Time in a Box: Advancing Knowledge Graph Completion with Temporal Scopes

Ling Cai
lingcai@ucsb.edu
UC Santa Barbara

Krzysztof Janowicz
UC Santa Barbara

Bo Yan
UC Santa Barbara

Rui Zhu
UC Santa Barbara

Gengchen Mai
UC Santa Barbara

ABSTRACT

Almost all statements in knowledge bases have a temporal scope during which they are valid. Hence, knowledge base completion (KBC) on temporal knowledge bases (TKB), where each statement *may* be associated with a temporal scope, has attracted growing attention. Prior works assume that each statement in a TKB *must* be associated with a temporal scope. This ignores the fact that the scoping information is commonly missing in a KB. Thus prior work is typically incapable of handling generic use cases where a TKB is composed of temporal statements with/without a known temporal scope. In order to address this issue, we establish a new knowledge base embedding framework, called TIME2BOX, that can deal with atemporal and temporal statements of different types simultaneously. Our main insight is that answers to a temporal query always belong to a subset of answers to a time-agnostic counterpart. Put differently, time is a filter that helps pick out answers to be correct during certain periods. We introduce boxes to represent a set of answer entities to a time-agnostic query. The filtering functionality of time is modeled by intersections over these boxes. In addition, we generalize current evaluation protocols on time interval prediction. We describe experiments on two datasets and show that the proposed method outperforms state-of-the-art (SOTA) methods on both link prediction and time prediction.

CCS CONCEPTS

• **Computing methodologies → Knowledge representation and reasoning**; **Temporal reasoning**.

KEYWORDS

Temporal Knowledge Base, TIME2BOX, Link Prediction, Time Prediction

ACM Reference Format:
Ling Cai, Krzysztof Janowicz, Bo Yan, Rui Zhu, and Gengchen Mai. 2021. Time in a Box: Advancing Knowledge Graph Completion with Temporal Scopes. In *Proceedings of The Eleventh International Conference on Knowledge Capture (K-CAP 2021).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

K-CAP 2021, December 2 - 3, 2021, A virtual conference

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

REFERENCES

[1] Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2019. Tensor Decompositions for Temporal Knowledge Base Completion. In *ICML*.

A DATA STATISTICS

Generation of WIKIDATA114k. We extracted a sport-centric subgraph from WIKIDADA432k. We first picked out statements where the relation *memberOfSportsTeam* appears and obtained an entity set from those statements. Then we find all the statements that entities obtained from the previous step participate in as our initial subgraph. Finally, we ensure that each entity/relation is associated with at least 5 statements and the time period is restricted to [1883, 2023] for temporal statements, which encloses most of the temporal statements in the initial subgraph. This results in 1.7 million statements with 114k entities and 126 relations, and thus named as WIKIDATA114k. See Table 1 for data statistics.

		WIKIDATA12k	WIKIDATA114k
#entities		12,544	114,351
#relations		24	126
time period		[19, 2020]	[1883, 2023]
train	#all	32,497	1,670,969
	#time instant	14,099	175,637
	#start time only	4,089	44,809
	#end time only	1,273	2,164
	#full time interval	13,035	402,135
	#no time	0	1,046,224
valid	#all	4,051	11,720
	#time instant	1,857	1,177
	#start time only	322	342
	#end time only	76	11
	#full time interval	1,796	2,655
	#no time	0	7,535
#test	#all	4,043	11,854
	#time instant	1,844	1,219
	#start time only	324	306
	#end time only	56	15
	#full time interval	1,819	2,790
	#no time	0	7,524

Table 1: Statistics of these datasets used.

B HYPERPARAMETER SETTINGS

We tune models by the MRR on the validation set. Grid search is performed over negative samples k = [16, 32, 64, 128], learning rate lr = [0.003, 0.002, 0.001], batch size b = [1500, 2000, 2500, 3000, 3500]; dimension d = [200, 300, 400], and weight for time smoothness regularizer β = [0.0, 0.1, 0.001, 0.0001], as shown in Table 2.[1] We find that effects of different hyperparameters are minimal except for learning rate as the trained model usually converge to similar MRRs as long as they are trained thoroughly. We also observe that time smoothness regularizer is useful in learning time embeddings on WIKIDATA12k while failing to improve the model on WIKIDATA114k. This may be due to data sparsity with regard to time. As the time span of WIKIDATA114k is much smaller, time information is intensive and thus models are capable of learning temporal order between timestamps implicitly.

[1]Experiments are terminated after 10000 steps.

	#negative samples				# learning rate		
	16	32	64	128	0.003	0.002	0.001
MRR	36.02	36.68	37.06	37.30	36.64	36.82	37.30
MR	97	100	98	101	126	103	101
HITS@1	25.96	26.81	27.25	27.38	26.83	26.73	27.38
	#batch size				# dimension		
	2000	2500	3000	3500	200	300	400
MRR	36.71	36.87	37.30	36.78	36.12	36.88	37.30
MR	100	114	101	100	106	101	101
HITS@1	26.59	26.88	27.38	26.77	25.98	26.88	27.38

Table 2: Effects of hyper-parameters on WIKIDATA12k

C EXPERIMENTAL SETUP

Upon inspection on implementations of TKBC models, we find there are two common issues.

First, SOTAs only learn time embeddings for timestamps that appear in training set, which would be problematic at testing. For instance, suppose a sorted (ascending) list of timestamps occurring in training set is [1540, 1569, 1788, 1789, 1790], SOTAs only learn embeddings for these timestamps, while ignoring intermediate timestamps. As a result, they cannot answer queries when the associated time is not in the list, such as (s, r, ?o, *1955*). This problem would be even worse regarding time interval generation. As when we need to grow a time point to a time interval by extending it to the left or the right, we may jump from one year to a year far away from it. For instance, from 1569 to 1540 (left) or 1788 (right). This is not reasonable and thus may severely affect the evaluation on time prediction. In order to address this issue, we enumerate all the time points in the time span of the training set with a fixed granularity (i.e., year) and use them for all models at training periods.

The other issue is about the evaluation of link prediction task on time interval-based statements (including closed interval-based and left/right-open interval-based statements). In existing works, the evaluation boils down to assessing the correctness of answering a timestamp-based query by randomly picking one timestamp from a set of timestamps within the time interval and then measuring the performance on the newly generated query (i.e., the timestamp-based query). However, this is problematic. For closed interval-based samples, the evaluation results may vary from randomly sampled timestamps and thus may not be stable. For left/right-open interval-based statements, it is more severe. For instance, for a left-open interval-based test sample *(Albert Einstein, educatedAt, ?o, [-, 1905])*, Lacroix et al. [1] randomly pick a year before 1905, say 1000, and evaluate whether a model can output the correct answer (*University of Zurich*) to the new query *(Albert Einstein, educatedAt, ?o, 1000)*. Clearly, there is no correct answer at all since he was born in 1879. Therefore, the evaluation on such test samples may not be plausible. In order to address these issues, for a closed interval-based sample, we enumerate all the time points in the interval and do evaluation on each time point separately. Then we use the average performance over them as the overall evaluation. For the latter, we only consider the known endpoint in an interval, namely $(s, r, ?o, st)$ for right-open cases and $(s, r, ?o, et)$ for left-open cases.

D LINK PREDICTION PERFORMANCE BY TYPES OF VALIDITY INFORMATION

Table 3 shows the comparison between different methods in terms of different types of validity information.

Datasets	WIKIDATA12k							
Types	Time Interval (O)		Time Interval (C)		Time Instant		No Time	
Methods	TIMEPLEX base	TIME2BOX	TIMEPLEX base	TIME2BOX	TIMEPLEX base	TIME2BOX	TIMEPLEX base	TIME2BOX
MRR	46.74	**51.48**	25.30	**28.44**	41.11	**43.13**	-	-
MR	203	**68**	273	**84**	350	**125**	-	-
HITS@1	19.21	**41.05**	11.54	**18.5**	32.6	**33.30**	-	-
Datasets	WIKIDATA114k							
Types	Time Interval (O)		Time Interval (C)		Time Instant		No Time	
Methods	TIMEPLEX	TIME2BOX	TIMEPLEX	TIME2BOX	TIMEPLEX	TIME2BOX	TIMEPLEX	TIME2BOX
MRR	**22.63**	22.43	17.72	**18.85**	20.81	**21.32**	67.85	**68.40**
MR	346	**168**	155	**147**	**176**	**193**	430	**172**
HITS@1	4.98	**11.21**	3.94	**8.35**	11.07	**11.16**	**61.52**	60.30

Table 3: Link prediction evaluation by types of validity information. Time Interval (O) denotes left/right-open interval-based statements, and Time Interval (C) refers to closed interval-based statements.

E TIME PREDICTION PERFORMANCE BY DURATION LENGTH

Table 4 and 5 compare the performance of TIMEPLEX and TIME2BOX on the time prediction task across different duration lengths on two datasets. Test samples are first classified into three groups by duration (du) and then evaluate the performance of each group. For an interval I, $du = I_{max} - I_{min} + 1$. It shows that our improvements are more pronounced in terms of shorter durations in general.

	WIKIDATA12k					
Duration (du)	du=1		1<du<=5		du>5	
Method	TIMEPLEX base	TIME2BOX	TIMEPLEX base	TIME2BOX	TIMEPLEX base	TIME2BOX
gIOU@1	30.29	**38.09**	39.51	**43.68**	**47.4**	46.99
aeIOU@1	20.84	**28.34**	15.86	**22.95**	**18.23**	13.20
gaeIOU@1	12.47	**18.62**	11.73	**16.34**	**16.85**	11.20

Table 4: Time prediction by duration on WIKIDATA12k

	WIKIDATA114k					
Duration (du)	du=1		1<du<=5		du>5	
Method	TIMEPLEX base	TIME2BOX	TIMEPLEX base	TIME2BOX	TIMEPLEX base	TIME2BOX
gIOU@1	28.75	**37.03**	29.77	**38.36**	27.99	**39.07**
aeIOU@1	25.80	**34.16**	16.52	**21.54**	7.09	**9.94**
gaeIOU@1	14.69	**21.08**	10.50	**14.50**	3.85	**7.02**

Table 5: Time prediction by duration on WIKIDATA114k

F MODEL PARAMETER COMPARISON

Table 6 summarizes the number of parameters used in each method.

Models	**Number of parameters**
TNTComplex	$2d(\|E\| + \|T\| + 4\|R\|)$
TIMEPLEX base	$2d(\|E\| + \|T\| + 6\|R\|)$
TIME2BOX	$d(\|E\| + 2\|T\| + 2\|R\|) + 4d^2$

Table 6: Number of parameters for each model