illustrated observations convey to us that in evolving knowledge graphs, knowledge temporarily grows over time and the generation of new knowledge is influenced by the historical ones, which we term as *knowledge evolution*.

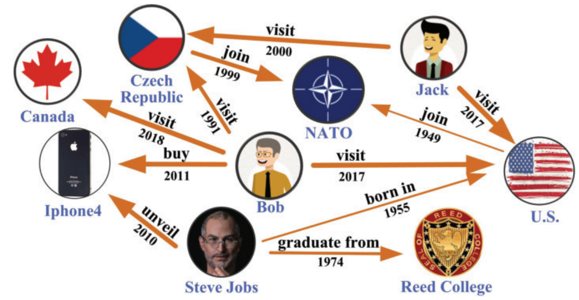


Fig. 1. An example of evolving knowledge graphs, where a fact happened more recently is represented by the line with a larger width.

We note that different from traditional knowledge graphs where knowledge interactions happen among knowledge at a same generation time, knowledge evolution depicts the interactions among knowledge across different generation times. Put differently, in knowledge evolution, there is an additional dimension - time, which can not be modeled by traditional knowledge graphs due to their incompleteness. Besides, most existing algorithms designed to learn knowledge graphs lack ability to take advantage of rich temporary information available in evolving knowledge graphs, which results in a limited performance on characterizing and utilizing cross-time influence happened in knowledge evolution. Motivated by this, in this work we aim to investigate from the following two aspects:

- **Modal characterization:** Does there exist any typical feature of knowledge evolution? If so, how to mathematically characterize them and how to quantify the cross-time influence among knowledge with different generation times?
- **Algorithmic implementation:** With the characterization of knowledge's cross-time influence, how to design and efficiently implement the algorithms to predict the generation of new knowledge by leveraging the historical ones?

To answer the questions, we first involve two factors, which we term as consistency and attenuation, to capture knowledge evolution of interest. i) *Consistency:* the generation likelihoods of a fact, at two different time points, are consistent if no new facts occurred between the time points; ii) *Attenuation:* the fact with a more recent timestamp has a greater influence on the future event compared with the one occurred earlier. For better interpretation, let us revisit the example given above. In general, the likelihoods that Bob visits Canada, in Jan. or Jun. 2018 are consistent if there are no new facts happened to him and Canada (Consistency). Besides, by noting Bob has more than one historical visits, we can infer that his visit on Canada in year 2018 is mainly influenced by his later visit on U.S., rather than the early one on Czech Republic (Attenuation).

In order to characterize the above factors and launch algorithmic learning in evolving knowledge graphs, we propose a novel framework - EvolveKG. The key idea of EvolveKG is to quantify the effectiveness of an evolving knowledge graph, via an ingeniously proposed Derivative Graph, which is a weighted snapshot of evolution at a certain time. Here each weight representing the effectiveness of a fact is calculated by a decaying function incorporating both consistency and attenuation. Leveraging effective historical knowledge, EvolveKG can further enable applications such as cross-time knowledge prediction. Meanwhile, EvolveKG provably returns ease of implementation with regard to both storage and computational complexities. Our work makes the following contributions:

- We characterize the facts in evolving knowledge graphs by quadruplets. Compared with static knowledge graphs whose edge only contains three items - subject entity, relation and object entity, our work integrally records an additional item - generation time of the fact. Thus the fact is represented as a quadruplet *(subject entity, relation, object entity, timestamp)*.
- EvolveKG characterizes effective knowledge contained in evolving knowledge graphs with Derivative Graph, whose size reflects the amount of effective knowledge and from which we can know the knowledge evolution is accelerated, constant or decelerated.
- The framework enables potential applications like cross-time knowledge prediction in evolving knowledge graphs. This application, which leverages historical knowledge to predict the future one, can not be achieved by traditional knowledge completion that aims to reproduce the missing data from those within the same generation time.
- We present performance evaluations on the prediction accuracy of EvolveKG. Experiments are conducted on two real datasets where it outperforms the baselines with up to 26.1% and 40.3% gains in terms of MeanRank and Hits@10.

To our best knowledge, this is the first attempt to explore cross-time interactions among knowledge, especially the influence of historical knowledge on future one. And we believe that it has potential to inspire a new research direction in modeling and analyzing multi-relational settings with time dimension.

II. Related Work

Knowledge Graph Learning: Knowledge graphs have been verified to be useful in wide applications such as information extraction [10], [11], question answering [8], [9], named entity disambiguation [12], [13] and semantic parsing [14], [15]. One of fundamental and important techniques in knowledge graphs is embedding, whose key idea is to embed the items in knowledge graphs, i.e., entities and relations, in continuous vector spaces, so as to make simplification while preserving the network structure. Towards this aim, many models have been proposed. TransE [1] is one of the most representative models. It learns vector embeddings of entities and relations, i.e., $\boldsymbol{e^s}$, $\boldsymbol{e^o}$ and $\boldsymbol{r}$, in space $\mathbb{R}^k$, $k > 0$. The basic idea of it is that the functional translation corresponds to a translation of embeddings, i.e., the model aims to learn parameters that make $\boldsymbol{e^s} + \boldsymbol{r} \approx \boldsymbol{e^o}$ when (e^s, r, e^o) holds. Besides, there are some other extended models such as TransH [2], TransR [3], etc. However, all the above models consider knowledge graphs as static ones, lack ability to capture their evolution and thus are limited in cross-time knowledge prediction.

TABLE I
NOTATIONS AND DEFINITIONS

Notation	Definition
$G(V, E)$	Knowledge graph with entity set V and relation set E.
$\widetilde{G}(V', E')$	Derivative Graph of $G(V, E)$.
e_v^s / e_v^o	Subject / Object entity v.
(e_v^s, r, e_u^o, t)	Fact where relation r occurs between subject entity e_v^s and object entity e_u^o at time t.
$d_v^s(t)$ / $d_v^o(t)$	Subject / object degree of entity v at time t.
$\hat{d}_v^s(t)$ / $\hat{d}_v^o(t)$	Weighted subject / object degree of entity v at time t.
$\hat{e}(t)$	Average of weighted number of edges in $\widetilde{G}(V', E')$.
λ	Parameter of the attenuation function.

Evolving Network Modeling: A multitude of previous studies [18]–[21] have clarified that network structure evolves over time. Regarding this, some models have been proposed, among which *Preferential Attachment* is a simple but useful one. In preferential attachment, for various reasons, nodes with more existing edges are more likely to create a new one. It further leads to a multiplicative process which is known to give power-law distributions. Due to its usability, preferential attachment has been widely used as a basic rule in varying scenes such as social networks [22], protein networks [23] and nanoparticles in liquid [24]. Besides preferential attachment, there are some other extended models [19]–[21]. However, all these models are formulated using probability theory as the mathematical tool, which can not be applied to capture evolving knowledge graphs since most of algorithms used in knowledge graphs are heuristic ones based on machine leaning implementation.

III. The Proposed Framework: EvolveKG

This section presents the framework's basic idea and some requirements, implementation, as well as complexity analysis.

A. Evolving Knowledge Graph Representation

Before the illustration on the proposed framework, we first give some definitions and common notations that will be used in the remainder of this paper. We model *Evolving Knowledge Graphs* as directed graphs with timestamped edges. In evolving knowledge graphs, each edge that points from a node to the other represents an event occurred in real word. In order to introduce time dimension, we extend the triplet (e^s, r, e^o) that is used in static knowledge graphs to a quadruplet (e^s, r, e^o, t). It represents a fact that relation r occurs between subject entity e^s and object entity e^o at time t. In the remaining parts, we adopt superscript s or o to indicate whether an entity acts as a subject or an object, and subscript to indicate the entity ID. At time t, for an entity e_v, the number of facts in which e_v acts as a subject is called *subject degree*, denoted by $d_v^s(t)$, and the number of facts in which e_v acts as an object is called *object degree*, denoted by $d_v^o(t)$. For convenience, we present Table I to list all notations that will be used later.

B. Basic Idea and Requirements of EvolveKG

Before the introduction on basic idea and requirements, we note that there are two widely accepted observations:

- The current behavior of a subject (an object) is influenced by all its historical facts where it acts as a subject (an object).
- A fact with an earlier generation time has a smaller influence on the entity's current behavior, while the one with a later generation time has a greater influence.

Inspired by these two observations, we model the influence of an entity's historical facts on its current behavior as a sum of the influence of each fact, multiplied by a weight. The weight is calculated by a function, called *attenuation function*, which ranges from 0 to 1 and decreases with the fact's existing time. Thus, how to determine the expression of attenuation function and judge whether it is a good one? We mathematically rewrite the observations in more details and declare that an effective attenuation function should satisfy all the below requirements. Denoting the probability that entity e_v^s forms a relation r with entity e^o at time t as $P\{(e_v^s, r, e_o, t)\}$, the requirements are

1) **Consistency:** If no fact happened during time t to $t+\Delta t$, the probability, where entity e_v acts as a subject, remains unchanged, i.e., $P\{(e_v^s, r, e^o, t)\} = P\{(e_v^s, r, e^o, t+\Delta t)\}$.
2) **Attenuation:** If some facts, denoted by the set ϕ, happened during time t to $t+\Delta t$, the probability satisfies

R1: $P\{\,(e_v^s, r, e^o, t+\Delta t)|\,\phi_1\} > P\{\,(e_v^s, r, e^o, t+\Delta t)|\,\phi_2\}$, where $\phi_1 \neq \emptyset$ and $\phi_2 = \emptyset$.

R2: $P\{\,(e_v^s, r, e^o, t+\Delta t)|\,\phi_1\} \geq P\{\,(e_v^s, r, e^o, t+\Delta t)|\,\phi_2\}$, where $\phi_1 = \{(e_v^s, r, e^o, t_1)\}$, $\phi_2 = \{(e_v^s, r, e^o, t_2)\}$, $t_1 \geq t_2$.

R3: $P\{\,(e_v^s, r, e^o, t+\Delta t)|\,\phi_1\} \geq P\{\,(e_v^s, r, e^o, t+\Delta t)|\,\phi_2\}$, where $\phi_1 \supset \phi_2$.

The above requirements indicate that the connecting probability remains unchanged if no new facts occurred during the time range; while a subject entity e_v^s with some historical facts (**R1**), a fact happened more recently (**R2**), or a larger coverage of historical facts (**R3**), is more likely to create new relations. These requirements also hold when an entity acts as an object.

Through calculation and analysis, we find that the requirements can be satisfied if we set the attenuation function as $f(t_i) = e^{-\lambda(t-t_i)}$, where t and t_i denote the current time and the generation time of the given fact, respectively. The proof of this result is provided in Section IV.

C. Implementation of EvolveKG

To solve the problems illustrated above, we propose a novel framework – EvolveKG. The framework owns two remarkable advantages: i) *Algorithm feasibility* - EvolveKG makes it feasible to learn evolving knowledge graphs by algorithms designed for static knowledge graphs; ii) *Cost effectiveness* - EvolveKG reduces storage complexity without loss of performance and involves no extra computation complexity.

The implementation of the framework includes two steps:

Step 1: Transformation from an evolving knowledge graph to a newly proposed graph, i.e., Derivative Graph.

Step 2: Training on the Derivative Graph.

Then, we discuss the two steps in more details.

1) Discussion on Step 1: First, we present the definition of Derivative Graph as below.

Definition 1 (Derivative Graph)**.** *For an evolving knowledge graph $G(V,E)$, its Derivative Graph at time t is defined as a weighted graph $\widetilde{G}(V',E')$, where $V'=V$ and for each edge $(v,u)\in E$ that is generated at time t_i, there exists an edge $(v,u)\in E'$ with a weight $e^{-\lambda(t-t_i)}$ in $\widetilde{G}(V',E')$.*

We note that the above definition has shown how to transform an evolving knowledge graph into its Derivative Graph. Since that each edge in Derivative Graph has a corresponding weight, we accordingly present the definition of an entity's *Weighted Subject Degree* and *Weighted Object Degree* as below.

Definition 2 (Weighted Subject/Object Degree)**.** *For an entity e_v with subject facts set $\{(e_v^s, r_i, e_i^o, t_i)\}$, $1\leq i\leq d_v^s(t)$, and object facts set $\{(e_i^s, r_i, e_v^o, t_i)\}$, $1\leq i\leq d_v^o(t)$, the entity's weighted subject degree and weighted object degree at time t are defined as*

$$\hat{d}_v^s(t) = \sum_{1\leq i\leq d_v^s(t)} e^{-\lambda(t-t_i)} \;\; and \;\; \hat{d}_v^o(t) = \sum_{1\leq i\leq d_v^o(t)} e^{-\lambda(t-t_i)}.$$

From an intuitive view, Derivative Graph is a characterization of effective information in evolving knowledge graphs. Note that the information is sensitive to time and the weight of an edge in Derivative Graph actually reflects the effectiveness of this fact in the knowledge graph. Based on this, in the second step of the framework, we can distinguish data according to their effectiveness, and then conduct a biased training.

2) Discussion on Step 2: The major challenge of this step lies in the problem that – *How to proceed the training with a weighted dataset?* Before the discussion on solutions, we first briefly review the training process. Given a loss function and a training set with m data, in each epoch, data d_i results in a loss l_i, and then the parameters are optimized by minimizing the total loss $L=\sum_{1\leq i\leq m} l_i$. Based on this process, we come up with two optional solutions to give a biased training:

1) **Data side:** Adopt a high sampling rate for data with high effectiveness, and a low one for data with low effectiveness.
2) **Loss side:** Increase the ratio of loss caused by data with high effectiveness, and inversely, decrease that resulted by data with low effectiveness.

In the implementation, either of the two solutions can achieve a biased training. One can make the selection between them based on the specific applications. In this paper, we adopt the second one and the loss function should be modified as

$$L(t) = \sum\nolimits_i e^{-\lambda(t-t_i)} l_i.$$

Following such a procedure, training and optimization can be proceeded on Derivative Graph. And finally, the framework makes it feasible to study evolving knowledge graphs through existing algorithms designed for static ones.

To illustrate the proposed framework, we give an example of an evolving knowledge graph with four facts. The example is presented in Figure 2. In Step 1, the evolving knowledge graph is transformed into a static weighted one. The weight of each edge is calculated by attenuation function $f(t_i) = e^{-\lambda(t-t_i)}$. In Step 2, each fact i results in a loss l_i, $i\in\{1,2,3,4\}$. The loss l_i is calculated by the function $l(\cdot)$, which is determined

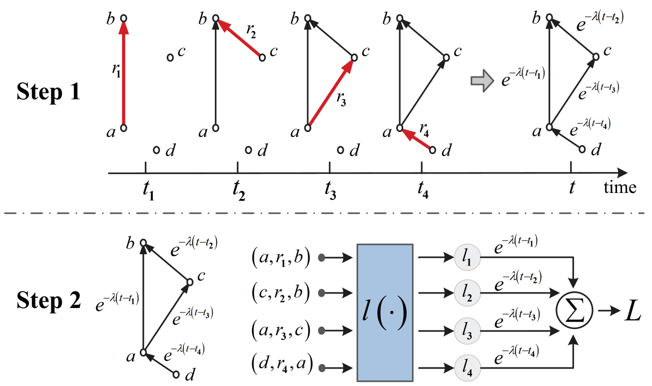


Fig. 2. An illustration on the proposed framework – EvolveKG.

by the applied algorithm. Then, the total loss L is obtained by summing over all the weighted loss $e^{-\lambda(t-t_i)}l_i$. And finally, training process is proceeded to minimize the loss.

We give an example here to show how to apply EvolveKG with TransE. *Step 1:* Transform an evolving knowledge graph $G(V, E)$ to its corresponding Derivative Graph $\widetilde{G}(V', E')$ as described. *Step 2:* Modify the loss function $L(t)$ to enable a biased training. Particularly, denoting the margin parameter as γ, the set of correct quadruplets as S and that of corrupted ones as S', and representing the embedding of an entity (relation) by its boldface, the loss function is modified as:

$$L(t) = \sum_{\substack{(e^s,r,e^o,t_i)\in S \\ (e^{s\prime},r,e^{o\prime},t_i)\in S'}} [\gamma + e^{-\lambda(t-t_i)}P - e^{-\lambda(t-t_i)}N]_+,$$

where $P = d(\boldsymbol{e^s} + \boldsymbol{r}, \boldsymbol{e^o})$, $N = d(\boldsymbol{e^{s\prime}} + \boldsymbol{r}, \boldsymbol{e^{o\prime}})$ and d is a dissimilarity measure.

D. Storage and Computation Complexities

Storage Complexity: First of all, let us consider a question: *Whether an entity's current behavior is influenced by all its historical facts? If no, under what conditions does a fact have no influence?* The answer is given in Lemma 1 with its proof in Appendix A. Results show that the influence of a fact with a sufficiently long existing time can be ignored. Based on this result, we demonstrate that without performance loss, storage complexity of EvolveKG is $\sum_{i=t_k}^{t} N(i)$, where $N(i)$ denotes the number of facts with timestamp i and $t - t_k = \Theta(1)$.

***Lemma* 1.** *For an entity e_v, assume that it acts as a subject entity in d_v^s historical facts, and denote the facts by a set $\{(e_v^s, r_i, e_i^o, t_i)\}$, where $t_i \le t_j$ for any $i \le j$. The influence of the fact with timestamp t_k, $1 \le k \le d_v^s$, on the probability $P\{(e_v^s, r, e^o, t)\}$ can be ignored if $t_{d_v^s} - t_k \to \infty$.*

Computation Complexity: We regard it from the two steps of EvolveKG. In Step 1, it calculates the weight for each edge, which results in a complexity $\Theta(|E|)$. In Step 2, it maks a biased training. Assume the complexity of applied algorithm per epoch is $\Theta(m)$, where m depends on the algorithm, e.g., $(n_e + n_r)k$ for TransE. Since the only additional operation is the modification on loss function, giving a complexity $\Theta(1)$, the complexity in this step is $\Theta(mn + 1)$, where n is the number of epochs. In general, $\Theta(|E|) \ll \Theta(mn)$ and thus we conclude that EvolveKG introduces no additional complexity.

Algorithm 1: Evolution of Knowledge Graph $G_t(V, E)$

Choose a speed function $s(t)$.

At time $t = 0$:

The initial graph $G_0(V, E)$ is given.

At time $t > 0$:

(Creation of New Facts) $s(t)$ new facts are created.

(Preferential Attachment) For each new fact, an entity e_v is selected as the subject with the probability proportional to its weighted subject degree $\hat{d}_v^s(t)$. Selection of the object follows a symmetrical process.

IV. Theoretical Analysis

This section proposes a model of knowledge graph evolution, based on which we (i) prove the correctness of attenuation function and (ii) present some properties of Derivative Graph.

A. Model and Assumptions

In order to theoretically characterize how a knowledge graph evolves over time, we adopt a modified version of *Preferential Attachment Model*. This model is a classical one and has been widely used to capture network evolution in various fields.

The model is presented in Algorithm 1. First of all, an initial graph $G_0(V, E)$ and a speed function $s(t)$ are provided. The network topology of $G_0(V, E)$ could be generated randomly because it has little effect on $G_t(V, E)$ when t is sufficiently large. By setting $s(t)$ as a time dependent function, we allow the evolution speed of the network change with time. Then, in each time slot t, $s(t)$ new facts are created. Each of the fact selects a subject entity and an object entity based on the preferential attachment manner. Accordingly, the connecting probability of two entities can be calculated as in Lemma 2.

***Lemma* 2.** *The probability that a fact happens between subject entity e_v^s and object entity e_u^o at time t is*

$$P\{(e_v^s, r, e_u^o, t)\} = \frac{1}{G_t}\hat{d}_v^s(t)\hat{d}_u^o(t),$$

where $G_t = \sum_v \sum_u P\{(e_v^s, r, e_u^o, t)\}$ is the normalization factor.

Noting that an entity's weighted degree reflects the effectiveness of its historical facts, **the intuition of Algorithm 1** could be explained as follows. An entity's current behavior is influenced by its historical facts, and in addition, an entity with more effective historical facts has a greater probability to perform similarly in current time. For better understanding, let us recall the example where Bob is an active business man who often visits around in past several years. Since Bob is an active business man based on his historical facts, it is reasonable to assume that he is likely to visit somewhere else currently.

In the adopted model, entity type and relation type are not considered. It is because that in this section, our study focuses on how the effectiveness of a fact evolves over time, rather than the interactions among entities, where the latter one has been well explored by many existing algorithms designed for static knowledge graphs. Thus, for concise of notation, we use triplet (e^s, e^o, t) to denote a fact in the remaining part.

B. Proofs on the Correctness of Attenuation Function

We now com to the theoretical proofs of of attenuation function. The basic requirements of attenuation function, illustrated in Section III-B, are proved in Theorem 1 and Theorem 2.

Theorem 1. *For any two subject-object entity pairs (e_v^s, e_u^o) and (e_w^s, e_z^o), if no fact, that involves any of the entities in the given pairs, happens during time t to $t+\Delta t$, we have*

$$\frac{P\{(e_v^s, e_u^o, t)\}}{P\{(e_w^s, e_z^o, t)\}} = \frac{P\{(e_v^s, e_u^o, t+\Delta t)\}}{P\{(e_w^s, e_z^o, t+\Delta t)\}}.$$

Proof. According to the definition of connecting probability $P\{(e_v^s, e_u^o, t)\}$, we have

$$P\{(e_v^s, e_u^o, t)\} = \frac{1}{G_t}\hat{d}_v^s \hat{d}_u^o = \frac{1}{G_t}\sum_{i=1}^{d_v^s} e^{-\lambda(t-t_i)} \sum_{j=1}^{d_u^o} e^{-\lambda(t-t_j)}$$

$$= \frac{1}{G_t} e^{2\lambda\Delta t} \sum_{i=1}^{d_v^s} e^{-\lambda(t+\Delta t-t_i)} \sum_{j=1}^{d_u^o} e^{-\lambda(t+\Delta t-t_j)}.$$

And similarly, we have

$$P\{(e_w^s, e_z^o, t)\} = \frac{1}{G_t} e^{2\lambda\Delta t} \sum_{i=1}^{d_w^s} e^{-\lambda(t+\Delta t-t_i)} \sum_{j=1}^{d_z^o} e^{-\lambda(t+\Delta t-t_j)}.$$

Thus, the ratio of connection probabilities between subject-object pairs (e_v^s, e_u^o) and (e_w^s, e_z^o) is

$$\frac{P\{(e_v^s, e_u^o, t)\}}{P\{(e_w^s, e_z^o, t)\}} = \frac{\sum_{i=1}^{d_v^s} e^{-\lambda(t+\Delta t-t_i)} \sum_{j=1}^{d_u^o} e^{-\lambda(t+\Delta t-t_j)}}{\sum_{i=1}^{d_w^s} e^{-\lambda(t+\Delta t-t_i)} \sum_{j=1}^{d_z^o} e^{-\lambda(t+\Delta t-t_j)}}.$$

Then, use the same method and we can obtain

$$\frac{P\{(e_v^s, e_u^o, t+\Delta t)\}}{P\{(e_w^s, e_z^o, t+\Delta t)\}} = \frac{\sum_{i=1}^{d_v^s} e^{-\lambda(t+\Delta t-t_i)} \sum_{j=1}^{d_u^o} e^{-\lambda(t+\Delta t-t_j)}}{\sum_{i=1}^{d_w^s} e^{-\lambda(t+\Delta t-t_i)} \sum_{j=1}^{d_z^o} e^{-\lambda(t+\Delta t-t_j)}}.$$

Comparing the two equations we can observe that these two ratios are the same, and thus we complete the proof. □

***Corollary* 1.** *If there is no fact happened in a knowledge graph during time t to $t+\Delta t$, the connecting probabilities $P\{(e^s, e^o, t)\}$ of any subject-object entity pairs in the graph remain unchanged.*

In addition to consistency proved in Theorem 1, we present the proof of attenuation in Theorem 2, before which we first give a useful lemma as below.

***Lemma* 3.** *For any variables $a_1 > a_2 \geq 0$, $b > 0$ and $c > 0$, it is satisfied that*

$$\frac{c+b+a_1}{c+b+a_2} \cdot \frac{b+a_2}{b+a_1} < 1.$$

Theorem 2. *For an entity e_v, given its connecting probability $P\{(e_v^s, e^o, t)\}$, the probability at time $t+\Delta t$, with the facts happened during time t to $t+\Delta t$ as the condition, satisfies*

R1: *For any $k \geq 1$,*

$$P\{(e_v^s, e^o, t+\Delta t)|\{(e_v^s, e_i^o, t_i)\}, 1 \leq i \leq k\} > P\{(e_v^s, e^o, t+\Delta t)|\emptyset\};$$

R2: *For any $t_1 \geq t_2$,*

$$P\{(e_v^s, e^o, t+\Delta t)|\{(e_v^s, e_1^o, t_1)\}\} \geq P\{(e_v^s, e^o, t+\Delta t)|\{(e_v^s, e_2^o, t_2)\}\};$$

R3: *For two fact sets $\phi_1 \supset \phi_2$,*

$$P\{(e_v^s, e^o, t+\Delta t)|\phi_1\} \geq P\{(e_v^s, e^o, t+\Delta t)|\phi_2\}.$$

The results are similar when entity e_v acts as an object.

Proof. Note that the difference among the probabilities given in the theorem lies in their conditions, i.e., set of facts that happened during time t to $t+\Delta t$. Therefore, before the proof of the results, we first consider a general case – comparison of $P\{(e_v^s, e^o, t+\Delta t)|\chi\}$ and $P\{(e_v^s, e^o, t+\Delta t)|\psi\}$, where χ and ψ are two arbitrary sets of historical facts.

According to the definition, we have

$$P\{(e_v^s, e^o, t+\Delta t)|\chi\} = \sum_u P\{(e_v^s, e_u^o, t+\Delta t)|\chi\}$$

$$= \frac{1}{G_{t+\Delta t}} \hat{d}_v^s(t+\Delta t) \sum_u \hat{d}_u^o(t+\Delta t)\Big|\chi$$

$$= \frac{\hat{d}_v^s(t+\Delta t)\sum_u \hat{d}_u^o(t+\Delta t)}{\left(\sum_{i,i\neq v}\hat{d}_i^s(t+\Delta t) + \hat{d}_v^s(t+\Delta t)\right)\sum_u \hat{d}_u^o(t+\Delta t)}\Bigg|\chi$$

$$\overset{a}{=} \frac{\hat{d}_v^s(t+\Delta t)\Big|\chi}{\sum_{i,i\neq v}\hat{d}_i^s(t+\Delta t) + \hat{d}_v^s(t+\Delta t)\Big|\chi}.$$

The equality (a) holds since that $\sum_{i,i\neq v}\hat{d}_i^s(t+\Delta t)$ is independent of the condition χ. Similarly, we have

$$P\{(e_v^s, e^o, t+\Delta t)|\psi\} = \frac{\hat{d}_v^s(t+\Delta t)\Big|\psi}{\sum_{i,i\neq v}\hat{d}_i^s(t+\Delta t) + \hat{d}_v^s(t+\Delta t)\Big|\psi}.$$

And therefore, the ratio of the two probabilities is

$$\frac{P\{(e_v^s, e^o, t+\Delta t)|\chi\}}{P\{(e_v^s, e^o, t+\Delta t)|\psi\}} = \frac{\sum_{i,i\neq v}\hat{d}_i^s(t+\Delta t) + \hat{d}_v^s(t+\Delta t)\Big|\psi}{\sum_{i,i\neq v}\hat{d}_i^s(t+\Delta t) + \hat{d}_v^s(t+\Delta t)\Big|\chi} \cdot \frac{\hat{d}_v^s(t+\Delta t)\Big|\chi}{\hat{d}_v^s(t+\Delta t)\Big|\psi}.$$

Let $c = \sum_{i,i\neq v}\hat{d}_i^s(t+\Delta t)$.

R1: Letting $\chi = \emptyset$ and $\psi = \{(e_v^s, e_i^0, t_i)\}, 1 \leq i \leq k$, we have

$$\hat{d}_v^s(t+\Delta t)\Big|\chi = \hat{d}_v^s(t)$$

$$\hat{d}_v^s(t+\Delta t)\Big|\psi = \hat{d}_v^s(t) + \sum\nolimits_{1\leq i\leq k} e^{-\lambda(t-t_i)}.$$

Let $b = \hat{d}_v^s(t)$, $a_1 = \sum_{1\leq i\leq k} e^{-\lambda(t-t_i)}$ and $a_2 = 0$. Finally, with Lemma 3, we obtain the result.

R2: Letting $\chi = \{(e_v^s, e_2^o, t_2)\}$, $\psi = \{(e_v^s, e_1^o, t_1)\}$, we have

$$\hat{d}_v^s(t+\Delta t)\Big|\chi = \hat{d}_v^s(t) + e^{-\lambda(t-t_2)}$$

$$\hat{d}_v^s(t+\Delta t)\Big|\psi = \hat{d}_v^s(t) + e^{-\lambda(t-t_1)}.$$

Let $b = \hat{d}_v^s(t)$, $a_1 = e^{-\lambda(t-t_1)}$ and $a_2 = e^{-\lambda(t-t_2)}$. Finally, with Lemma 3, we obtain the result.

R3: Let $\chi = \phi_2$ and $\psi = \phi_1$, and we have

$$\hat{d}_v^s(t+\Delta t)\Big|\chi = \hat{d}_v^s(t) + \sum\nolimits_{(e_v^s, e_i^o, t_i)\in\phi_2} e^{-\lambda(t-t_i)}$$

$$\hat{d}_v^s(t+\Delta t)\Big|\psi = \hat{d}_v^s(t) + \sum\nolimits_{(e_v^s, e_i^o, t_i)\in\phi_1} e^{-\lambda(t-t_i)}.$$

Similarly, let $b = \hat{d}_v^s(t)$, $a_1 = \sum_{(e_v^s, e_i^o, t_i)\in\phi_1} e^{-\lambda(t-t_i)}$ and $a_2 = \sum_{(e_v^s, e_i^o, t_i)\in\phi_2} e^{-\lambda(t-t_i)}$. Finally, according to Lemma 3, we finish the proof. Following the same method, we can obtain the results when the entity e_v acts as an object. □

Discussion on the requirements of attenuation function: Combining the results presented in Theorem 1 and Theorem 2, we have theoretically proved that $f(t_i) = e^{-\lambda(t-t_i)}$ is an effective attenuation function. Next, we give a discussion on these requirements. In fact, attenuation proved in Theorem 2 can be satisfied with any increasing function, i.e., $\forall t_1 > t_2$, $f(t_1) > f(t_2)$. While, consistency proved in Theorem 1 is a more rigorous one. We have tried several other forms of attenuation function, such as negative linear functions, i.e., $f(t_i) = -a(t-t_i)+b$, reciprocal functions, i.e., $f(t_i) = a\frac{1}{t-t_i} + b$ and find that the requirements can not be satisfied with these functions. However, we believe $f(t_i) = e^{-\lambda(t-t_i)}$ is not the only feasible choice for attenuation function. Some more attenuation functions may be developed in future work.

C. Properties of Derivative Graph

We first present some necessary preliminaries. According to the evolution speed $s(t)$, the evolving knowledge graph can be classified into three types:

Accelerated evolution: evolution speed $s(t)$ increases with time, i.e., $\forall t_1 > t_2$, $s(t_1) > s(t_2)$.

Constant evolution: evolution speed $s(t)$ remains unchanged with time, i.e., $\forall t_1, t_2$, $s(t_1) = s(t_2)$.

Decelerated evolution: evolution speed $s(t)$ decreases with time, i.e., $\forall t_1 > t_2$, $s(t_1) < s(t_2)$.

Theorem 3. *Assume t_0 is a sufficient large time that satisfies $t_0 \to \infty$. Denote the evolution speed of graph G as $s(t)$ and we have that if G follows an accelerated evolution, i.e., $s(t)$ increases with time t, then $\forall t_1 > t_2 \geq t_0$, $\forall e_v$,*

$$\hat{e}(t_1) > \hat{e}(t_2) \quad and \quad E[\hat{d}_v(t_1)] > E[\hat{d}_v(t_2)];$$

if G follows a constant evolution, i.e., $s(t)$ remains unchanged with time t, then $\forall t_1, t_2 \geq t_0$, $\forall e_v$,

$$\hat{e}(t_1) = \hat{e}(t_2) \quad and \quad E[\hat{d}_v(t_1)] = E[\hat{d}_v(t_2)];$$

and if G follows a decelerated evolution, i.e., $s(t)$ decreases with time t, then $\forall t_1 > t_2 \geq t_0$, $\forall e_v$,

$$\hat{e}(t_1) < \hat{e}(t_2) \quad and \quad E[\hat{d}_v(t_1)] < E[\hat{d}_v(t_2)],$$

where $\hat{e}(t)$ is the averaged number of edges in $\widetilde{G}$ and $E[\hat{d}_v(t)]$ is the average of weighted degree of entity e_v.

Proof. Firstly, we consider the change of $\hat{e}(t)$ at different time slots. To calculate $\hat{e}(t)$, note that in time slot t_i, there are $s(t_i)$ edges generated in the graph and their contributions to $\hat{e}(t)$ at time t are $s(t_i)e^{-\lambda(t-t_i)}$. Then, considering all the edges that are generated from time slot 1 to t, we have

$$\hat{e}(t) = \sum\nolimits_{0\leq t_i\leq t} s(t_i)e^{-\lambda(t-t_i)}. \tag{1}$$

In the following part, we discuss how $\hat{e}(t)$ changes with time in the three given cases.

Case 1: G follows an accelerated evolution, which indicates that $\forall t_1 > t_2 \geq t_0$, $s(t_1) > s(t_2)$. According to Equation (1), we can obtain that

$$\hat{e}(t_1) = \sum_{0\leq t_i\leq t_1} s(t_i)e^{-\lambda(t_1-t_i)} > \sum_{t_1-t_2\leq t_i\leq t_1} s(t_i)e^{-\lambda(t_1-t_i)}.$$

Let $t_j = t_i + t_2 - t_1$, the equation can be rewrote as

$$\hat{e}(t_1) > \sum\nolimits_{0\leq t_j\leq t_2} s(t_j + t_1 - t_2)e^{-\lambda(t_2-t_j)}. \tag{2}$$

And similarly, according to Equation (1), we have

$$\hat{e}(t_2) = \sum\nolimits_{0\leq t_i\leq t_2} s(t_i)e^{-\lambda(t_2-t_i)}. \tag{3}$$

Since that $\forall t > t_0$, $s(t+t_1-t_2) > s(t)$, combining Equation (2) and Equation (3) we have $\hat{e}(t_1) > \hat{e}(t_2)$.

Case 2: G follows a constant evolution, which indicates that $\forall t_1, t_2 \geq t_0$, $s(t_1) = s(t_2)$. With Equation (1), we have

$$\hat{e}(t_1) = \sum\nolimits_{0\leq t_i\leq t_1} s(t_i)e^{-\lambda(t_1-t_i)} = s(0)\frac{1}{1-e^{-\lambda}}.$$

$$\hat{e}(t_1) = s(0)\frac{1}{1-e^{-\lambda}}.$$

Therefore, $\hat{e}(t_1) = \hat{e}(t_2)$ and we complete the proof.

Case 3: G follows a decelerated evolution, which indicates that $\forall t_1 > t_2 \geq t_0$, $s(t_1) < s(t_2)$. With Equation (1), we have

$$\begin{aligned}
&\hat{e}(t_1) = \sum\nolimits_{0\leq t_i\leq t_1} s(t_i)e^{-\lambda(t_1-t_i)} \\
&= \sum_{0\leq t_i\leq t_2} s(t_i)e^{-\lambda(t_2-t_i)}e^{-\lambda(t_2-t_2)} + \sum_{t_1\leq t_i\leq t_2+1}^{t_1} s(t_i)e^{-\lambda(t_1-t_i)} \\
&= \hat{e}(t_2) - \hat{e}(t_2)\left(1-e^{-\lambda(t_2-t_2)}\right) + \sum_{t_1\leq t_i\leq t_2+1} s(t_i)e^{-\lambda(t_1-t_i)}.
\end{aligned}$$

In order to make the comparison between $\hat{e}(t_1)$ and $\hat{e}(t_2)$, we should first calculate the value of $\hat{e}(t_2)\left(1-e^{-\lambda(t_2-t_2)}\right)$ and that of $\sum_{t_i=t_2+1}^{t_1} s(t_i)e^{-\lambda(t_1-t_i)}$.

$$\hat{e}(t_2)\left(1-e^{-\lambda(t_1-t_2)}\right) > s(t_2)\frac{1-e^{-\lambda(t_1-t_2)}}{1-e^{-\lambda}}, \tag{4}$$

where $\hat{e}(t_2) > s(t_1)\sum_{t_i=0}^{t_1} e^{-\lambda(t_1-t_i)} = s(t_1)\frac{1}{1-e^{-\lambda}}$. And for the second item, we have

$$\sum\nolimits_{t_1\leq t_i\leq t_2+1} s(t_i)e^{-\lambda(t_1-t_i)} < s(t_2+1)\frac{1-e^{-\lambda(t_1-t_2)}}{1-e^{-\lambda}}. \tag{5}$$

Finally, we have that $\hat{e}(t_1) < \hat{e}(t_2)$.

Secondly, we discuss the value of $E[\hat{d}_v(t)]$. From time slot $t-1$ to t, the change of $E[\hat{d}_v^s(t-1)]$ is resulted by two parts:

- Part 1: Decrease resulted by existing edges.
- Part 2: Increase resulted by new generated edges.

Part 1: Time elapse results in the decrease of the weight of each edge, by a factor $e^{-\lambda}$. Therefore, we have

$E[\hat{d}_v^s(t)] = e^{-\lambda}E[\hat{d}_v^s(t-1)]$ and $E[\hat{d}_v^o(t)] = e^{-\lambda}E[\hat{d}_v^o(t-1)]$.

Part 2: Based on the evolving model, in time slot t, there are $s(t)$ new edges generated. The probability that one of these edges connects to entity e_v, where e_v acts as a subject, is

$$\begin{aligned} P\{(e_v^s, e^o, t)\} =& s(t-1)\sum_u P\{(e_v^s, e_u^o, t-1)\} \\ =& s(t-1)\cdot\frac{1}{G_{t-1}}\hat{d}_v^s(t-1)\sum_u \hat{d}_u^o(t-1), \end{aligned} \tag{6}$$

where G_{t-1} is the normalization coefficient that satisfies

$$\begin{aligned} G_{t-1} &= \sum_v\sum_u \hat{d}_v^s(t-1)\hat{d}_u^o(t-1) \\ &= \sum_v \hat{d}_v^s(t-1)\cdot\sum_u \hat{d}_u^o(t-1) = \hat{e}^2(t-1) \end{aligned}$$

Then, plugging the value of G_{t-1} into Equation (6) and using Equation (1), we have

$$P\{(e_v^s, e^o, t)\} = \hat{d}_v^s(t-1)\cdot s(t-1)\frac{1}{\sum_{t_i=0}^{t-1} s(t_i)e^{-\lambda(t-1-t_i)}}.$$

Combining the results in Part 1 and Part 2, we have

$$\begin{aligned} E[\hat{d}_v^s(t)] &= e^{-\lambda}E[\hat{d}_v^s(t-1)] + P\{(e_v^s, e^o, t)\} \\ &= E[\hat{d}_v^s(t-1)]\left(e^{-\lambda} + \frac{s(t-1)}{\sum_{t_i=0}^{t-1} s(t_i)e^{-\lambda(t-1-t_i)}}\right). \end{aligned}$$

In the following part, we discuss how $E[\hat{d}_v^s(t)]$ changes with time in the three given cases.

In Case 1, graph G follows an accelerated evolution, which indicates that $\forall t_1 > t_2 \geq t_0$, $s(t_1) > s(t_2)$. Then, by noting that $\forall 0 \leq t_i \leq t-1, s(t_i) < s(t-1)$, the equation given above can be calculated as

$$\begin{aligned} E[\hat{d}_v^s(t)] &> E[\hat{d}_v^s(t-1)]\left(e^{-\lambda} + \frac{s(t-1)}{s(t-1)\sum_{t_i=0}^{t-1} e^{-\lambda(t-1-t_i)}}\right) \\ &= E[\hat{d}_v^s(t-1)]\left(e^{-\lambda} + \frac{1-e^{-\lambda}}{1-e^{-\lambda t}}\right) = E[\hat{d}_v^s(t-1)]. \end{aligned}$$

The third equality holds since that $t \geq t_0$. The result indicates that $\forall t \geq t_0$, $\forall e_v$, $E[\hat{d}_v^s(t)] > E[\hat{d}_v^s(t-1)]$. According to this result, we can easily get the final conclusion, given in the theorem, that $\forall t_1 > t_2 \geq t_0$, $\forall e_v$, $E[\hat{d}_v^s(t_1)] > E[\hat{d}_v^s(t_2)]$.

Following a similar method, by noting that $\forall 0 \leq t_i \leq t-1, s(t_i) = s(t-1)$ in Case 2 and $\forall 0 \leq t_i \leq t-1, s(t_i) > s(t-1)$ in Case 3, we finish the proof. □

Intuitions on the properties of Derivative Graph: Results in Theorem 3 indicate that, the size of Derivative Graph grows if an accelerated evolution happens in the evolving knowledge graph; remains unchanged if a constant evolution happens; and decreases if a decelerated evolution happens. Since the relationship between the size of $G(V, E)$ and that of $\widetilde{G}(V', E')$ follows a similar manner as that between a function and its derivative function, we call $\widetilde{G}(V', E')$ the Derivative Graph of $G(V, E)$. In addition, we note that the size of Derivative Graph actually reflects the amount of effective information hidden in the corresponding knowledge graph. And therefore, results in Theorem 3 signify that an accelerated evolution stimulates an increasing amount of effective information in an evolving knowledge graph, which may be helpful for applications such as knowledge prediction, graph implementation, etc.

TABLE II
STATISTICAL PROPERTIES OF DATASETS.

	ICEWS	GDELT
# Facts	133,008	145,508
# Entities	15,624	6,863
# Relations	236	227
Start Time	2016-01-01	2013-01-01
End Time	2016-12-31	2013-12-31
Time Granularity	1 day	1 day

V. EXPERIMENTAL MEASUREMENTS

Our proposed framework is evaluated by its performance of knowledge prediction, applied to an existing algorithm TransE.

A. Description on Datasets

The experiments are conducted on two real datasets: Global Database of Events, Language, and Tone (GDELT) [25] and Integrated Crisis Early Warning System (ICEWS) [26].

GDELT is a CAMEO-coded dataset. The data are collected from reports in a variety of international news sources such as BBC Monitoring, Washington Post, New York Times, etc. **ICEWS** is a temporal dataset that records coded interactions among actors, i.e., cooperative or hostile actions between individuals and groups. Facts are automatically identified and extracted from news by the BBN ACCENT event coder.

In the above two datasets, each item consists of a subject entity, a relation, an object entity and a timestamp, that naturally form a quadruplet and represent the fact in evolving knowledge graphs. More details on the datasets are listed in Table II.

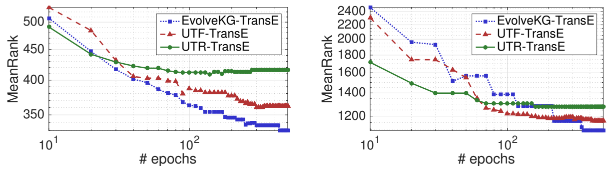
B. Experimental Settings

1) Evaluation Protocol: For evaluation, we use two metrics, i.e., *MeanRank* and *Hits@10*, which is the same as that in [1]. For each test quadruple, the corrupted quadruplets are created by removing and replacing the subject entity by each of entities of the dataset in turn. Then, repeat this procedure by removing the object entity instead of the subject entity. Dissimilarities of both correct quadruplets and corrupted quadruplets are calculated by the model and sorted by an ascending order. We report the mean of those predicted facts as *MeanRank* and the proportion of correct facts ranked in the top 10 as *Hits@10*.

2) Baseline Schemes: As illustrated, EvolveKG can be applied to many existing algorithms of static knowledge graphs. In the evaluation, we choose TransE as the target algorithm. Other algorithms, like TransR, TransH are also appliable but we do not include them since they are all derivatives of TransE. For explicitation, we rename it as EvolveKG-TransE. Besides, we include other two schemes to make the comparison:

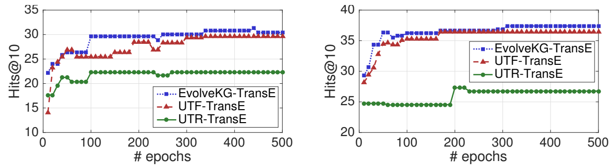
- **UTF-TransE:** A modification of TransE, giving an Uniform Training with the Full historical facts.
- **UTR-TransE:** A modification of TransE, giving an Uniform Training with only Recent historical facts.

These schemes given above adopt different strategies to deal with historical facts. UTF-TransE assumes an entity's current behavior is influenced by all its historical facts with a same weight. UTR-TransE assumes only recent historical facts have



(a) GDELT. (b) ICEWS.

Fig. 3. Evaluation on MeanRank.



(a) GDELT. (b) ICEWS.

Fig. 4. Evaluation on Hits@10.

influence on an entity's current behavior. By comparing the performances of them we can figure out that whether historical knowledge has influence on the future one and whether the effectiveness of historical knowledge decays with its generation time. Particularly, in both ICEWS and GDELT, UTR-TransE only includes historical facts generated in recent two months.

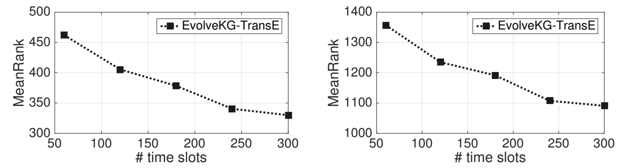
3) Training Process and Parameter Setting: Owing to the introduction of time, the training process and the validation in our experiments are different from that in tradition one. Rather than randomly sampling from the dataset, in our experiments, we use data with an earlier timestamp for training and that with a later timestamp for validation and test. For example, a dataset with timestamp ranging from 1 to 100 is separated into two parts: a training set with timestamp from 1 to 80 and validation and test sets with timestamp from 81 to 100.

For the proposed framework, we set its attenuation function parameter as $\lambda = 0.01$, which leads to a great attenuation on a fact's effectiveness, e.g., the effectiveness of a fact generated one year ago is only 0.027, an extremely small influence on the future event. For all the three algorithms, we set parameters as learning rate $l = 0.01$, embedding dimension $k = 50$, and select margin γ from the set $\{0.01, 0.05, 0.1, 0.2, 0.5\}$. The optimal margin setting in GDELT is $\gamma = 0.05$ for EvolveKG-TransE and UTF-TransE and $\gamma = 0.2$ for UTR-TransE, and in ICEWS is $\gamma = 0.2$ for both EvolveKG-TransE and UTF-TransE and $\gamma = 0.5$ for UTR-TransE. The number of total epochs is set as 500. And the best models are selected by early stopping using MeanRank on validation sets.

C. Quantitive Results

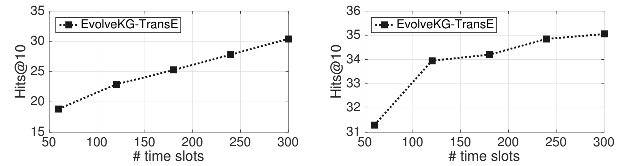
1) Examples on Knowledge Prediction: Table III presents examples of knowledge prediction on object entity. The experiments are conducted with EvolveKG-TransE on ICEWS. Fix an subject entity, a relation and a timestamp (in this example it is relaxed to a small time range, i.e., one month), all the top predicted object entities, along with the true one represented in boldface, are listed. In addition, following a similar method, one can also predict subject entity, relation, or timestamp.

2) Performance on Knowledge Prediction: The evaluation results on knowledge prediction are demonstrated in Figure 3 and Figure 4. From the figures we can observe that EvolveKG-TransE outperforms the other two baselines in terms of both



(a) GDELT. (b) ICEWS.

Fig. 5. MeanRank with varying time ranges of historical knowledge.



(a) GDELT. (b) ICEWS.

Fig. 6. Hits@10 with varying time ranges of historical knowledge.

MeanRank and Hits@10. UTF-TransE's second best performance, compared with UTR-TransE, shows that the historical knowledge does have influence on the formation of future one, and it is beneficial to take them into account for knowledge prediction. Whereas, it fails to capture the attenuation feature of knowledge evolution and therefore has a lower performance compared with EvolveKG-TransE. We also design experiments to explore the influence of historical knowledge with different time ranges on prediction performance. Results are provided in Figure 5 and Figure 6, in which we attempt to predict the knowledge occurred in time 300 to 360. The horizontal axis of the figures represents the time range of utilized historical knowledge. For example, the data point with 120 time slots means that the prediction is conducted based on the historical facts with timestamps ranging from 180 to 300. Based on the results, we conclude that a wider range of historical knowledge helps to improve the performance on prediction. We note that our results in Lemma 1 can not be observed directly from the figures, may resulted by the limited time range of datasets.

3) Performance on Complexity: Computation complexities of the schemes are listed in Table IV, where they are evaluated by running time per epochs on a computer with configuration Intel(R) Xeon(R) CPU E5-2630 2.40GHz. Results show that UTR-TransE has the smallest running time due to a reduced size of training set, while EvolveKG-TransE and UTF-TransE, sharing a common training set, have a roughly similar performance. Based on the above results we conclude that, the proposed framework is efficiently implemented, which brings no additional complexity compared with the target algorithm.

VI. Conclusion and Future Work

In this paper, we theoretically model the evolving knowledge graph and propose EvolveKG - a novel framework to learn it. The idea of EvolveKG is to transform the evolving knowledge graph to Derivative Graph and studies it through a biased training. We present theoretical analysis on EvolveKG. Results show that EvolveKG is efficiently implemented with regard to both storage and computation, and the attenuation function incorporates all the requirements. In addition to theoretical analysis, we also conduct experimental measurements on two realistic temporary datasets. Results declare that the

TABLE III
EXAMPLES OF PREDICTION ON OBJECT ENTITY.

Subject Entity	Relation	Time	Object Entity
Japan	Engage in negotiation	2016-11	Japan, **South Korea**, China, North Korea, Southeast Asia, Association of Southeast Asian Nations,Government (Japan), Vietnam, United States, Laos
Pakistan	Accuse	2016-11	Pakistan, Afghanistan, **India**, Other Authorities / Officials (Pakistan), Taliban, Sri Lanka, Government (Afghanistan), Citizen (Pakistan), Saudi Arabia, Bangladesh
Judiciary (India)	Make an appeal or request	2016-11	Education (India), **Government (India)**, Business (India), Company - Owner or Operator (India), Member of the Judiciary (India), Party Member (India), Other Authorities / Officials (India), Attacker (India), Criminal (India), Thief (India)

TABLE IV
RUNNING TIME PRE EPOCH.

	EvolveKG-TransE	UTF-TransE	UTR-TransE
GDELT	12.578s	11.745s	7.364s
ICEWS	11.054s	10.474s	6.098s

proposed framework outperforms other baseline algorithms in terms of both MeanRank and Hits@10.

There remains some future directions that can be explored. For example, this work only discusses Derivative Graph from a theoretical view. A desirable future work is to empirically study Derivative Graph, which has potential to be a guide of data selection since it characterizes the data effectiveness.

ACKNOWLEDGMENT

This work was supported by NSF China (No. 61602303, 61822206, 61532012, 61829201, 61832013), National Key R&D Program of China 2018YFB1004705, CCF-Tencent RAGR 20180116, Agri-X2017010, the Science and Technology Innovation Program of Shanghai under Grant 17511105103 and Shanghai Key Laboratory of Scalable Computing and Systems.

REFERENCES

[1] A. Bordes, N. Usunier, A. Garcia-Duran, *et al.*, "Translating embeddings for modeling multi-relational data," in *Advances in neural information processing systems*, 2013, pp. 2787–2795.
[2] Z. Wang, J. Zhang, J. Feng, *et al.*, "Knowledge graph embedding by translating on hyperplanes." in *AAAI*, vol. 14, 2014, pp. 1112–1119.
[3] Y. Lin, Z. Liu, M. Sun, *et al.*, "Learning entity and relation embeddings for knowledge graph completion." in *AAAI*, vol. 15, 2015, pp. 2181–2187.
[4] G. Ji, S. He, L. Xu, *et al.*, "Knowledge graph embedding via dynamic mapping matrix," in *ACL*, vol. 1, 2015, pp. 687–696.
[5] K. Bollacker, C. Evans, P. Paritosh, *et al.*, "Freebase: a collaboratively created graph database for structuring human knowledge," in *ACM SIGMOD*, vol. 2, 2008, pp. 1247–1250.
[6] J. Lehmann, R. Isele, M. Jakob, *et al.*, "Dbpedia – a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.
[7] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *WWW*, 2007, pp. 697–706.
[8] D. Lukovnikov, A. Fischer, J. Lehmann, *et al.*, "Neural network-based question answering over knowledge graphs on word and character level," in *WWW*, 2017, pp. 1211–1220.
[9] W.-t. Yih, M. Richardson, C. Meek, *et al.*, "The value of semantic parse labeling for knowledge base question answering," in *ACL*, vol. 2, 2016, pp. 201–206.
[10] R. Hoffmann, C. Zhang, X. Ling, *et al.*, "Knowledge-based weak supervision for information extraction of overlapping relations," in *ACL*, 2011, pp. 541–550.
[11] J. Daiber, M. Jakob, C. Hokamp, *et al.*, "Improving efficiency and accuracy in multilingual entity extraction," in *Semantic Systems*, 2013, pp. 121–124.
[12] D. Damljanovic and K. Bontcheva, "Named entity disambiguation using linked data," in *Semantic Web*, 2012, pp. 231–240.
[13] Z. Zheng, X. Si, F. Li, *et al.*, "Entity disambiguation with freebase," in *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, vol. 1, 2012, pp. 82–89.
[14] J. Berant, A. Chou, R. Frostig, *et al.*, "Semantic parsing on freebase from question-answer pairs," in *Empirical Methods in Natural Language Processing*, 2013, pp. 1533–1544.
[15] L. Heck, D. Hakkani-Tür, and G. Tur, "Leveraging knowledge graphs for web-scale unsupervised semantic parsing," *Interspeech*, 2013.
[16] G. Ji, K. Liu, S. He,*et al.*, "Knowledge graph completion with adaptive sparse transfer matrix." in *AAAI*, 2016, pp. 985–991.
[17] J. Feng, M. Huang, M. Wang, *et al.*, "Knowledge graph embedding by flexible translation." in *KR*, 2016, pp. 557–560.
[18] N. Bhushan, J. Li, D. Malladi, *et al.*, "Network densification: the dominant theme for wireless evolution into 5g," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, 2014.
[19] L. Meng, Y. Hulovatyy, A. Striegel, *et al.*, "On the interplay between individuals' evolving interaction patterns and traits in dynamic multiplex social networks," *IEEE Trans. on Network Science and Engineering.*, vol. 3, no. 1, pp. 32–43, 2016.
[20] Q. Liu, X. Zhao, W. Willinger, *et al.*, "Self-similarity in social network dynamics," *ACM Trans. on Modeling and Performance Evaluation of Computing Systems.*, vol. 2, no. 1, pp. 32–43, 2016.
[21] S. Lattanzi and D. Sivakumar, "Affiliation networks," in *ACM STOC*, 2009, pp. 427–434.
[22] K. Zuev, M. Boguná, G. Bianconi, *et al.*, "Emergence of soft communities from geometric preferential attachment," *Scientific reports*, vol. 5, p. 9421, 2015.
[23] E. Eisenberg and E. Y. Levanon, "Preferential attachment in the protein network evolution," *Physical review letters*, vol. 91, no. 13, p. 138701, 2003.
[24] D. A. Welch, T. J. Woehl, C. Park, *et al.*, "Understanding the role of solvation forces on the preferential attachment of nanoparticles in liquid," *ACS nano*, vol. 10, no. 1, pp. 181–187, 2015.
[25] K. Leetaru and P. A. Schrodt, "Gdelt: Global data on events, location, and tone, 1979–2012," in *ISA annual convention*, vol. 2, no. 4, 2013, pp. 1–49.
[26] E. Boschee, J. Lautenschlager, S. O?Brien, *et al.*, "Icews coded event data," *Harvard Dataverse*, vol. 12, 2015.

APPENDIX A
PROOF OF LEMMA 1

We first define the statement "the influence of the fact can be ignored". Consider the conditional probability, denote the connecting probability of e^s_v and e^o, with historical facts $\{(e^s_v, r_1, e^o_1, t_1), ..., (e^s_v, r_k, e^o_k, t_k), ..., (e^s_v, r_{d^s_v}, e^o_{d^s_v}, t_{d^s_v})\}$ as P and that with facts $\{(e^s_v, r_1, e^o_1, t_1), ..., (e^s_v, r_{k-1}, e^o_{k-1}, t_{k-1}), (e^s_v, r_{k+1}, e^o_{k+1}, t_{k+1}), ..., (e^s_v, r_{d^s_v}, e^o_{d^s_v}, t_{d^s_v})\}$ as $P_{\overline{k}}$. We regard the influence of the fact (e^s_v, r_k, e^o_k, t_k) on the probability P can be ignored if $\frac{P}{P_{\overline{k}}} = 1$. According to the above definition, we now come to the proof of the lemma. Firstly, we have

$$\begin{aligned}\frac{P}{P_{\overline{k}}} &= \frac{\frac{1}{G_t}\sum_{1\le i\le d^s_v} e^{-\lambda(t-t_i)}\sum_{1\le j\le d^o} e^{-\lambda(t-t_j)}}{\frac{1}{G_t}\sum_{1\le i\le d^s_v, i\neq k} e^{-\lambda(t-t_i)}\sum_{1\le j\le d^o} e^{-\lambda(t-t_j)}} \\ &= \frac{\sum_{1\le i\le d^s_v} e^{-\lambda(t-t_i)}}{\sum_{1\le i\le d^s_v, i\neq k} e^{-\lambda(t-t_i)}}.\end{aligned} \tag{7}$$

Obviously, $\frac{P}{P_{\overline{k}}} = 1$ holds when $\frac{e^{-\lambda(t-t_k)}}{\sum_{1\le i\le d^s_v, i\neq k} e^{-\lambda(t-t_i)}} = 0$.

Note that the recently happened fact, i.e., $(e^s_v, r_{d^s_v}, e^o_{d^s_v}, t_{d^s_v})$, has the greatest influence on connection probability and thus

$$\frac{e^{-\lambda(t-t_k)}}{\sum_{1\le i\le d^s_v, i\neq k} e^{-\lambda(t-t_i)}} \le \frac{e^{-\lambda(t-t_k)}}{e^{-\lambda(t-t_{d^s_v})}} = e^{-\lambda(t_{d^s_v}-t_k)} \tag{8}$$

Combine Equation (7) and Equation (8), and finally we have $\lim_{t_{d^s_v}-t_k\to\infty} \frac{P}{P_{\overline{k}}} = 1$. Therefore, we complete the proof.