Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols

Prachi Jain[*1], **Sushant Rathi**[*1], **Mausam**[1] and **Soumen Chakrabarti**[2]
[1] Indian Institute of Technology Delhi
[2] Indian Institute of Technology Bombay
{p6.jain, rathisushant5}@gmail.com, mausam@cse.iitd.ac.in, soumen.chakrabarti@gmail.com

Abstract

Research on temporal knowledge bases, which associate a relational fact (s, r, o) with a validity time period (or time instant), is in its early days. Our work considers predicting missing entities (link prediction) and missing time intervals (time prediction) as joint Temporal Knowledge Base Completion (TKBC) tasks, and presents TIMEPLEX, a novel TKBC method, in which entities, relations and, time are all embedded in a uniform, compatible space. TIMEPLEX exploits the recurrent nature of some facts/events and temporal interactions between pairs of relations, yielding state-of-the-art results on both prediction tasks.

We also find that existing TKBC models heavily overestimate link prediction performance due to imperfect evaluation mechanisms. In response, we propose improved TKBC evaluation protocols for both link and time prediction tasks, dealing with subtle issues that arise from the partial overlap of time intervals in gold instances and system predictions.

1 Introduction

A knowledge base (KB) is a collection of triples (s, r, o), with a subject s, a relation type r and an object o. KBs are usually incomplete, necessitating completion (KBC), i.e., inferring facts not provided in the KB. A KBC model is often evaluated via link prediction: supplying missing arguments to queries of the form $(s, r, ?)$ and $(?, r, o)$.

Many relations are transient or impermanent. Temporal KBs annotate each fact (event) with the time period (instant) in which it holds (occurs) (Hoffart et al., 2013). A person is born in a city in an instant, a politician can be a country's president for several years, and a marriage may last between years and decades. Temporal KBs represent these by (s, r, o, T) tuples, where T is a span of time. Temporal KBC (TKBC) performs completion of temporal KBs. It is also primarily evaluated by link prediction queries $(s, r, ?, T)$ and $(?, r, o, T)$. Recently, time prediction $(s, r, o, ?)$ has also been considered for predicting time instants, but not time intervals (Lacroix et al., 2020).

While KBC has been intensely researched, TKBC is only beginning to be explored. TKBC presents novel challenges in task definition and modeling. For instance, little is known about how best to predict intervals for $(s, r, o, ?)$ queries, or how to evaluate a system response interval. Moreover, we show that even for link prediction queries, evaluation faces subtle complications owing to the inclusion of T in $(s, r, ?, T)$ queries and requires careful rethinking of evaluation protocols. In this paper, we propose improved evaluation protocols for both link and time prediction tasks in a TKBC.

TKBC also brings unique modeling opportunities. A TKBC system can learn typical durations of relation validity, or distributions over time gaps between events, from training data. E.g., a person must be born before becoming president, which must precede death. A nation rarely has two presidents at the same time. Such constraints can better inform both link and time predictions.

In response, we present TIMEPLEX, a novel TKBC model, which obtains state-of-the-art results on benchmark datasets for both link and time prediction. At a high level, TIMEPLEX performs tensor factorization of a temporal KB, using complex-valued embeddings for relations, entities and time points. It enables these embeddings to capture implicit temporal relationships across facts and relations, by providing temporal differences as explicit features. Our contributions are summarized as:

- We propose evaluation protocols for link and time interval prediction queries for TKBC. For link prediction, we highlight that existing evaluations seriously over/under-estimate system

[*] Equal contribution

performance, and offer a time-aware filtering method for more reliable evaluation. For time interval prediction, we propose an evaluation metric that rewards a model for predicting an interval with partial overlap with gold interval, as well as for nearness to gold in case of no overlap.
- We present TIMEPLEX, a TKBC model that factorizes a temporal KB using entity, relation and time embeddings. It can learn and exploit soft ordering and span constraints between potentially all relation pairs (including that of a relation with itself). It beats recent and competitive models on several recent standard TKBC data sets.

We will release an open-source implementation[1] of all models and experiments discussed here.

2 Preliminaries and Prior Work

2.1 Time-Agnostic KBC

Time-agnostic KBC has been intensely researched (Bordes et al., 2013; Yang et al., 2015; Nickel et al., 2016; Jain et al., 2018a; Lacroix et al., 2018; Jain et al., 2018b). A common approach is to score an (s, r, o) triple as a function over jointly learned entity and relation embeddings. The models are trained using loss functions imposing - scores for known triples should be higher than (randomly sampled) negative triples.

Our work is based on ComplEx (Trouillon et al., 2016), abbreviated as CX. It embeds s, r, o to vectors of complex space $\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o} \in \mathbb{C}^D$. CX defines the score ϕ of a fact (s, r, o) as $\mathrm{Re}(\langle \boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o}^\star \rangle)$ where

$$\langle \boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o}^\star \rangle = \sum_{d=1}^{D} s[d]\ r[d]\ o^\star[d] \qquad (1)$$

is a 3-way inner product, $\boldsymbol{o}^\star$ is the complex conjugate of $\boldsymbol{o}$, and $\mathrm{Re}(c)$ is real part of $c \in \mathbb{C}$. If real embeddings are used instead, the above formula reduces to DistMult (Yang et al., 2015). We choose CX as our base model, because it is competitive with recent KBC models (Ruffinelli et al., 2020).

2.2 Temporal KBC Problem Setup

A temporal KB associates the validity of a triple (s, r, o) with one or more time intervals $T \subseteq \mathfrak{T}$, where $\mathfrak{T}$ is the domain of "all time". Each interval T is represented as $[t_b, t_e]$, with begin and end time instants. Some event-style facts (e.g., born in) may have $t_b = t_e$. For simplicity, we assume that $\mathfrak{T}$ is discretized to a suitable granularity and is represented by a set of integers. Temporal KB facts have the form (s, r, o, T), and are partitioned into train, dev and eval (test) folds, abbreviated as tr, de, ev. System predictions are abbreviated as pr.

Given the train and dev folds, our goal is to learn a model that scores any unseen fact. A system is evaluated via link prediction queries $(?, r, o, T)$ and $(s, r, ?, T)$, and time interval prediction queries $(s, r, o, ?)$. In our setting, KB incompleteness exists at all times — the eval fold may include instances from any interval in time, arbitrarily overlapping train and dev fold instances.[2]

2.3 Recent TKBC Systems

Recent work adopts a common style for extending $\phi(s, r, o)$ to temporal score $\phi(s, r, o, t)$. Lacroix et al. (2020) embed each time instant t to vector $\boldsymbol{t}$ and use the form $\langle \boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o}^\star, \boldsymbol{t} \rangle$ (called TNT-ComplEx). This can be interpreted as any *one* of $\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o}^\star$ becoming $\boldsymbol{t}$-dependent. Goel et al. (2020) make *both* subject and object embeddings time-dependent; the 'diachronic' embedding $\boldsymbol{e} \in \mathbb{R}^D$ of entity e is characterized by $\boldsymbol{e}_t[d] = a_e[d] \sin(w_e[d]\, t + b_e[d])$, where $d \in D$ and the sinusoidal nonlinearity affords the capacity to switch "entity features" on and off with time t. HyTE (Dasgupta et al., 2018) models $\boldsymbol{t} \in \mathbb{R}^D, \|\boldsymbol{t}\|_2 = 1$ and project *all* of $\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o}$ on to $\boldsymbol{t}$: $\boldsymbol{x} \downarrow \boldsymbol{t} = \boldsymbol{x} - (\boldsymbol{x} \cdot \boldsymbol{t})\boldsymbol{t}$, where $\boldsymbol{x} \in \{\boldsymbol{s}, \boldsymbol{r}, \boldsymbol{o}\}$. In all cases, time-dependent entity embeddings are plugged into standard scoring functions like DistMult, CX, or SimplE (Kazemi and Poole, 2018). A very different approach (García-Durán et al., 2018) encodes the string representation of relation and time with an LSTM, which is used in TransE (TA-TransE) or DistMult (TA-DM).

These formulations do not directly model recurrences of a relation or interactions (e.g., mutual exclusion) between relations. There is some prior work on explicitly providing ordering constraints between relations (e.g., born, married, died) (Jiang et al., 2016). In contrast, TIMEPLEX assumes no such additional engineered inputs; it has explicit components to enable learning of temporal (soft) constraints, as model weights, jointly with embeddings of entities, relations, and time instants. Such constraint based reasoning has also been exploited (in a limited way) for a different task, namely, temporal question answering (Jia et al., 2018).

[1] github.com/dair-iitd/tkbi

[2] A different TKBC task studies only future fact predictions (Trivedi et al., 2017; Jin et al., 2019).

2.4 Standard Evaluation Schemes

Link Prediction: Link prediction queries in KBC are of the form $(s, r, ?)$ with a gold response o^{ev}. Similarly, for TKBC they are of the form $(s, r, ?, T)$. The cases of $(?, r, o)$ and $(?, r, o, T)$ are symmetric and receive analogous treatment. Link prediction performance is evaluated by finding the rank of o^{ev} in the list of all entities ordered by decreasing score ϕ assigned by the model, and computing MRR. Other measures include the fraction of queries where o^{ev} is recalled within the top 1 or top 10 ranked predictions (HITS@1 and HITS@10).

A query may have multiple correct answers. A model must not be penalized for ranking a different *correct* entity over o^{ev}. In KBC this is achieved by filtering out all correct entities above o^{ev} in ranked list before computing the metrics. In TKBC, filtering requires additional care, as depicted in Table 1. We develop time-aware filtering in Section 3.2.

Time Prediction: Time prediction queries of the form $(s, r, o, ?)$ will require comparing a gold time interval $T^{\text{ev}} = [t_b^{\text{ev}}, t_e^{\text{ev}}]$ with a predicted interval $T^{\text{pr}} = [t_b^{\text{pr}}, t_e^{\text{pr}}]$. Since this is an understudied task, evaluation metrics have not yet been standardized. One might adapt the TAC metric popular in Temporal Slot Filling (Ji et al., 2011; Surdeanu, 2013). Adapted to TKBC, TAC[3] will compute a score as $\frac{1}{2}\left[\frac{1}{1+|t_b^{\text{ev}}-t_b^{\text{pr}}|} + \frac{1}{1+|t_e^{\text{ev}}-t_e^{\text{pr}}|}\right]$. Unfortunately, TAC score is not entirely satisfactory for this task. For instance, TAC will assign the same merit score when gold interval [10,20] is compared with predicted interval [5,15], versus when gold [100,200] is compared with prediction [95,195]. However, a human would judge the latter more favorably, because a 5-minute delay in a 10-minute trip would usually be considered more serious than in a 100-minute journey. In response, we investigate alternative evaluation metrics inspired by bounding box evaluation protocols from Computer Vision, in Section 3.1.

3 Evaluation Metrics and Filtering

The preceding discussion motivates why we need clearly-thought-out filtering and evaluation schemes, not only for time interval prediction queries, but also because time affects link prediction evaluation in subtle but fundamental ways. This section addresses both issues.

3.1 Time Interval Prediction

One possible way to evaluate time prediction is to adapt measures to compare bounding boxes in computer vision, e.g., Intersection Over Union (IOU): $\text{IOU}(T^{\text{ev}}, T^{\text{pr}}) = \frac{\text{vol}(T^{\text{ev}} \cap T^{\text{pr}})}{\text{vol}(T^{\text{ev}} \cup T^{\text{pr}})} \in [0, 1]$, where vol for our case simply refers to the size of the interval. Unfortunately, IOU loses discrimination once $T^{\text{ev}} \cap T^{\text{pr}} = \varnothing$; e.g., $\text{IOU}([1,2],[3,4]) = \text{IOU}([1,2],[30,40]) = 0$. This has been noticed recently in computer vision also, and a metric called gIOU been introduced (Rezatofighi et al., 2019):

$$\text{gIOU}(T^{\text{ev}}, T^{\text{pr}}) = \text{IOU}(T^{\text{ev}}, T^{\text{pr}}) - \frac{\text{vol}((T^{\text{ev}} \Cup T^{\text{pr}}) \setminus (T^{\text{ev}} \cup T^{\text{pr}}))}{\text{vol}(T^{\text{ev}} \Cup T^{\text{pr}})} \in (-1, 1]. \quad (2)$$

$T^{\text{ev}} \Cup T^{\text{pr}}$ is the smallest single contiguous interval (**hull**) containing all of T^{ev} and T^{pr}. E.g., $[1,2] \Cup [30,40] = [1,40]$.

gIOU can be negative, which is not ideal for a performance metric that is aggregated over instances. A simple fix (gIOU′) is to scale it to [0,1] via $(\text{gIOU} + 1)/2$, but we notice that the tiniest overlap between T^{ev} and T^{pr} yields gIOU′ to be at least half, regardless of $\text{vol}(T^{\text{ev}})$ or $\text{vol}(T^{\text{pr}})$. In response, we propose a novel *affinity enhanced IOU*:

$$\text{aeIOU}(T^{\text{ev}}, T^{\text{pr}}) = \frac{\max\{1, \text{vol}(T^{\text{ev}} \cap T^{\text{pr}})\}}{\text{vol}(T^{\text{ev}} \Cup T^{\text{pr}})} \quad (3)$$

When $T^{\text{ev}} \cap T^{\text{pr}} = \varnothing$, the denominator includes "wasted time", reducing aeIOU. The '1' in the numerator represents the smallest granularity of time in the data (see Section 2.2).

Comparison of Evaluation Metrics: A good time interval prediction metric (M) must satisfy the property (P) that: if two predicted intervals have intersections of the same size (possibly zero) with the gold interval, then the prediction that has a smaller hull with the gold interval should be scored higher by M. Formally, let T^{pr_1} and T^{pr_2} be two predictions made for T^{ev}.

Property P: Let $\text{vol}(T^{\text{ev}} \cap T^{\text{pr}_1}) = \text{vol}(T^{\text{ev}} \cap T^{\text{pr}_2})$. Then, $M(T^{\text{ev}}, T^{\text{pr}_1}) > M(T^{\text{ev}}, T^{\text{pr}_2})$ if and only if $\text{vol}(T^{\text{ev}} \Cup T^{\text{pr}_1}) < \text{vol}(T^{\text{ev}} \Cup T^{\text{pr}_2})$.

Theorem: IOU and gIOU′ do not satisfy property P, whereas aeIOU satisfies it.

The proof for the theorem is in Appendix B. This suggests that aeIOU is a more defensible metric for our task, compared to other alternatives.

[3] TAC's original score compares gold and predicted *bounds* on begin and end of an interval. This formula is its adaptation, where begin and end are each a specific time point.

Eval query: (s = French National Assembly, r = has member, o =?, T^{ev} = $[2000, 2003]$)							
Candidates o, system ordered	Known duration of o (any fold)	**Method 1** Unfiltered	**Method 2** Time-insensitive	**Method 3** Time-sensitive			
				2000	2001	2002	2003
Pierre	$[2002, 2003]$	1	0	1	1	0	0
Paul	$[2003, 2008]$	1	0	1	1	1	0
Alain	$[2008, 2009]$	1	0	1	1	1	1
Claude	$[2000, 2003]$	1	0	0	0	0	0
Jean	-	-	-	-	-	-	-
Time-sensitive rank of *Jean*		1+4=5	1+0=1	1+3=4	1+3=4	1+2=3	1+1=2

Table 1: *Jean* is the gold answer (o^{ev}). Rows are ranked system predictions, which may be seen with same s and r for different intervals (Column 2). Columns 3–4 show the filtering of existing methods (1:unfiltered, 0:filtered). Columns 5–8 (Method 3, our proposal) show the filtering for each time instant. The bottom row shows ranks of *Jean* as computed by different methods. Existing methods over- or under-estimate performance. Method 3 assigns *Jean* a rank of 3.25, which is the average of the filtered ranks $\{4, 4, 3, 2\}$ for each time instant in $[2000, 2003]$.

3.2 Link Prediction

We first illustrate the unique challenges offered by TKBC link prediction queries through an example in Table 1. The query asks for the name of a person who was a member of the French National Assembly in interval $[2000, 2003]$. Let the gold answer (object) o^{ev} be Jean, which is ranked at the fifth position by the model. All four entities above Jean are seen with the same subject and relation in the data, but for different time intervals. E.g., Pierre is also a member of the assembly, but during $[2002, 2003]$. The key question is: how should the four entities above Jean be filtered to compute its final rank?

We argue (Table 1) that existing filtering approaches are unsatisfactory. Dasgupta et al. (2018) underrate model performance by not performing any filtering (Method 1). In this example, the model is penalized for Claude, even though the time-interval for Claude exactly matches the query. On the other hand, García-Durán et al. (2018) and Jin et al. (2019) ignore time information altogether and filter out *all* entities seen with gold (s, r). This can greatly overestimate system quality (Method 2). For instance, the model is not penalized for predicting Alain, even though its membership interval has no overlap with the query interval.

Ideally, filtering must account for the overlap between the query time interval and the time intervals associated with system-proposed entities. We propose such a filtering strategy (Method 3). We split the query interval into time instants, and compute a filtered rank for each time point independently. Entities that have full time overlap (or no overlap) will always (respectively, never) get filtered for a time instant. Partially overlapping entities will get filtered in only overlapping instants (e.g., 2 out of 4 for Pierre). After computing filtered ranks for each time instant, we output the final rank as an average of all such filtered ranks. In this example, this approach will compute the average of $\{4, 4, 3, 2\}$, which is 3.25. This average rank is used when computing standard metrics like MRR and HITS@10.

Note that the run-time complexity of the proposed evaluation protocol is linear in the size of interval, because we compute a filtered rank for each time point separately.

4 The Proposed TIMEPLEX Framework

Similar to TNT-Complex, TIMEPLEX learns complex-valued entity, relation, and time instant embeddings. However, it has several differences from TNT-Complex. (1) Its base scoring function $\phi^{TX}(s, r, o, t)$ adds several products of three embeddings, instead of a single four-way product (Section 4.1). (2) It has a fully automatic mechanism to introduce additional features to capture recurrent nature of a relation, as well as temporal interactions between pairs of relations (Section 4.2). (3) It uses a two-phase training (Section 4.3) curriculum that estimates first the embeddings and then novel additional parameters. (4) Its testing protocol can output a missing time-interval T for time-interval prediction queries (Section 4.4).

4.1 TIMEPLEX Base Model

Just as a joint distribution is often approximated using lower-order marginals in graphical models (Koller and Friedman, 2009), TIMEPLEX constructs a base score (ϕ^{TX}) by augmenting CX score with three time-dependent terms:

$$\begin{aligned}\phi^{TX}(s, r, o, t) &= \langle \boldsymbol{s}, \boldsymbol{r}^{\text{SO}}, \boldsymbol{o}^{\star} \rangle + \alpha \langle \boldsymbol{s}, \boldsymbol{r}^{\text{ST}}, \boldsymbol{t}^{\star} \rangle \\ &\quad + \beta \langle \boldsymbol{o}, \boldsymbol{r}^{\text{OT}}, \boldsymbol{t}^{\star} \rangle + \gamma \langle \boldsymbol{s}, \boldsymbol{o}, \boldsymbol{t}^{\star} \rangle. \quad (4)\end{aligned}$$

Here, $\boldsymbol{s}, \boldsymbol{o}, \boldsymbol{t} \in \mathbb{C}^D$, whereas each r is represented as a collection of three such vectors $(\boldsymbol{r}^{\text{SO}}, \boldsymbol{r}^{\text{ST}}, \boldsymbol{r}^{\text{OT}})$, and hence requires three times the parameters. $\boldsymbol{r}^{\text{ST}}$ represents a relation which is true for entity s at time t (similarly for $\boldsymbol{r}^{\text{SO}}$ and $\boldsymbol{r}^{\text{OT}}$). α, β and γ are hyperparameters.

Jiang et al. (2016) observed that several relations attach to a subject or object only at specific time points. E.g., subject Barack Obama was president in 2009, regardless of the object United States. In such cases, the formulation above is fully expressive. To extend from single time instants t to an interval T, we propose

$$\phi^{TX}(s, r, o, T) = \textstyle\sum_{t \in T} \phi^{TX}(s, r, o, t). \quad (5)$$

4.2 Relation Recurrence and Pair Scores

We extend TIMEPLEX's base model via additional (soft) temporal constraints that can help in better assessing the validity of a tuple. We aim to capture three types of temporal constraints:

Relation Recurrence: Many relations do not recur for a given entity (e.g., a person is born only once). Some relations recur with fixed periodicity (e.g., Olympic games recur every four years). Recurrences of other relations may be distributed around a mean time period.

Ordering Between Relations: A relation precedes another, for a given entity. E.g., *personBornYear* should precede *personDiedYear* for a given subject entity (person).

Time Gaps Between Relations: The difference in time instants of two relations (wrt to an entity) is distributed around a mean, e.g., *personDiedYear* minus *personBornYear* has a mean of about 70 with some observed variance.

The first constraint concerns a single relation, whereas the latter two concern pairs of relations. Jiang et al. (2016) attempted to capture relation ordering constraints as model regularization, but their approach does not take into account time differences. Nor does it model relation recurrence.

Basic TIMEPLEX may not be able to learn these constraints from data either, since each time instant is modeled as a separate embedding with *independent* parameters — it has no explicit understanding of the difference between two time instants. In response, we augment TIMEPLEX with additional features that capture how soon an event recurs, or how soon after the occurrence of one relation, another relation is likely to follow. We define two scoring functions ϕ^{Rec} and ϕ^{Pair} for these two cases, to be aggregated with ϕ^{TX} (eqn. 4).

Inspired by García-Durán and Niepert (2018), we model time gaps as drawn from Gaussian distributions. We use $\mathcal{N}(x|\mu, \sigma)$ to denote the probability density of a Gaussian distribution with mean μ and std deviation σ at the time (difference) value x (See Figure 1 (a)). We denote as KB^{tr} all tuples in the train fold.

Recurrence Score: We say that (s, r, o) *recurs* if there are at least two distinct intervals T such that $(s, r, o, T) \in \text{KB}^{\text{tr}}$. If there are at least K^{Rec} distinct pairs (s, o) such that (s, r, o) recurs, then r is considered *recurrent*. K^{Rec} is a hyperparameter.

For each recurrent relation r, our model learns three new parameters: μ_r, σ_r, and b_r. Intuitively, $\mathcal{N}(\cdot|\mu_r, \sigma_r)$ represents a distribution of typical durations between two recurring instances of a relation (with a specific subject and object entity) and b_r is the bias term. For non-recurrent relations, only the bias b_r is learnt. While computing recurrence features, all training tuples of the form (s, r, o, T) are reduced to (s, r, o, t), i.e., with a singleton time interval, where $t = t_b$, the start time of T. TIMEPLEX sets a fact recurrence score, ϕ^{Rec}, as follows:

1. If $(s, r, o, \star) \notin \text{KB}^{\text{tr}}$, set $\phi^{\text{Rec}} = 0$.
2. Else, if r is not recurrent, set $\phi^{\text{Rec}} = b_r$. This allows the model to learn to penalize repetition of relations that do not recur.
3. Find time gap (δ) to its closest recurrence:
$$\delta = \min_{\{(s,r,o,t') \in \text{KB}^{\text{tr}}: t' \neq t\}} |t - t'|. \quad (6)$$
Then, set
$$\phi^{\text{Rec}}(s, r, o, T = [t_b, t_e]) = \phi^{\text{Rec}}(s, r, o, t_b) = w_r \mathcal{N}\big(\delta|\mu_r, \sigma_r\big) + b_r. \quad (7)$$

The intuition is that ϕ^{Rec} should penalize the proposed (s, r, o, T) if δ is not close to the mean gap μ_r. For example, (Presidential election, held in, USA, 2017) should be penalized, if (Presidential election, held in, USA, 2016) is known, and the event reoccurs every 4 years ($\mu_r = 4, \sigma_r \approx 0$).

Relation Pairs Score: TIMEPLEX also learns soft time constraints between pairs of relations. We describe this mechanism for subjects; objects are handled analogously. For each relation pair (r, r'), we maintain four parameters, $\mu_{rr'}$, $\sigma_{rr'}$, $b_{rr'}$ and $w_{rr'}$, whose purpose we will describe presently. As with recurrence scores, all training tuples (s, r, o, T) are reduced to (s, r, o, t), where $t = t_b$, the start time of T. Given the candidate tuple

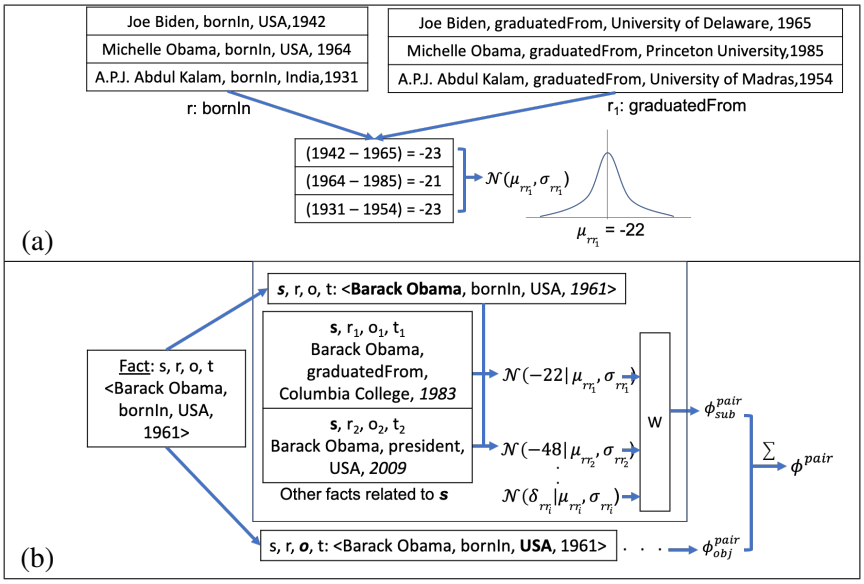


Figure 1: (a) *Pre-training Data Statistics Collection Strategy* for relation pair (bornIn, graduatedFrom). Such statistics are computed for all relation pairs, and (b) *Relation Pair Score Computation* of a fact using the statistics collected in part (a). Here, $\delta_{rr_i} = (t - t_i)$.

(s, r, o, t) to score, we collect fact tuples

$$\{f_i = (s, r_i, o_i, t_i) \in \text{KB}^{\text{tr}}, r_i \neq r\}, \quad (8)$$

$sc(f_i) = \mathcal{N}(t - t_i|\mu_{rr_i}, \sigma_{rr_i}) + b_{rr_i}$ having the same subject but a different relation, into the set called $\text{KB}^{\text{Pair}}(s)$. The i^{th} tuple in $\text{KB}^{\text{Pair}}(s)$ is scored as $sc(f_i) = \mathcal{N}(t - t_i|\mu_{rr_i}, \sigma_{rr_i}) + b_{rr_i}$. This represents the contribution of f_i in the validity of candidate tuple, based on their (signed) time difference, and typical time differences observed between these two relations. $\phi^{\text{Pair}}_{\text{sub}}$ needs to aggregate these over f_i. The (trained) parameter $w_{rr'}$ measures how much the times associated with r' influence our belief in (s, r, o, t). Using these, we define the weighted average

$$\phi^{\text{Pair}}_{\text{sub}}(s, r, o, t) = \sum_{f_i \in \text{KB}^{\text{Pair}}(s)} sc(f_i) \frac{\exp(w_{rr_i})}{\sum_{f_j} \exp(w_{rr_j})}.$$

A similar $\phi^{\text{Pair}}_{\text{obj}}$ score is computed for the object entity, and overall $\phi^{\text{Pair}} = \phi^{\text{Pair}}_{\text{sub}} + \phi^{\text{Pair}}_{\text{obj}}$ (See Figure 1 (b)). The **final scoring function** of TIMEPLEX is

$$\phi(s, r, o, T) = \phi^{TX}(s, r, o, T) + \kappa\phi^{\text{Pair}}(s, r, o, T) + \lambda\phi^{\text{Rec}}(s, r, o, T), \quad (9)$$

where κ and λ are model hyperparameters.

4.3 Training

We train TIMEPLEX in a curriculum of two phases. In the first phase, we optimize embeddings for all entities, relations and time-instants by minimizing the log-likelihood loss using only the base model TX. We compute the probability of predicting a response o for a query $(s, r, ?, T)$ as:

$$\Pr(o|s, r, T) = \frac{\exp(\phi^{TX}(s, r, o, T))}{\sum_{o'} \exp(\phi^{TX}(s, r, o', T))} \quad (10)$$

We can similarly compute $\Pr(s|r, o, T)$ and similar terms for time instant queries, e.g., $\Pr(o|s, r, t)$ and $\Pr(t|s, r, o)$. We then convert every $(s, r, o, T = [t_b, t_e]) \in \text{KB}^{\text{tr}}$ in time-instant format by enumerating all (s, r, o, t), for $t \in [t_b, t_e]$. Training of embeddings minimizes the **log-likelihood loss**:

$$-\sum_{\langle s,r,o,t\rangle \in KB^{\text{tr}}} \Big(\log \Pr(o|s, r, t; \theta) + \log \Pr(s|o, r, t; \theta) + \log \Pr(t|s, r, o; \theta)\Big) \quad (11)$$

In the second phase, we freeze all embeddings and train the parameters of the recurrence and pairs models. Here, too, we use the log-likelihood loss, except that ϕ^{TX} is replaced by the overall ϕ function. Parameters $\mu_{rr'}$ and $\sigma_{rr'}$ of the relation-pairs model component are not trained via backpropagation. Instead, they are fitted separately, using the difference distributions for the pair of relations in the training KB. This improves the overall stability of training.

4.4 Inference

At test time, for a link prediction query, TIMEPLEX ranks all entities in decreasing order of $\Pr(o|s, r, T)$ or $\Pr(s|r, o, T)$ scores. For time prediction, its goal is to output a predicted time duration T^{pr}. We first compute a probability

distribution over time instants $\Pr(t|s,r,o) = \frac{\exp(\phi(s,r,o,t))}{\sum_{t' \in \mathfrak{T}} \exp(\phi(s,r,o,t'))}$. We then greedily coalesce time instants to output the best duration. For greedy coalescing, we tune a threshold parameter θ_r for each relation r using the dev fold (such that shorter θ_r prefers short duration and vice versa). We then initialize the predicted interval T^{pr} as $\text{argmax}_t \Pr(t|s,r,o)$. Then, as long as total probability of the interval, i.e., $\sum_{t \in T^{\text{pr}}} \Pr(t|s,r,o)$ is less than θ_r, we extend T^{pr} with the instant to its left or right, whichever has a higher probability.

5 Experiments

We investigate the following research questions. (1) Does TIMEPLEX convincingly outperform the best time-agnostic and time-aware KBC systems on link prediction and time interval prediction tasks? (2) Are recurrent and pairwise features helpful in the final performance? (3) Are TIMEPLEX's time embeddings meaningful, i.e., do they capture the passage of time in an interpretable manner? (4) Do TIMEPLEX predictions honor temporal constraints between relations?

5.1 Datasets & Experimental Setup

Datasets: We report on experiments with four standard TKBC datasets. WIKIDATA12k and YAGO11k (Dasgupta et al., 2018) are two knowledge graphs with a time interval associated with each triple. These contain relational facts like (David Beckham, plays for, Manchester United; [1992, 2003]). ICEWS14 and ICEWS05-15 (García-Durán et al., 2018) are two event-based temporal knowledge graphs, with facts from Integrated Crisis Early Warning System repository. These primarily include political events with timestamps (no nontrivial intervals). We consider the time granularity for interval datasets as 1 year, and for ICEWS datasets as 1 day. See Table 5 in Appendix A for salient statistics of these datasets. By experimenting across the spectrum, from 'point' events to facts with duration, we wish to ensure the robustness of our observations.

García-Durán et al. (2018) also report performance on the Yago15k dataset. However, for this dataset, only 17% of the facts have associated temporal information. In contrast, all the datasets we used had at least 99% of facts with temporal information. Hence, we believe a temporal model will not substantially improve the performance of a time-agnostic model on this dataset. Note that TNT-Complex (Lacroix et al., 2020) also obtained only a slight improvement over a time-agnostic model on Yago15k, supporting our hypothesis. A contemporaneous work by (Ahrabian et al., 2020) proposed new multi-relational temporal Knowledge Graph based on the daily interactions between artifacts in GitHub. We leave exploration of this dataset for future work.

Algorithms compared: We compare against our reimplementations of CX, HyTE, TA-family, and TNT-Complex. In all cases we verify that our implementations give comparable or better scores as reported in literature. We combine HyTE and TA, with scoring functions from TransE, DistMult and CX and present the best results. We also compare against reported results in DE-SimplE.

Experimental Details: For all models, we optimize parameters with AdaGrad running for 500 epochs for all losses, with early stopping on dev fold. We control for an approximately comparable number of parameters and set dimensionality of 200 for all complex embeddings and 400 for all real embeddings. We follow other best practices in the literature, such as L2 regularization only on embeddings used in the current batch (Trouillon et al., 2016), adding inverted facts (o, r^{-1}, s, T) , using 1vsAll negative sampling (Dettmers et al., 2018) whenever applicable, and using temporal smoothing for ICEWS datasets (Lacroix et al., 2020).

Some instances in interval datasets have t_b or t_e missing. Following Dasgupta et al. (2018), we replace missing values by $-\infty$ or $+\infty$, respectively. For time prediction queries, we remove such instances from test sets. For ICEWS datasets we set $t_b = t_e$. For time interval prediction, all models use our greedy coalescing inference from Section 4.4.

For TIMEPLEX, we perform a grid search for all hyperparameters, and pick the best values based on MRR scores on valiations set. Hyperparameters for all datasets are described in Appendix G.

5.2 Results and Observations

Table 2 compares all algorithms for link prediction. We find that the best performing baseline among existing TKBC systems is the recently proposed TNT-Complex model. TIMEPLEX outperforms TNT-Complex by over 3 MRR points in ICEWS datasets. Its gains (3.25 and 5.6 pts) are even more pronounced in interval datasets. All differences are statistically significant using paired t-test with

Dataset→	WIKIDATA12k			YAGO11k			ICEWS05-15			ICEWS14		
↓Methods	MRR	HITS@1	HITS@10	MRR	HITS@1	HITS@10	MRR	HITS@1	HITS@10	MRR	HITS@1	HITS@10
CX	24.82	14.30	48.90	18.14	11.46	31.11	48.68	37.00	72.63	45.50	33.87	69.73
TA (CX)	22.78	12.69	46.00	15.24	9.36	26.26	49.23	37.6	72.69	40.97	29.58	63.87
HyTE (TransE)	25.28	14.70	48.26	13.55	3.32	29.81	23.73	3.11	62.76	24.91	2.98	65.30
DE-SimplE	25.29	14.68	49.05	15.12	8.75	26.74	51.30	39.20	74.80	52.60	41.80	72.50
TNT-Complex	30.10	19.73	50.69	18.01	11.02	31.28	60.58	51.14	78.50	56.72	47.04	75.40
TIMEPLEX (base)	32.38	22.03	52.79	18.35	10.99	31.86	63.91	**54.62**	81.42	60.25	51.29	77.05
TIMEPLEX	**33.35**	**22.78**	**53.20**	**23.64**	**16.92**	**36.71**	**63.99**	54.51	**81.81**	**60.40**	**51.50**	**77.11**

Table 2: Link prediction performance across four datasets. The last row reports results for TIMEPLEX(base) augmented with pair/recurrent features.

Datasets→	YAGO11k	WIKIDATA12k
↓Methods	aeIOU	aeIOU
HyTE	5.41	5.41
TNT-Complex	8.40	23.35
TIMEPLEX (base)	14.21	26.20
TIMEPLEX	**20.03**	**26.36**

Table 3: Time prediction performance.

$p < 0.01$. These scores establish a new state of the art for link prediction on all four datasets.

A contemporaneous work, ATiSE (Nayyeri et al., 2020) models KB entities and relations using time dependent Gaussian embedding, but show weaker performance (see Table 2 and Table 11).

We are the first to look at the task of predicting *time intervals*, and we report performance using our novel aeIOU metric (Table 3). We see that TIMEPLEX outperforms TNT-Complex on both datasets, with a huge 11+ pt jump on the Yago11K dataset. It is also noteworthy that even the base model of TIMEPLEX is consistently better than TNT-Complex across all experiments.

On Pair/recurrent features: We find that recurrent features are very helpful in both interval datasets, and significantly improve link prediction performance. Relation pair features particularly help in YAGO11k — over 5 pt aeIOU boost in time prediction, but on WIKIDATA12k they make only a marginal difference. On inspecting the datasets, we find that 78% of entities in WIKIDATA12k are seen with a single, recurring relation (such as *award received*, or *member of sports team*); therefore, relation pair features cannot help.

ICEWS datasets are scraped from news events. On inspecting the datasets, we find that the events do not follow any temporal ordering and are fairly non-regular in event recurrence as well. Hence, TIMEPLEX's improvements over the base model are limited. We further investigate the differing performance on datasets and the value of pair features in the next section.

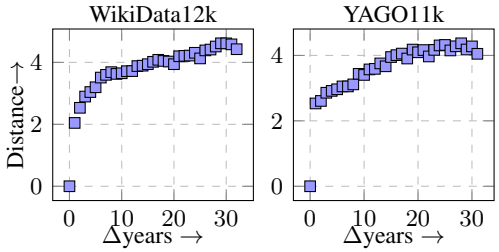


Figure 2: L_2 distances (y-axis) between TIMEPLEX time embeddings increase with time gap (x-axis).

5.3 Diagnostics

Time gap vs. embedding distances: Longevity of relations, or gaps between events, are often determined by physical phenomena that are smooth and continuous in nature. Therefore, we expect the embedding of the year 1904 to be closer to that of 1905 compared to the embedding of, say, 1950.

To validate this hypothesis, we compute mean L2 distance between embeddings of time instants which are apart by a given time gap. To filter noise, we drop instant pairs with extreme gaps that have low support (less than 30). For WIKIDATA12k we used embeddings of years $[1984, 2020]$ and for YAGO11k we use embeddings of years $[1958, 2017]$.

Figure 2 shows that L_2 distance between pairs of time embeddings increases with the actual year gap between them. Since we enumerate all time points in the given fact time-interval, years that are closer share a lot of facts (triples), and are hence closer in the embedding space. This has a smoothing effect on time embeddings. Hence they correlate well with actual time-gaps. This strongly suggests that the time embeddings learnt by TIMEPLEX naturally represent physical time.

Temporal ordering of relation pairs: Both YAGO11k and WIKIDATA12k contain relations with temporal dependencies, e.g., *bornInPlace* should always precede *diedInPlace* for the same

	YAGO11k	WIKIDATA12k
CX	10.04	0.7
HyTE	7.2	0.4
TNT-Complex	8.82	0.3
TIMEPLEX (Base)	6.6	0.3
TIMEPLEX	**1.9**	**0.2**

Table 4: Ordering constraint violations among top predictions of various models (% of facts in test set).

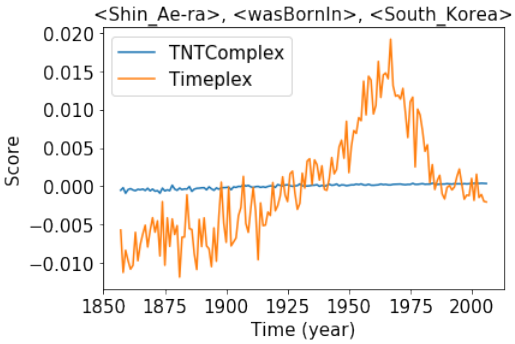


Figure 3: Time prediction comparison for two systems.

person. We now study whether TIMEPLEX models are able to learn these natural constraints from data.

We first exhaustively extract all relation pairs (r_1, r_2), where the existence of both $(s, r_1, \star, t_1)$ and $(s, r_2, \star, t_2)$ is accompanied by $t_1 < t_2$ at least 99% of the time, with a minimum support of 100 entities s.[4] We now verify whether TIMEPLEX honors r_1 before r_2 when making predictions.

For each query $(?, r, o, t)$ in the test set, we check whether the top model prediction violates any known temporal ordering constraint in this list. For example, for a query *(?, **hasWonPrize**, Nobel Prize, **1925**)*, if the model predicted *Barack Obama* and the KB already had Barack Obama born in Hawaii in 1961, then this will be considered as an ordering violation. Table 4 reports the number such violations as fraction of test set size. TIMEPLEX significantly reduces such errors for YAGO11k; this is also reflected in its superior time prediction performance. For WIKIDATA12k, the errors for TIMEPLEX (base) are already low, hence pair features are not found to be particularly helpful.

As an illustrative example, we consider the time prediction query *(Shinae-ra, wasBornIn, South Korea, ?)*, with the gold answer 1969. The only other fact seen for *Shinae-ra* in the train KB is *(Shinae-ra, isMarriedTo, ChaIn-Pyo, (1995, -))*. TIMEPLEX predicts 1967 for this query (earning an aeIOU credit of 33.33). However, TNTComplex predicts 2013 (earning almost no credit) – this also highlights that it does not capture commonsense that a person can marry only after they are born.

To understand further, we plot the normalized scores for this query in time range $[1850, 2010]$ in Figure 3. The peak around 1967 for the TIMEPLEX plot can be attributed to the fact that mean difference for *isMarriedTo* and *wasBornIn* relations is around 30 in the dataset. Standard tensor factorization models like TNT-Complex are unable to exploit this, but our Pair features provide a way to the model to make very reasonable predictions. Other similar plots can be found in the Appendix.

6 Discussion

TIMEPLEX cannot exploit the influence that an entity can have on time difference distributions. For example, the life expectancy of a person (mean difference between *diedIn* and *bornIn* events) would be around 85 in Japan, but 54 in Lesotho. Extending our model to learn separate parameters for each ⟨rel, entity⟩ pair may be difficult due to sparsity. Also, recurrent facts may admit exceptions: Winter Olympics are held every 4 years except for 1992 and 1994. However, we do not expect even humans to do well in such cases. Exceptions like these are sparse and difficult to learn, except by rote.

7 Conclusion

We presented TIMEPLEX, a new TKBC framework, which combines representations of time with representations of entities and relations. It also learns soft temporal consistency constraints, which allow knowledge of one temporal fact to influence belief in another fact. TIMEPLEX exceeds the performance of existing TKBC systems. Time embeddings are temporally meaningful, and TIMEPLEX makes fewer temporal consistency and ordering mistakes. We also argue that current evaluation schemes for both link and time prediction have limitations, and propose more meaningful schemes.

Acknowledgements

This work is partly supported by IBM AI Horizons Network grants. IIT Delhi authors are supported by an IBM SUR award, grants by Google, Bloomberg and 1MG, Jai Gupta Chair professorship and a Visvesvaraya faculty award by the Govt. of India. The fourth author is supported by a Jagadish Bose Fellowship. We thank IIT Delhi HPC facility for compute resources. We thank Sankalan, Vaibhav and Siddhant for their helpful comments on an early draft of the paper.

[4] The list of such relation pairs is given in the Appendix C

References

Kian Ahrabian, Daniel Tarlow, Hehuimin Cheng, and Jin L. C. Guo. 2020. Software engineering event modeling using relative time in temporal knowledge graphs. *CoRR*, abs/2007.01231.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS Conference*, pages 2787–2795.

Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha P. Talukdar. 2018. HyTE: Hyperplane-based temporally aware knowledge graph embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2001–2011. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

Alberto García-Durán, Sebastijan Dumancic, and Mathias Niepert. 2018. Learning sequence encoders for temporal knowledge graph completion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4816–4821. Association for Computational Linguistics.

Alberto García-Durán and Mathias Niepert. 2018. Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 372–381. AUAI Press.

Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. Diachronic embedding for temporal knowledge graph completion. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3988–3995. AAAI Press.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia: Extended abstract. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 3161–3165. IJCAI/AAAI.

Prachi Jain, Pankaj Kumar, Mausam, and Soumen Chakrabarti. 2018a. Type-sensitive knowledge base inference without explicit type supervision. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 75–80. Association for Computational Linguistics.

Prachi Jain, Shikhar Murty, Mausam, and Soumen Chakrabarti. 2018b. Mitigating the effect of out-of-vocabulary entity pairs in matrix factorization for knowledge base inference. In *IJCAI*.

Heng Ji, Ralph Grishman, Hoa Trang Dang, and Kira Griffitt. 2011. Overview of the TAC2011 knowledge base population (KBP) track. In *Text Analysis Conference (TAC)*, pages 14–15.

Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. TEQUILA: temporal question answering over knowledge bases. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 1807–1810. ACM.

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. 2016. Towards time-aware knowledge graph completion. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 1715–1724. ACL.

Woojeong Jin, Changlin Zhang, Pedro A. Szekely, and Xiang Ren. 2019. Recurrent event network for reasoning over temporal knowledge graphs. *CoRR*, abs/1904.05530.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 4289–4300.

Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor decompositions for temporal knowledge base completion. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2869–2878. PMLR.

Mojtaba Nayyeri, Fouad Alkhoury, Hamed Yazdi, and Jens Lehmann. 2020. Temporal knowledge graph completion based on time series gaussian embedding. *arXiv preprint arXiv:1911.07893v2*.

Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. 2016. Holographic embeddings of knowledge graphs. In *AAAI Conference*, pages 1955–1961.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 658–666. Computer Vision Foundation / IEEE.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. 2020. You CAN teach an old dog new tricks! on training knowledge graph embeddings. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mihai Surdeanu. 2013. Overview of the TAC2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *Text Analysis Conference (TAC)*.

Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3462–3471. PMLR.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *ICML*, pages 2071–2080.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Temporal Knowledge Base Completion: New Algorithms and Evaluation Protocols (Appendix)

A Dataset statistics

See Table 5 for some salient statistics of the datasets we used for experiments. Yago11k and Wikidata12k are interval based datasets. ICEWS14 and ICEWS05-15 are instant based datasets.

B Discussion on time evaluation metrics

We re-state the desired property P for a time evaluation metric-
Let $\text{vol}(T^{\text{ev}} \cap T^{\text{pr}_1}) = \text{vol}(T^{\text{ev}} \cap T^{\text{pr}_2})$ $M(T^{\text{ev}}, T^{\text{pr}_1}) > M(T^{\text{ev}}, T^{\text{pr}_2})$ if and only if $\text{vol}(T^{\text{ev}} \Cup T^{\text{pr}_1}) < \text{vol}(T^{\text{ev}} \Cup T^{\text{pr}_2})$.

aeIOU satisfies P:
For a fixed $\text{vol}(T^{\text{ev}} \cap T^{\text{pr}})$, we have $\text{aeIOU}(T^{\text{ev}}, T^{\text{pr}}) \propto 1/\text{vol}(T^{\text{ev}} \Cup T^{\text{pr}})$ (see Eqn 3). Hence, aeIOU satisfies property P.

IoU and gIOU do not satisfy P:
IoU: This metric gives 0 score to a model, if model's predicted interval does not intersect with the gold, irrespective of the hull. Hence IoU do not satisfy property P.
gIoU: Let us look at the following example. Suppose gold interval is [2002,2005], and consider 2 predictions- [1999,2001] and [1900,2001]. For both predictions, $\text{vol}((T^{\text{ev}} \Cup T^{\text{pr}}) \setminus (T^{\text{ev}} \cup T^{\text{pr}}))$ is zero, so the hull for the two predictions will be ignored (see Eqn 2), resulting in same scores for both predictions. Hence gIoU does not satisfy property P.

Model Performance with respect to various time evaluation metrics:
Table 6 reports the TAC, gIOU, and IOU scores of various temporal methods discussed in the paper.

C Temporal Constraints: Relation Ordering

Table 7 and 8 lists automatically extracted high confidence relation orderings seen in Yago11k and Wikidata12k datasets respectively. These orderings are used to guide TIMEPLEX at the time of training.

D Time prediction performance across relation classes

Instant relations include *wasBornIn, diedIn, hasWonPrize*, which are events that don't span an interval.
Short relations include *graduatedFrom, playsFor* whose duration averages less than 5 years.
Long relations include *isMarriedTo, isAffiliatedTo* whose duration averages more than 5 years.

E Comparison of filtering methods

In Table 11, we report the performance of most competitive baseline and TIMEPLEX, the reported performance use a filtering strategy that does not enumerate time points in an interval and filters out entities on exact matching time-interval. Note that our model consistently outperforms TNT-Complex, even with a stricter filtering.

F Ablation Study

In this study, we remove each component of TIMEPLEX (see equation 9) by making either κ=0 or λ=0, to understand the importance of each component (see Table 12).

G Details of Hyperparameters and Model training

All models are trained on a single NVIDIA Tesla K40 GPU. Our final model TIMEPLEX consist of a base model and two time-based gadgets.
TIMEPLEX*(base)* takes less than 10 minutes to train on all datasets except for ICEWS05-15, where it takes 80 minutes. Table 10 lists best hyperparameters of TIMEPLEX(base) on respective dataset.
Both gadgets are trained independently in less than 10 minutes. The parameter λ=5.0 gave best results for interval datasets, while λ=1.0 gave best results on event datasets. On Yago11k κ=3.0, while for rest κ=0.0. The gadget weights are L2 regularized, with a regularization penalty of 0.002. The model use 100 negative samples per correct fact for training.

H Model parameters

The number of parameters for the TIMEPLEXand baseline models are compared in Table 13.

	YAGO11k	WIKIDATA12k	ICEWS14	ICEWS05-15
Entities	10622	12554	7128	10488
Relations	10	24	230	251
#Instants	251	237	365	4017
#Intervals	6651	2564	0	0
Train	16408	32497	72826	368962
Valid	2051	4062	8941	46275
Test	2050	4062	8943	46092

Table 5: Details of datasets used.

Datasets→	YAGO11k				WIKIDATA12k			
↓Methods	TAC	gIOU	IOU	aeIOU	TAC	gIOU	IOU	aeIOU
HyTE	5.59	15.96	1.91	5.41	6.13	14.55	1.40	5.41
TNT-Complex	9.90	20.78	3.99	8.40	26.98	36.63	11.68	23.25
TIMEPLEX (base)	16.57	26.22	5.48	14.21	30.36	39.2	**13.20**	26.20
TIMEPLEX	**22.66**	**32.64**	**8.24**	**20.03**	**30.71**	**39.34**	13.15	**26.36**

Table 6: Time prediction performance using - TAC, gIOU, IOU and aeIOU

graduatedFrom → diedIn
graduatedFrom → hasWonPrize
wasBornIn → graduatedFrom
wasBornIn → diedIn
wasBornIn → isAffiliatedTo
wasBornIn → hasWonPrize
wasBornIn → playsFor
wasBornIn → worksAt
wasBornIn → isMarriedTo
isAffiliatedTo → diedIn
worksAt → diedIn
isMarriedTo → diedIn

Table 7: High confidence (99%) relation orderings extracted from YAGO11k.

educated at → position held
educated at → employer
educated at → member of
educated at → award received
educated at → academic degree
educated at → nominated for
instance of → head of government
residence → award received
academic degree → nominated for
spouse → position held
located in the administrative territorial entity → award received

Table 8: High confidence (99%) relation orderings extracted from WIKIDATA12k

I Training details of TIMEPLEX, HyTE

Each dataset spans along a *time range*, with a certain *time granularity*, which can be year, month or day. TIMEPLEX learns a time embedding for every point in this time range, discretized on the basis of the dataset's granularity (years for the interval datasets WIKIDATA12k and YAGO11k, and days for ICEWS datasets). At training time, TIMEPLEX looks at a single time point at a time - for this, we sample a time point uniformly at random from the query interval $[t_b, t_e]$ associated with the fact. In contrast, HyTE maps each time point to bin (heuristically determined), making the data granularity coarser, and learns representation of these bins. HyTE looks at time points in an interval as well, but enumerates each interval fact to produce a separate fact for each time point beforehand. Our method of sampling is efficient as the data size is unchanged. It also ensures each fact is sampled uniformly, not hurting link prediction performance by oversampling of long duration facts.

HyTE time prediction: HyTE can only predict a bin for the test fact. To convert predicted bins to years (or days), we take a mean of all years seen with the predicted bin and then do greedy coalescing to output time interval in years.

	Instant	Short	Long
TNT-Complex	4.24	16.34	3.73
Timeplex	**18.39**	**20.63**	**24.8**

Table 9: aeIOU@1 across relation classes on YAGO11k

J More diagnostics

We plot the normalized scores of TIMEPLEX, TIMEPLEX(base), and TNTComplex for different time queries in time range $[1850, 2010]$ in Table 14. Figure (a) highlights how TNTComplex model fails to learn that one cannot marry before birth. Figure (b) shows how with the limited background knowledge on the subject in question, TIMEPLEX can predict the gold time-interval.

	Learning Rate	Reg wt	Batch size	Temporal smoothing	α	β	γ
YAGO11k	0.1	0.03	1500	0.0	5.0	5.0	0.0
WIKIDATA12k	0.1	0.005	1500	0.0	5.0	5.0	5.0
ICEWS05-15	0.1	0.005	1000	5.0	5.0	5.0	5.0
ICEWS14	0.1	0.005	1000	1.0	5.0	5.0	5.0

Table 10: Hyperparameters for training TIMEPLEX(base) model embeddings on various datasets, tuned on MRR for validation set. Temporal smoothing was found to help on ICEWS datasets, however it gave no improvement for interval datasets. We tuned the parameters in a staged manner - first we tune learning rate (lr), regularization weight (r), batch size(b), and temporal smoothing weight (ts). We performed a random search in the following ranges: $lr \in [0.0001, 1.0]$, $r \in [0.0001, 1.0]$, $b \in [100, 5000]$, and $ts \in [0.0001, 10.0]$. The models were most sensitive to regularization weight and learning rate. After finding best values for these parameters, we tuned α, β and γ weights for each dataset, doing a grid search over the set $\{0.0, 2.0, 5.0, 7.0, 10.0\}$

.

	WIKIDATA12k			YAGO11k		
Method	MRR	HITS@1	HITS@10	MRR	HITS@1	HITS@10
TNT-Complex	27.35	17.59	48.51	15.78	10.21	28.64
TIMEPLEX	**30.61**	**20.79**	**51.78**	**22.77**	**16.33**	**36.3**

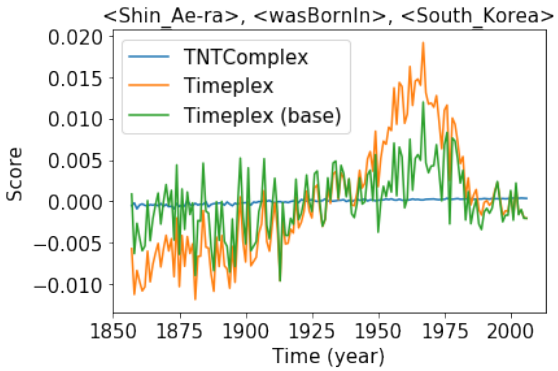
Table 11: Performance of the best models using a filtering strategy that does not enumerate time points in an interval, and filters on an exact match instead. We find that while TIMEPLEX convincingly outperforms the previous SOTA TNT-Complex using this filtering strategy as well.

Prediction task→	Link			Time interval
↓Method	MRR	HITS@10	HITS@1	aeIOU@1
TIMEPLEX	23.64	16.92	36.71	20.03
TIMEPLEX-Pair	23.15	16.63	36.27	14.21
TIMEPLEX- Rec	18.93	11.46	32.74	20.03
TIMEPLEX- Pair - Rec	18.35	10.99	31.86	14.21

Table 12: Ablation study on Yago11k. Recurrence feature significantly help in link prediction while relation pair feature helps time-interval prediction.

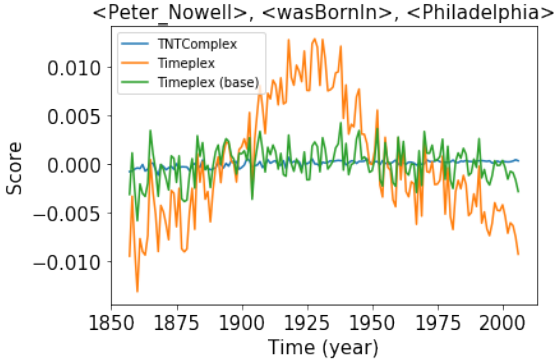
Models	Number of parameters
HytE	d($\|E\| + \|T\| + \|R\|$)
DE-SimplE	2d($(3\delta + (1-\delta))\|E\| + \|R\|$)
TNTComplex	2d($\|E\| + \|T\| + 4\|R\|$)
Timeplex(base)	2d($\|E\| + \|T\| + 6\|R\|$)
Timeplex	2d($\|E\| + \|T\| + 6\|R\|$) + 2($\|R\|^2$ + $\|R\|$)

Table 13: Number of parameters for each model. For HyTE we assume bucket size = 1 here. δ is the fraction of dimension to represent time in DA-SimplE model.



Info about query e1 in train set:	
<Shin_Ae-ra, isMarriedTo, Cha_In-pyo>(1995, 3000)	
Gold answer	1969
Timeplex prediction	1967
Timeplex (base) prediction	1967
TNTComplex prediction	2013

(a) TIMEPLEX, TIMEPLEX(base) both predict the correct answer but TNTComplex cannot model that one cannot marry before birth.



Info about query e1 in train set:	
<Peter_Nowell, graduatedFrom, Wesleyan_University>(1948, 3000) <Peter_Nowell, graduatedFrom, University_of_Pennsylvania>(1952, 3000)	
Gold answer	1928
Timeplex prediction	1928
Timeplex (base) prediction	1938
TNTComplex prediction	1918

(b) TIMEPLEX(base) cannot model that one is unlikely to graduate at the age of 10. TIMEPLEX (base) and TNTComplex do not have a clear vote like Timeplex.

Table 14: Comparing time prediction performance of TIMEPLEX, TIMEPLEX(base) and TNTComplex.