Temporal Knowledge Graph Completion Based on Time Series Gaussian Embedding

Chenjin Xu[1](✉), Mojtaba Nayyeri[1], Fouad Alkhoury[1], Hamed Yazdi[1], and Jens Lehmann[1,2]

[1] Smart Data Analytics Group, University of Bonn, Bonn, Germany
{xuc,nayyeri,shariat,jens.lehmann}@cs.uni-bonn.de
[2] Enterprise Information Systems Department, Fraunhofer IAIS, Sankt Augustin, Germany
jens.lehmann@iais.fraunhofer.de
https://sda.tech/

Abstract. Knowledge Graph (KG) embedding has attracted more attention in recent years. Most KG embedding models learn from time-unaware triples. However, the inclusion of temporal information besides triples would further improve the performance of a KGE model. In this regard, we propose **ATiSE**, a temporal KG embedding model which incorporates time information into entity/relation representations by using **A**dditive **Ti**me **Se**ries decomposition. Moreover, considering the temporal uncertainty during the evolution of entity/relation representations over time, we map the representations of temporal KGs into the space of multi-dimensional Gaussian distributions. The mean of each entity/relation embedding at a time step shows the current expected position, whereas its covariance (which is temporally stationary) represents its temporal uncertainty. Experimental results show that ATiSE significantly outperforms the state-of-the-art KGE models and the existing temporal KGE models on link prediction over four temporal KGs.

Keywords: Temporal knowledge graph · Knowledge representation and reasoning · Time series decomposition

1 Introduction

Knowledge Graphs (KGs) are being used for gathering and organizing scattered human knowledge into structured knowledge systems. YAGO [22], DBpedia [1], WordNet [18] and Freebase [3] are among existing KGs that have been successfully used in various applications including question answering, assistant systems, information retrieval, etc. In these KGs, knowledge can be represented as RDF triples (*s, p, o*) in which *s* (subject) and *o* (object) are entities (nodes), and *p* (predicate) is the relation (edge) between them.

J. Z. Pan et al. (Eds.): ISWC 2020, LNCS 12506, pp. 654–671, 2020.
https://doi.org/10.1007/978-3-030-62419-4_37

KG embedding attempts to learn the representations of entities and relations in high-dimensional latent feature spaces while preserving certain properties of the original graph. Recently, KG embedding has become a very active research topic due to the wide ranges of downstream applications. Different KG embedding models have been proposed so far to efficiently learn the representations of KGs and perform KG completion as well as inferencing [4,9,23,25,28,30].

We notice that most of existing KG embedding models solely learn from time-unknown facts and ignore the useful temporal information in the KBs. In fact, there are many time-aware facts (or events) in some temporal KBs. For example, (*Obama, wasBornIn, Hawaii*) happened at August 4, 1961. (*Obama, presidentOf, USA*) was true from 2009 to 2017. These temporal KGs, e.g. Integrated Crisis Early Warning System (ICEWS) [14], Global Database of Events, Language, and Tone (GDELT) [16], YAGO3 [17] and Wikidata [6], store such temporal information either explicitly or implicitly. Traditional KBE models such as TransE learn only from time-unknown facts. Therefore, they cannot distinguish entities with similar semantic meaning. For instance, they often confuse entities such as *Barack Obama* and *Bill Clinton* when predicting $(?, \textit{presidentOf}, \textit{USA}, 2010)$.

To tackle this problem, temporal KGE models [5,7,15] encode time information in their embeddings. TKGE models outperform traditional KGE models on link prediction over temporal KGs. It justifies that incorporation of time information can further improve the performance of a KGE model. Most existing TKGE models embed time information into a latent space, e.g. representing time as a vector. These models cannot capture some properties of time information such as the length of time interval as well as order of two time points. Moreover, these models ignore the uncertainty during the temporal evolution. We argue that the evolution of entity representations has randomness, because the features of an entity at a certain time are not completely determined by the past information. For example, (*Steve Jobs, diedIn, California*) happened on 2011-10-05. The semantic characteristics of this entity should have a sudden change at this time point. However, due to the incompleteness of knowledge in KGs, this change can not be predicted only according to its past evolutionary trend. Therefore, the representation of *Steve Jobs* is supposed to include some random components to handle this uncertainty, e.g. a Gaussian noise component.

In order to address the above problems, in this paper, we propose a temporal KG embedding model, ATiSE[1], which uses additive time series decomposition to capture the evolution process of KG representations. ATiSE fits the evolution process of an entity or relation as a multi-dimensional additive time series which composes of a trend component, a seasonal component and a random component. Our approach represents each entity and relation as a multi-dimensional Gaussian distribution at each time step to introduce a random component. The mean of an entity/relation representation at a certain time step indicates its current expected position, which is obtained from its initial representation, its linear change term, and its seasonality term. The covariance which describes

[1] The code is available at https://github.com/soledad921/ATISE.

the temporal uncertainty during its evolution, is denoted as a constant diagonal matrix for computing efficiency. Our contributions are as follows.

– Learning the representations for temporal KGs is a relatively unexplored problem because most existing KG embedding models only learn from time-unknown facts. We propose ATiSE, a new KG embedding model to incorporate time information into the KG representations.
– We specially consider the temporal uncertainty during the evolution process of KG representations. Thus, we model each entity/relation as a Gaussian distribution at each time step. As shown in Fig. 1, the mean vectors of multi-dimensional Gaussian distributions of entities and relations indicate their position which changes over time and the covariance matrices indicate the corresponding temporal uncertainty. A symmetric KL-divergence between two Gaussian distributions is designed to compute the scores of facts for optimization.
– Different from the previous temporal KG embedding models which use time embedding to incorporate time information, ATiSE fits the evolution process of KG representations as a multi-dimensional additive time series. Our work establishes a previously unexplored connection between relational processes and time series analysis with a potential to open a new direction of research on reasoning over time.
– Our experimental results show that ATiSE significantly outperforms other TKG models and some state-of-the-art static KGE on link prediction over four TKG datasets.

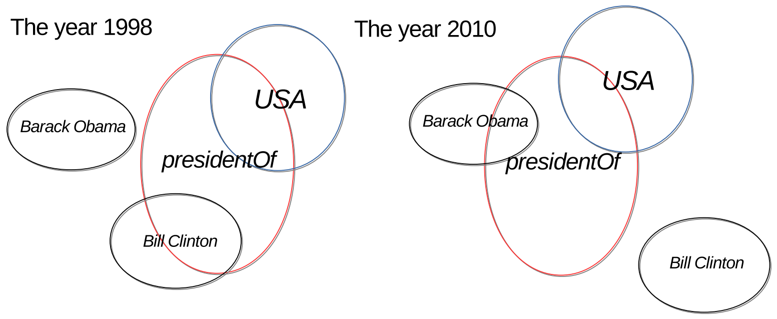

Fig. 1. Illustration of the means and (diagonal) variances of entities and relations in a temporal Gaussian Embedding Space. The labels indicate their position. In the representations, we might infer that *Bill Clinton* was *presidentOf USA* in 1998 and *Barack Obama* was *presidentOf USA* in 2010.

The rest of the paper is organized as follows: In the Sect. 2, we first review related works; in the Sect. 3, we introduce the architecture and the learning process of our proposed models; in the Sect. 4, we compare the performance of our models with the state-of-the-art models; in the Sect. 5, we make a conclusion in the end of this paper.

2 Related Work

A large amount of research has been done in KG embeddings [26]. A few examples of state-of-the-art KGE models include TransE [4], TransH [27], TransComplEx [20], RotatE [23], DistMult [28], ComplEx [25], ComplEx-N3 [13] and QuatE [30].

The above methods achieve good results on link prediction in KGs. However, these time-unaware KGE models have limitations on reasoning over TKGs. More concretely, given two quadruples with the same subjects, predicates, objects and different time stamps, i.e., (*Barack Obama*, *presidentOf*, *USA*, 2010) and (*Barack Obama*, *presidentOf*, *USA*, 2020), static KGE models will model them with the same scores due to their ignorance of time information, while the validities of these two quadruples might be different.

Recent researches illustrate that the performances of KG embedding models can be further improved by incorporating time information in temporal KGs.

TAE [11] captures the temporal ordering that exists between some relation types as well as additional common-sense constraints to generate more accurate link predictions.

TTransE [15] and HyTE [5] adopt translational distance score functions and encode time information in the entity-relation low dimensional spaces with time embeddings and temporal hyperplanes.

Know-Evolve [24] models the occurrence of a fact as a temporal point process. However, this method is built on a problematic formulation when dealing with concurrent events, as shown in Sect. 4.3.

TA-TransE and TA-DistMult [7] utilize recurrent neural networks to learn time-aware representations of relations and use standard scoring functions from TransE and DistMult. These models can model time information in the form of time points with or without some particular temporal modifiers, i.e., '*occursSince*' and '*occursUntil*'.

DE-SimplE [8] incorporates time information into diachronic entity embeddings and achieves the state of the art results on event-based TKGs. However, same as TA-TransE and TA-DistMult, DE-SimplE can not model facts involving time intervals shaped like [2005, 2008].

Moreover, TEE [2] encodes representations of years into entity embeddings by aggregating the representations of the entities that occur in event-based descriptions of the years.

3 Our Method

In this section, we present a detailed description of our proposed method, ATiSE, which not only uses relational properties between entities in triples but also incorporates the associated temporal meta-data by using additive time series decomposition.

3.1 Additive Time Series Embedding Model

A time series is a series of time-oriented data. Time series analysis is widely used in many fields, ranging from economics and finance to managing production operations, to the analysis of political and social policy sessions [19]. An important technique for time series analysis is additive time series decomposition. This technique decomposes a time series Y_t into three components as follows,

$$Y_t = T_t + S_t + R_t. \tag{1}$$

where T_t, S_t and R_t denote the trend component, the seasonal component and the random component (i.e. "noise"), respectively. Figure 2 shows an instance of the additive time series decomposition of a time series.

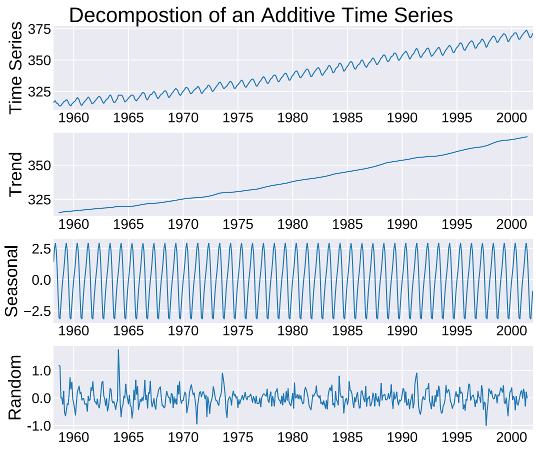

Fig. 2. Illustration of additive time series decomposition.

In our method, we regard the evolution of an entity/relation representation as an additive time series. For each entity/relation, we use a linear function and a Sine function to fit the trend component and the seasonal component respectively due to their simplicity. Considering the efficiency of model training, we model the irregular term by using a Gaussian noise instead of a moving average model (MA model) [10], since training an MA model requires a global optimization algorithm which will lead to more computation consumption.

To incorporate temporal information into traditional KGs, a new temporal dimension is added to fact triples, denoted as a quadruple (s, p, o, t). It represents the creation of relationship edge p between subject entity s, and object entity o at time step t. The score term $x_{spot} = f_t\left(e_s, r_p, e_o\right)$ can represent the conditional probability or the confidence value of this event x_{spot}, where $e_s, e_o \in \mathbf{R}^{L_e}$, $r_p \in \mathbf{R}^{L_r}$ are representations of s, o and p. In term of a long-term fact $(s, p, o, [t_s, t_e])$, we consider it to be a positive triple for each time step between t_s and t_e. t_s and t_e denote the start and end time during which the triple (s, p, o) is valid.

At each time step, the time-specific representations of an entity e_i or a relation r_p should be updated as $e_{i,t}$ or $r_{p,t}$. Thus, the score of a quadruple (s, p, o, t) can be represented as $x_{spot} = f_e\left(e_{s,t}, r_{p,t}, e_{o,t}\right)$ or $x_{spot} = f_r\left(e_s, r_{p,t}, e_o\right)$. We utilize additive time series decomposition to fit the evolution processes of each entity/relation representation as:

$$\begin{aligned} e_{i,t} &= e_i + \alpha_{e,i} w_{e,i} t + \beta_{e,i} \sin(2\pi\omega_{e,i} t) + \mathcal{N}(0, \varSigma_{e,i}) \\ r_{p,t} &= r_p + \alpha_{r,p} w_{r,p} t + \beta_{r,p} \sin(2\pi\omega_{r,p} t) + \mathcal{N}(0, \varSigma_{r,p}) \end{aligned} \tag{2}$$

where the e_i and r_p are the time-independent latent representations of the ith entity which is subjected to $||e_i||_2 = 1$ and the pth relation which is subjected to $||r_p||_2 = 1$. $e_i + \alpha_{e,i} w_{e,i} t$ and $r_p + \alpha_{r,p} w_{r,p} t$ are the trend components where the coefficients $|\alpha_{e,i}|$ and $|\alpha_{r,p}|$ denote the evolutionary rates of $e_{i,t}$ and $r_{p,t}$, the vectors $w_{e,i}$ and $w_{r,p}$ represents the corresponding evolutionary directions which are restricted to $||w_{e,i}||_2 = ||w_{r,p}||_2 = 1$. $\beta_{e,i}\sin(2\pi\omega_{e,i} t)$ and $\beta_{r,p}\sin(2\pi\omega_{r,p} t)$ are the corresponding seasonal components where $|\beta_{e,i}|$ and $|\beta_{r,p}|$ denote the amplitude vectors, $|\omega_{e,i}|$ and $|\omega_{r,p}|$ denote the frequency vectors. The Gaussian noise terms $\mathcal{N}(0, \varSigma_{e,i})$ and $\mathcal{N}(0, \varSigma_{r,p})$ are the random components, where $\varSigma_{e,i}$ and $\varSigma_{r,p}$ denote the corresponding diagonal covariance matrices.

In other words, for a fact (s, p, o, t), entity embeddings $e_{s,t}$ and $e_{o,t}$ obey Gaussian probability distributions: $\mathcal{P}_{s,t} \sim \mathcal{N}\left(\overline{e}_{s,t}, \varSigma_s\right)$ and $\mathcal{P}_{o,t} \sim \mathcal{N}\left(\overline{e}_{o,t}, \varSigma_o\right)$, where $\overline{e}_{s,t}$ and $\overline{e}_{o,t}$ are the mean vectors of $e_{s,t}$ and $e_{o,t}$, which do not include the random components. Similarly, the predicate is represented as $\mathcal{P}_{r,t} \sim \mathcal{N}\left(r_p, \varSigma_r\right)$.

Similar to translation-based KGE models, we consider the transformation result of ATiSE from the subject to the object to be akin to the predicate in a positive fact. We use the following formula to express this transformation: $\mathcal{P}_{s,t} - \mathcal{P}_{o,t}$, which corresponds to the probability distribution $\mathcal{P}_{e,t} \sim \mathcal{N}\left(\mu_{e,t}, \varSigma_e\right)$. Here, $\mu_{e,t} = \overline{e}_{s,t} - \overline{e}_{o,t}$ and $\varSigma_e = \varSigma_s + \varSigma_o$. Combined with the probability of relation $\mathcal{P}_{r,t} \sim \mathcal{N}\left(r_{p,t}, \varSigma_r\right)$, we measure the similarity between $\mathcal{P}_{e,t}$ and $\mathcal{P}_r$ to score the fact.

KL divergence is a straightforward method of measuring the similarity of two probability distributions. We optimize the following score function based on the KL divergence between the entity-transformed distribution and relation distribution [29].

$$\begin{aligned} x_{spot} &= f_t\left(e_s, r_p, e_o\right) = \mathcal{D}_{\mathcal{KL}}(P_{r,t}, P_{e,t}) \\ &= \int_{x \in \mathcal{R}^{k_e}} \mathcal{N}(x; r_{p,t}, \varSigma_r) \log \frac{\mathcal{N}(x; \mu_{e,t}, \varSigma_e)}{\mathcal{N}(x; r_{p,t}, \varSigma_r)} dx \\ &= \frac{1}{2}\Big\{ tr(\varSigma_r^{-1}\varSigma_e) + (r_{p,t} - \mu_{e,t})^T \varSigma_r^{-1} (r_{p,t} - \mu_{e,t}) \\ &\quad - \log\frac{det(\varSigma_e)}{det(\varSigma_r)} - k_e \Big\} \end{aligned} \tag{3}$$

where, $tr(\varSigma)$ and $\varSigma^{-1}$ indicate the trace and inverse of the diagonal covariance matrix, respectively.

Since the computation of the determinants of the covariance matrices in Eq. 3 is time consuming, we define a symmetric similarity measure based on KL divergence to simplify the computation of the score function.

$$f_t\left(e_s, r_p, e_o\right) = \frac{1}{2}(\mathcal{D}_{\mathcal{KL}}(P_{r,t}, P_{e,t}) + \mathcal{D}_{\mathcal{KL}}(P_{e,t}, P_{r,t})) \tag{4}$$

Considering the simplified diagonal covariance, we can compute the trace and inverse of the matrix simply and effectively for ATiSE. The gradient of log determinant is $\frac{\partial \log det A}{\partial A} = A^{-1}$, the gradient $\frac{\partial x^T A^{-1} y}{\partial A} = -A^{-T} x y^T A^{-T}$, and the gradient $\frac{\partial tr(X^T A^{-1} Y)}{\partial A} = -(A^{-1} Y X^T A^{-1})^T$ [21].

3.2 Complexity

In Table 1, we summarize the scoring functions of several existing (T)KGE approaches and our models and compare their space complexities. n_e, n_r, n_t and n_{token} are numbers of entities, relations, time steps and temporal tokens used in [7]; d is the dimensionality of embeddings. $\langle x, y, z\rangle = \sum_i x_i y_i z_i$ denotes the tri-linear dot product; $\textsc{Re}(\cdot)$ denotes the real part of the complex embedding [25]; $\otimes$ denotes the Hamilton product between quaternion embeddings; $\triangleleft$ denotes the normalization of the quaternion embedding. $\mathcal{P}_t$ denotes the temporal projection for embeddings [5]; LSTM$(\cdot)$ denotes an LSTM neural network; $[r_p; t_{seq}]$ denotes the concatenation of the relation embedding and the sequence of temporal tokens [7]; $\overrightarrow{e}$ and $\overleftarrow{e}$ denote the temporal part and untemporal part of a time-specific diachronic entity embedding e^t [8]; p^{-1} denotes the inverse relation of p, i.e., $(s, p, o, t) \leftrightarrow (o, p^{-1}, s, t)$.

Table 1. Comparison of our models with several baseline models for space complexity.

Model	Scoring function	Space complexity	Time complexity
TransE	$\|\|e_s + r_p - e_o\|\|$	$\mathcal{O}(n_e d + n_r d)$	$\mathcal{O}(d)$
DistMult	$\langle e_s, r_p, e_o\rangle$	$\mathcal{O}(n_e d + n_r d)$	$\mathcal{O}(d)$
ComplEx	$\textsc{Re}(\langle e_s, r_p, \overline{e}_o\rangle)$	$\mathcal{O}(n_e d + n_r d)$	$\mathcal{O}(d)$
RotatE	$\|\|e_s \circ r_p - e_o\|\|$	$\mathcal{O}(n_e d + n_r d)$	$\mathcal{O}(d)$
QuatE	$e_s \otimes r_p^{\triangleleft} \cdot e_o$	$\mathcal{O}(n_e d + n_r d)$	$\mathcal{O}(d)$
TTransE	$\|\|e_s + r_p + w_t - e_o\|\|$	$\mathcal{O}(n_e d + n_r d + n_t d)$	$\mathcal{O}(d)$
HyTE	$\|\|P_t(e_s) + P_t(r_p) - P_t(e_o)\|\|$	$\mathcal{O}(n_e d + n_r d + n_t d)$	$\mathcal{O}(d)$
TA-TransE	$\|\|e_s + \text{LSTM}([r_p; t_{seq}]) - e_o\|\|$	$\mathcal{O}(n_e d + n_r d + n_{token} d)$	$\mathcal{O}(d)$
TA-DistMult	$\langle e_s, \text{LSTM}([r_p; t_{seq}]), e_o\rangle$	$\mathcal{O}(n_e d + n_r d + n_{token} d)$	$\mathcal{O}(d)$
DE-SimplE	$\frac{1}{2}(\langle \overrightarrow{e}^t_s, r_p, \overleftarrow{e}^t_o\rangle + \langle \overrightarrow{e}^t_0, r_{p^{-1}}, \overleftarrow{e}^t_s\rangle)$	$\mathcal{O}(n_e d + n_r d)$	$\mathcal{O}(d)$
ATiSE	$\mathcal{D}_{\mathcal{KL}}(\mathcal{P}_{e,t}, \mathcal{P}_{r,t})$	$\mathcal{O}(n_e d + n_r d)$	$\mathcal{O}(d)$

As shown in Sect. 3.2, our model has the same space complexity and time complexity as static KGE models listed in Sect. 3.2 as well as DE-SimplE. On the other hand, the space complexities of TTransE, HyTE, TA-TransE or TA-DistMult will be higher than our models if n_t or n_{token} is much larger than n_e and n_r.

3.3 Learning

In this paper, we use the same loss function as the negative sampling loss proposed in [23] for optimizing ATiSE. This loss function has been proved to be more effective than the margin rank loss function proposed in [4] on optimizing translation-based KGE models.

$$\mathcal{L} = \sum_{t \in [T]} \sum_{\xi \in \mathcal{D}_t^+} \sum_{\xi' \in \mathcal{D}_t^-} -\log\, \sigma(\gamma - f_t(\xi)) - \log\, \sigma(f_t(\xi^{'}) - \gamma) \tag{5}$$

where, $[T]$ is the set of time steps in the temporal KG, $\mathcal{D}_t^+$ is the set of positive triples with time stamp t, and $\mathcal{D}_t^-$ is the set of negative sample corresponding to $\mathcal{D}_t^+$. In this paper, we generate negative samples by randomly corrupting subjects or objects of the positives such as $(s^{'}, p, o, t)$ and $(s, p, o^{'}, t)$. Moreover, we adopt self-adversarial training proposed in [23] and reciprocal learning used in [8,13,30] to further enhance the performances of our model. To avoid overfitting, we add some regularizations while learning ATiSE. As described in the Sect. 3.1, the norms of the original representations of entities and relations, as well as the norms of all evolutionary direction vectors, are restricted by 1. Besides, the following constraint is used for guaranteeing that the covariance matrices are positive definite and of appropriate size when we minimize the loss:

$$\forall l \in \mathcal{E} \cup \mathcal{R}, c_{min} I \leq \Sigma_l \leq c_{max} I \tag{6}$$

where, $\mathcal{E}$ and $\mathcal{R}$ are the set of entities and relations respectively, c_{min} and c_{max} are two positive constants. We use $\Sigma_l \leftarrow \max(c_{min}, \min(c_{max}, \Sigma_l))$ to achieve this regularization for diagonal covariance matrices. This constraint 6 for the covariance is considered during both the initialization and training process.

Algorithm: The learning algorithm of **ATiSE**

```
    input: The training set $\mathcal{D}^+ = \{(s,p,o,t)\}$, entity set $\mathcal{E}$, relation set $\mathcal{R}$, embedding
    dimensionality $d$, margin $\gamma$, batch size $b$, the ratio of negative samples over the
    positives $\eta$, learning rate $lr$, restriction values $c_{min}$ and $c_{max}$ for covariance, and
    a score function $f_t(e_s, r_p, e_o)$ where $s, o \in \mathcal{E}$, $p \in \mathcal{R}$.
    output: Time-independent embeddings for each entity $e_i$ and relation $r_j$ (the
    mean vectors and the covariance matrices), the evolutionary rate $\alpha_i$and the
    evolutionary direction vector $w_i$ for each entity, where $i \in \mathcal{E}$, $j \in \mathcal{R}$.
1.  initialize $e_i, r_j \leftarrow$ uniform $(-\frac{6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$, $i \in \mathcal{E}$, $j \in \mathcal{R}$
2.              $w_{e,i}, w_{r,j} \leftarrow$ uniform $(-\frac{6}{\sqrt{d}}, \frac{6}{\sqrt{d}})$, $i \in \mathcal{E}$, $j \in \mathcal{R}$
3.              $\Sigma_{e,i}, \Sigma_{r,j} \leftarrow$ uniform $(c_{min}, c_{max})$, $i \in \mathcal{E}$, $j \in \mathcal{R}$
4.              $\omega_{e,i}, \omega_{r,j} \leftarrow$ uniform $(c_{min}, c_{max})$, $i \in \mathcal{E}$, $j \in \mathcal{R}$
5.              $\alpha_{e,i}, \alpha_{r,j} \leftarrow$ uniform $(0, 0)$, $i \in \mathcal{E}$, $j \in \mathcal{R}$
6.              $\beta_{e,i}, \beta r, j \leftarrow$ uniform $(0, 0)$, $i \in \mathcal{E}$, $j \in \mathcal{R}$
7.  loop
8.    $e_i \leftarrow e_i / ||e_i||_2$, $i \in \mathcal{E}$
9.    $r_j \leftarrow r_j / ||r_j||_2$, $j \in \mathcal{R}$
10.   $w_{e,i} \leftarrow w_{e,i} / ||w_{e,i}||_2$, $i \in \mathcal{E}$
11.   $w_{r,j} \leftarrow w_{r,j} / ||w_{r,j}||_2$, $j \in \mathcal{R}$
12.   $\mathcal{D}_b^+ \leftarrow$ sample$(\mathcal{D}^+, b)$ // sample a minibatch
13.   for $(s,p,o,t) \in \mathcal{D}_b^+$ do
14.     $\mathcal{D}_b^- = \{(s_k^{'}, p, o_k^{'}, t)\}_{k=1...\eta}$ // generate $\eta$ negative samples
15.   end for
16.   Update $e_i$, $w_i$, $\alpha_i$ and $r_j$ based on Equation 4 and 5 w.r.t.
      $\mathcal{L} = \sum_{\xi \in \mathcal{D}_b^+} \sum_{\xi' \in \mathcal{D}_b^-} -\log\ \sigma(\gamma - f_t(\xi)) - \log\ \sigma(f_t(\xi^{'}) - \gamma)$
17.   regularize the covariances for each entity and relation based on Constraint 6,
      $\Sigma_{e,i} \leftarrow max(cmin, min(cmax, \Sigma_{e,i}))$, $i \in \mathcal{E}$
      $\Sigma_{r,j} \leftarrow max(cmin, min(cmax, \Sigma_{r,j}))$, $j \in \mathcal{R}$
18. end loop
```

4 Experiment

To show the capability of ATiSE, we compared it with some state-of-the-art KGE models and the existing TKGE models on link prediction over four TKG datasets. Particularly, we also did an ablation study to analyze the effect of the dimensionality of entity/relation embeddings and various components of the additive time series decomposition.

4.1 Datasets

As mentioned in Sect. 1, common TKGs include ICEWS [14], Wikidata [6] and YAGO3 [17]. Four subsets of these TKGs are used as datasets in [7], i.e., ICEWS14, ICEWS05-15, YAGO15k and Wikidata11k. However, all of time intervals in YAGO15k and Wikidata11k only contain either start dates or end dates, shaped like *'occursSince 2003'* or *'occursUntil 2005'* while most of time intervals in Wikidata and YAGO are presented by both start dates and end dates. -Thus, we prefer using YAGO11k and Wikidata12k released in [5] instead of YAGO15k and Wikidta12k. The statistics of the datasets used in this paper are listed in Table 2.

Table 2. Statistics of datasets.

	#Entities	#Relations	#Time steps	Time span	#Training	#Validation	#Test
ICEWS14	6,869	230	365	2014	72,826	8,941	8,963
ICEWS05-15	10,094	251	4,017	2005–2015	368,962	46,275	46,092
YAGO11k	10,623	10	70	−453−2844	16,408	2,050	2,051
Wikidata12k	12,554	24	81	1709–2018	32,497	4,062	4,062

Table 3. Statistics of long-term facts

	#Long-term relations	#Training	#Validation	#Test
YAGO11k	8	12,579	1,470	1,442
Wikidata12k	20	18,398	2,194	2,200

ICEWS is a repository that contains political events with specific time annotations, e.g., (*Barack Obama*, *visits*, *Ukraine*, *2014-07-08*). ICEWS14 and ICEWS05-15 are subsets of ICEWS [14], which correspond to the facts in 2014 and the facts between 2005 to 2015. These two datasets are filtered by only selecting the most frequently occurring entities in the graph [7]. It is noteworthy that all of time annotations in ICEWS datasets are time points.

YAGO11k is a subset of YAGO3 [17]. Different from ICEWS, a part of time annotations in YAGO3 are represented as time intervals, e.g. (*Paul Konchesky*, *playsFor*, *England national football team*, [2003-##-##, 2005-##-##]). Following the setting used in HyTE [5], we only deal with year level granularity by dropping the month and date information and treat timestamps as 70 different time steps in the consideration of the balance about numbers of triples in different time steps. For a time interval with the missing start date or end date, e.g., [2003-##-##, ####-##-##] representing 'since 2003', we use the first timestep or the last timestep to represent the missing start time or end time.

Wikidata12k is a subset of Wikidata [6]. Similar to YAGO11k, Wikidata12k contains some facts involving time intervals. We treat timestamps as 81 different time steps by using the same setting as YAGO11k.

As shown in Table 3, most of facts in YAGO11k and Wikidata12k involve time intervals. For TKGE models, we discretized such facts $(s, p, o, [t_s, t_e])$ involving multiple timesteps into multiple quadruples which only involve single timesteps, i.e., $\{(s, p, o, t_s), (s, p, o, t_{s+1}), \cdots, (s, p, o, t_e)\}$, where t_s and t_e denote the start time and the end time.

4.2 Evaluation Metrics

We evaluate our model by testing the performances of our model on link prediction task over TKGs. This task is to complete a time-wise fact with a missing entity. For a test quadruple (s, p, o, t), we generate corrupted triples by replacing

s or o with all possible entities. We sort scores of all the quadruples including corrupted quadruples and the test quadruples and obtain the ranks of the test quadruples. For a test fact involving multiple time steps, e.g., $(s, p, o, [t_s, t_e])$, the score of one corrupted fact $(s, p, o', [t_s, t_e])$ is the sum of scores of multiple discreet quadruples, $\{(s, p, o', t_s), (s, p, o', t_{s+1}), \cdots, (s, p, o', t_e)\}$.

Two evaluation metrics are used here, i.e., Mean Reciprocal Rank and Hits@k. The Mean Reciprocal Rank (MRR) is the means of the reciprocal values of all computed ranks. And the fraction of test quadruples ranking in the top k is called Hits@k. We adopt the time-wise filtered setting used in source code released by [8]. Different from the original filtered setting proposed in [4], for a test fact (s, p, o, t) or $(s, p, o, [t_s, t_e])$, instead of removing all the triples that appear either in the training, validation or test set from the list of corrupted facts, we only filter the triples that occur at the time point t or throughout the time interval $[t_s, t_e]$ from the list of corrupted facts. This ensures that the facts that do not appear at t or throughout $[t_s, t_e]$ are still considered as corrupted triplets for evaluating the given test fact.

4.3 Baselines

We compare our approach with several state-of-the-art KGE approaches and existing TKGE approaches, including TransE [4], DistMult [28], ComplEx-N3 [13], RotatE [23], QuatE^2 [30], TTransE [15], TA-TransE, TA-DistMult [7] and DE-SimplE [8]. ComplEx-N3 has been proven to have better performance than ComplEx [25] on FreeBase and WordNet datasets. And QuatE^2 has the best performances among all variants of QuatE as reported in [30].

As mentioned in Sect. 2, TA-TransE, TA-DistMult and DE-SimplE mainly focus on modeling temporal facts involving time points with or without some particular temporal modifiers, *'occursSince'* and *'occursUntil'*, and cannot model time intervals shaped like [2003-##-##, 2005-##-##]. Besides, DE-SimplE needs specific date information including year, month and day to score temporal facts, while most of time annotations in YAGO and Wikidataset only contain year-level information. Thus, we cannot test these three models on YAGO11k and Wikidataset15k.

We do not take Know-Evolve [24] as baseline model due to its problematic formulation and implementation issues. Know-Evolve uses the temporal point process to model the temporal evolution of each entity. The intensity function of Know-Evolve (equation 3 in [24]) is defined as $\lambda_r^{s,o}(t|\bar{t}) = f(g_r^{s,o}(\bar{t}))(t - \bar{t})$, where $g(\cdot)$ is a score function, t is current time, and $\bar{t}$ is the most recent time point when either subject or object entity was involved in an event. This intensity function is used in inference to rank entity candidates. However, they don't consider concurrent event at the same time stamps, and thus $\bar{t}$ will become t after one event. For example, we have events $event_1 = (s, r, o_1, t_1)$, $event_2 = (s, r, o_2, t_1)$. After $event_1$, $\bar{t}$ will become t (subject s's most recent time point), and thus the value of intensity function for $event_2$ will be 0. This is problematic in inference since if $t = \bar{t}$, then the intensity function will always be 0 regardless of entity candidates. In their code, they give the highest ranks (first rank) for all entities including the ground truth object in this case, which we think is unfair since the scores of many entity candidates including the ground truth object might be 0 due to their formulation. It has been proven that the performances of Know-Evolve on ICEWS datasets drop down to almost zero after this issue fixed [12].

4.4 Experimental Setup

We used Adam optimizer to train our model and selected the optimal hyper-parameters by early validation stopping according to MRR on the validation set. We restricted the maximum epoch to 5000. We fixed the mini-batch size b as 512. We tuned the embedding dimensionalities d in $\{100, 200, 300, 400, 500\}$, the ratio of negatives over positive training samples η in $\{1, 3, 5, 10\}$ and the learning rate lr in $\{0.00003, 0.0001, 0.0003, 0.001\}$. The margins γ were varied in the range $\{1,\ 2,\ 3,\ 5,\ 10,\ 20,\ \cdots,\ 120\}$. We selected the pair of restriction values c_{min} and c_{max} for covariance among $\{(0.0001,\ 0.1),\ (0.003,\ 0.3),\ (0.005,\ 0.5),\ (0.01,\ 1)\}$. The default configuration for ATiSE is as follows: $lr = 0.00003$, $d = 500$, $\eta = 10$, $\gamma = 1$, $(c_{min}, c_{max}) = (0.005, 0.5)$. Below, we only list the non-default parameters: $\gamma = 120$, $(c_{min}, c_{max}) = (0.003, 0.3)$ on ICEWS14; $\gamma = 100$, $(c_{min}, c_{max}) = (0.003, 0.3)$ on ICEWS05-15.

4.5 Experimental Results

Table 4 and 5 show the results for link prediction task. On ICEWS14 and ICEWS05-15, ATiSE outperformed all baseline models, considering MR, MRR, Hits@10 and Hits@1. Compared to DE-SimplE which is a very recent state-of-the-art TKGE model, ATiSE got improvement of 4% on both datasets regarding MRR, and improved Hits@10 by 4% and 6% on ICEWS14 and ICEWS05-15 respectively. On YAGO11k and Wikidata12k where time annotations in facts are time intervals, ATiSE surpassed baseline models regarding MRR, Hits@1, Hits@3. Regarding Hits@10, ATiSE achieved the state-of-the-art results on Wikidata12k and the second best results on YAGO11k. As mentioned in Sect. 4.3, the results of TA-TransE, TA-DistMult and DE-SimplE on YAGO11k and Wikidata12k are unobtainable since they have difficulties in modeling facts involving time intervals in these two datasets.

A part of results listed on Table 4 and 5 are obtained based on the implementations released in [5,13,23]. We list the implementation details of some baseline models as follows:

– We used the implementation released in [23] to test RotatE on all four datasets, and DistMult on YAGO11k and Wikidata12k. The source code was revised to adopt the time-wise filtered setting. To search the optimal configurations for RotatE and DistMult, we followed the experimental setups reported in [23] except setting the maximum dimensionality as 500 and the maximum negative sampling ratio as 10. The default optimal configuration for RotatE and DistMult is as follows: $lr = 0.0001$, $b = 1024$, $d = 500$, $\eta = 10$.

Table 4. Link prediction results on ICEWS14 and ICEWS05-15. *: results are taken from [7]. $^\diamond$: results are taken from [8]. Dashes: results are unobtainable. The best results among all models are written bold.

	ICEWS14				ICEWS05-15			
Metrics	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE*	.280	.094	–	.637	.294	.090	–	.663
DistMult*	.439	.323	–	.672	.456	.337	–	.691
ComplEx-N3	.467	.347	.527	.716	.481	.362	.535	.729
RotatE	.418	.291	.478	.690	.304	.164	.355	.595
QuatE^2	.471	.353	.530	.712	.482	.370	.529	.727
TTransE$^\diamond$	.255	.074	–	.601	.271	.084	–	.616
HyTE$^\diamond$	.297	.108	.416	.655	.316	.116	.445	.681
TA-TransE*	.275	.095	–	.625	.299	.096	–	.668
TA-DistMult*	.477	.363	–	.686	.474	.346	–	.728
DE-SimplE$^\diamond$	.526	.418	.592	.725	.513	.392	.578	.748
ATiSE	**.545**	**.423**	**.632**	**.757**	**.533**	**.394**	**.623**	**.803**

Table 5. Link prediction results on YAGO11k and Wikidata12k. The best results among all models are written bold.

	YAGO11k				Wikidata12k			
Metrics	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	.100	.015	.138	.244	.178	.100	.192	.339
DistMult	.158	.107	.161	.268	.222	.119	.238	.460
ComplEx-N3	.167	.106	.154	.282	.233	.123	.253	.436
RotatE	.177	.113	.177	**.315**	.221	.116	.236	.461
QuatE^2	.164	.107	.148	.270	.230	.125	.243	.416
TTransE	.108	.020	.150	.251	.172	.096	.184	.329
HyTE	.105	.015	.143	.272	.180	.098	.197	.333
ATiSE	**.185**	**.126**	**.189**	.301	**.252**	**.148**	**.288**	**.462**

Below, we only list the non-default parameters: for RotatE, the optimal margins are $\gamma = 36$ on ICEWS14, $\gamma = 48$ on ICEWS05-15, $\gamma = 3$ on YAGO11k and $\gamma = 6$ on Wikidata12k; for DistMult, the optimal regularizer weights are $r = 0.00001$ on YAGO11k and Wikidata12k.

– We used the implementation released in [13] to test ComplEx-N3 and QuatE^2 on all four datasets. The source code was revised to adopt the time-wise filtered setting. To search the optimal configurations for ComplEx-N3 and QuatE^2, we followed the experimental setups reported in [13] except setting the maximum dimensionality as 500. The default optimal configuration for ComplEx-N3 and QuatE^2 is as follows: $lr = 0.1$, $d = 500$, $b = 1000$. Below, we

list the optimal regularizer weights: for ComplEx-N3, $r = 0.01$ on ICEWS14 and ICEWS05-15, $r = 0.1$ on YAGO11k and Wikidata12k; for QuatE^2, $r = 0.01$ on ICEWS14 and YAGO11k, $r = 0.05$ on ICEWS05-15, $r = 0.1$ on Wikidata.
– We used the implementation released in [5] to test TransE, TTransE and HyTE on YAGO11k and Wikidata12k for obtaining their performances regarding MRR, Hits@1 and Hits@3. We followed the optimal configurations reported in [5]. As shown in Table 5, Hits@10s of TransE and TTransE we got were better than those reported in [5].
– As shown in Table 4, other baseline results are taken from [7,8].

4.6 Ablation Study

In this work, we analyze the effects of the dimensionality and various components of entity/relation embeddings.

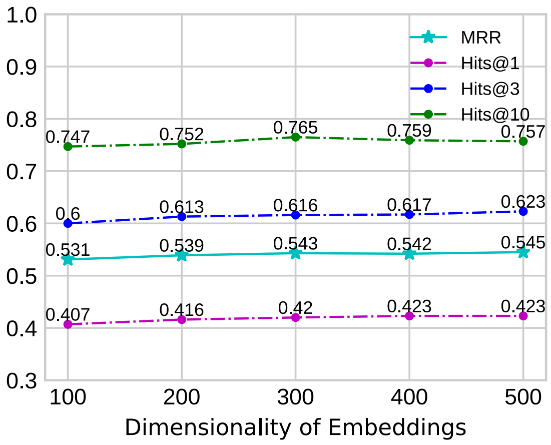

Fig. 3. Results for ATiSE with different embedding dimensionalities on ICEWS14.

The embedding dimensionality is an important hyperparameter for each (T)KGE model. A high embedding dimensionality might be beneficial to boost the performance of a (T)KGE model. For instance, ComplEx-N3 and QuatE^2 achieved the state-of-the-art results on link prediction over static KGs with 2000-dimensional embeddings [13,30]. On the other hand, a lower embedding dimensionality will lead to less consumption on training time and memory space, which is quite important for the applications of (T)KGE models on large-scale datasets. Figure 3 shows the performances of ATiSE with different embedding dimensionalities on ICEWS14. With a same embedding dimensionality of 100 as DE-SimplE [8], ATiSE still achieved the state-of-the-art results on ICEWS14. An ATiSE model with an embedding dimensionality of 100 trained on ICEWS14 had a memory size of 14.2 Mb while a DE-SimplE model and a QuatE^2 model with the same embedding dimensionality had memory sizes of 13.3 Mb and 12.4 Mb. And the memory size of an ATiSE model increases linearly with its embedding dimensionality. Moreover, training an ATiSE model with an embedding dimensionality of 100 took 2.8 s per epoch on a single GeForce RTX2080, and an ATiSE with 500-dimensional embeddings took 3.7 s per epoch.

To analyze the effects of different components of entity/relation representation in ATiSE, we developed three comparison models, namely, ATiSE-SN, ATiSE-TN and ATiSE-TS, which exclude the trend component, seasonal component and the noise component respectively. The entity representations of these three comparison models are as follows:

$$\begin{aligned} e_{i,t}^{SN} &= e_i + \beta_{e,i}\sin(2\pi\omega_{e,i}t) + \mathcal{N}(0, \Sigma_{e,i}) \\ e_{i,t}^{TN} &= e_i + \alpha_{e,i}w_{e,i}t + \mathcal{N}(0, \Sigma_{e,i}) \\ e_{i,t}^{TS} &= e_i + \alpha_{e,i}w_{e,i}t + \beta_{e,i}\sin(2\pi\omega_{e,i}t) \end{aligned} \tag{7}$$

For ATiSE-TS consisting of the trend component and the seasonal component, we used the translation-based scoring function [4] to measure the plausibility of the fact (s, p, o, t).

$$f_t^{TS}(e_s, r_p, e_o) = ||e_{s,t}^{TS} + r_{p,t}^{TS} - e_{o,t}^{TS}|| \tag{8}$$

We report the MRRs and Hits@10 of ATiSE-SN, ATiSE-TN and ATiSE-TS on link prediction over ICEWS14 and YAGO11k. As shown in Table 6, we find that the removal of the trend component and the noise component had a remarkable negative effect on the performance of ATiSE on link prediction since the model could not address the temporal uncertainty of entity/relation representations without the noise component and the trend component contained the main time information. In ATiSE, different types of entities might have big difference in the trend component. For instance, we found that the embeddings of entities representing people, e.g., *Barack Obama*, generally had higher evolution rates than those representing cities or nations, e.g., *USA*.

Table 6. Link prediction results of ablation experiments.

Datasets	ICEWS14				YAGO11K$_D$			
Metrics	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
ATiSE-SN	.405	.284	.488	.710	.139	.095	.143	.249
ATiSE-TN	.536	.407	.626	.771	.167	.115	.171	.292
ATiSE-TS	.323	.127	.429	.676	.115	.023	.145	.274
ATiSE	**.545**	**.423**	**.632**	**.757**	**.185**	**.126**	**.189**	**301**

ATiSE-TN performed worse than ATiSE on YAGO11k where facts involve time intervals. Different from ICEWS14 dataset which is an event-based dataset where all relations or predicates are instantaneous, there exist both short-term relations and long-term relations in YAGO11k. Adding seasonal components into evolving entity/relation representations is helpful to distinguish short-term patterns and long-term patterns in YAGO11k. It can be seen from Table 7 that short-term relations learned by ATiSE, e.g., *wasBornIn*, generally had higher evolutionary rates, and their seasonal components had smaller amplitudes and higher frequencies than long-term relations, e.g., *isMarriedTo*.

Table 7. Relations in YAGO11k and the mean step numbers of their duration time (TS), as well as the corresponding parameters learned from ATiSE, including the evolutionary rate $|\alpha_r|$, the mean amplitude $\overline{|\beta_r|}$ and the mean frequency $\overline{|\omega_r|}$ of the seasonal component for each relation.

Relations	#TS	$\lvert\alpha_r\rvert$	$\overline{\lvert\beta_r\rvert}$	$\overline{\lvert\omega_r\rvert}$
wasBornIn	1.0	0.142	0.000	1.032
worksAt	18.7	0.046	0.058	0.294
playsFor	4.7	0.071	0.046	0.766
hasWonPrize	28.6	0.010	0.107	0.041
isMarriedTo	16.5	0.049	0.076	0.090
owns	24.9	0.017	0.088	0.101
graduatedFrom	38.1	0.016	0.104	0.029
deadIn	1.0	0.249	0.006	0.897
isAffliatedTo	25.8	0.014	0.049	0.126
created	27.1	0.011	0.040	0.087

5 Conclusion

We introduce ATiSE, a temporal KGE model that incorporates time information into KG representations by fitting the temporal evolution of entity/relation representations over time as additive time series. Considering the uncertainty during the temporal evolution of KG representations, ATiSE maps the representations of temporal KGs into the space of multi-dimensional Gaussian distributions. The covariance of an entity/relation representation represents its randomness component. Experimental results demonstrate that our method significantly outperforms the state-of-the-art methods on link prediction over four TKG benchmarks.

Our work establishes a previously unexplored connection between relational processes and time series analysis with a potential to open a new direction of research on reasoning over time. In the future, we will explore to use more sophisticated models to model different components of relation/entity representations, e.g., an ARIMA model for the noise component and a polynomial model for the trend component.

Acknowledgements. This work is supported by the CLEOPATRA project (GA no. 812997), the German national funded BmBF project MLwin and the BOOST project.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
2. Bianchi, F., Palmonari, M., Nozza, D.: Towards encoding time in text-based entity embeddings. In: Vrandečić, D., et al. (eds.) ISWC 2018. LNCS, vol. 11136, pp. 56–71. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00671-6_4
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. ACM (2008)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795 (2013)
5. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HyTE: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2001–2011 (2018)
6. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing Wikidata to the linked data web. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 50–65. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11964-9_4
7. García-Durán, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. In: EMNLP (2018)
8. Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P.: Diachronic embedding for temporal knowledge graph completion. In: AAAI (2020)
9. He, S., Liu, K., Ji, G., Zhao, J.: Learning to represent knowledge graphs with Gaussian embedding. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 623–632. ACM (2015)
10. Ho, S., Xie, M.: The use of ARIMA models for reliability forecasting and analysis. Comput. Ind. Eng. **35**(1–2), 213–216 (1998)
11. Jiang, T., et al.: Towards time-aware knowledge graph completion. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1715–1724 (2016)
12. Jin, W., et al.: Recurrent event network: global structure inference over temporal knowledge graph. arXiv:1904.05530 (2019)
13. Lacroix, T., Usunier, N., Obozinski, G.: Canonical tensor decomposition for knowledge base completion. In: International Conference on Machine Learning, pp. 2869–2878 (2018)
14. Lautenschlager, J., Shellman, S., Ward, M.: ICEWS event aggregations (2015). https://doi.org/10.7910/DVN/28117
15. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Companion of the The Web Conference 2018 on The Web Conference 2018, pp. 1771–1776. International World Wide Web Conferences Steering Committee (2018)

16. Leetaru, K., Schrodt, P.A.: GDELT: global data on events, location, and tone, 1979–2012. In: ISA Annual Convention, vol. 2, pp. 1–49. Citeseer (2013)
17. Mahdisoltani, F., Biega, J., Suchanek, F.M.: YAGO3: a knowledge base from multilingual Wikipedias. In: CIDR (2013)
18. Miller, G.A.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
19. Montgomery, D.C., Jennings, C.L., Kulahci, M.: Introduction to Time Series Analysis and Forecasting. Wiley, Hoboken (2015)
20. Nayyeri, M., Xu, C., Yaghoobzadeh, Y., Yazdi, H.S., Lehmann, J.: Toward understanding the effect of loss function on the performance of knowledge graph embedding (2019)
21. Petersen, K.B., Pedersen, M.S., et al.: The matrix cookbook. Tech. Univ. Denmark **7**(15), 510 (2008)
22. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706. ACM (2007)
23. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: RotatE: knowledge graph embedding by relational rotation in complex space. In: ICLR (2019)
24. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-Evolve: deep temporal reasoning for dynamic knowledge graphs. In: ICML (2017)
25. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of ICML (2016)
26. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
27. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI, pp. 1112–1119. Citeseer (2014)
28. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR, p. 12 (2015)
29. Yu, D., Yao, K., Su, H., Li, G., Seide, F.: KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7893–7897. IEEE (2013)
30. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: Advances in Neural Information Processing Systems, pp. 2731–2741 (2019)