Encoding Temporal Information for Time-Aware Link Prediction

Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Sujian Li, Baobao Chang and **Zhifang Sui**

Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University
Collaborative Innovation Center for Language Ability, Xuzhou 221009 China
{tingsong,tianyu0421,taoge,shalei,lisujian,chbb,szf}@pku.edu.cn

Abstract

Most existing knowledge base (KB) embedding methods solely learn from time-unknown fact triples but neglect the temporal information in the knowledge base. In this paper, we propose a novel time-aware KB embedding approach taking advantage of the happening time of facts. Specifically, we use temporal order constraints to model transformation between time-sensitive relations and enforce the embeddings to be temporally consistent and more accurate. We empirically evaluate our approach in two tasks of link prediction and triple classification. Experimental results show that our method outperforms other baselines on the two tasks consistently.

1 Introduction

Knowledge bases (KBs) such as Freebase (Bollacker et al., 2008) and YAGO (Fabian et al., 2007) play a pivotal role in many NLP related applications. KBs consist of facts in the form of triplets (e_i, r, e_j), indicating that head entity e_i and tail entity e_j is linked by relation r. Although KBs are large, they are far from complete. Link prediction is to predict relations between entities based on existing triplets, which can alleviate the incompleteness of current KBs. Recently a promising approach for this task called knowledge base embedding (Nickel et al., 2011; Bordes et al., 2011; Socher et al., 2013) aims to embed entities and relations into a continuous vector space while preserving certain information of the KB graph. TransE (Bordes et al., 2013) is a typical model considering relation vector as trans-

lating operations between head and tail vector, i.e., $\mathbf{e}_i + \mathbf{r} \approx \mathbf{e}_j$ when (e_i, r, e_j) holds.

Most existing KB embedding methods solely learn from time-unknown facts but ignore the useful temporal information in the KB. In fact, there are many temporal facts (or events) in the KB, e.g., (*Obama, wasBornIn, Hawaii*) happened at August 4, 1961. (*Obama, presidentOf, USA*) is true since 2009. Current KBs such as YAGO and Freebase store such temporal information either directly or indirectly. The happening time of time-sensitive facts may indicate special temporal order of facts and time-sensitive relations. For example, *(Einstein, wasBornIn, Ulm)* happened in 1879, *(Einstein, wonPrize, Nobel Prize)* happened in 1922, *(Einstein, diedIn, Princeton)* occurred in 1955. We can infer the temporal order of time-sensitive relations from all such kinds of facts: *wasBornIn → wonPrize → diedIn*. Traditional KB embedding models such as TransE often confuse relations such as *wasBornIn* and *diedIn* when predicting *(person,?,location)* because TransE learns only from time-unknown facts and cannot distinguish relations with similar semantic meaning. To make more accurate predictions, it is non-trivial for existing KB embedding methods to incorporate temporal order information.

This paper mainly focuses on incorporating the temporal order information and proposes a time-aware link prediction model. A new temporal dimension is added to fact triples, denoted as a quadruple: (e_i, r, e_j, t_r), indicating the fact happened at time t_r[1]. To make the embedding space compati-

---

[1] t_r is the absolute beginning time when the fact is true, e.g., "1961-08-04" for (*Obama, wasBornIn, Hawaii*).

2350

ble with the observed triple in the fact dimension, relation vectors behave as translations between entity vectors similarly as TransE models. To incorporate temporal order information between pair-wise temporal facts, we assume that prior time-sensitive relation vector can evolve into a subsequent time-sensitive relation vector through a temporal transition. For example, we have two temporal facts sharing the same *head entity*: (e_i, r_i, e_j, t_1) and (e_i, r_j, e_k, t_2) and the temporal order constraint $t_1 < t_2$, i.e., r_i happens before r_j, then we propose the assumption that prior relation $\mathbf{r}_i$ after temporal transition should lie close to subsequent relation $\mathbf{r}_j$, i.e., $\mathbf{r}_i \mathbf{M} \approx \mathbf{r}_j$, here matrix $\mathbf{M}$ captures the temporal order information between relations. In this way, both semantic and temporal information are embedded into a continuous vector space during learning. To the best of our knowledge, we are the first to consider such temporal information for KB embedding.

We evaluate our approach on public available datasets and our method outperforms state-of-the-art methods in the time-aware link prediction and triple classification tasks.

2 Time-Aware KB Embedding

Traditional KB embedding methods encode only observed fact triples but neglect temporal constraints between time-sensitive entities and facts. To deal with this limitation, we introduce *Time-Aware KB Embedding* which constrains the task by incorporating temporal constraints.

To consider the happening time of facts, we formulate a temporal order constraint as an optimization problem based on a manifold regularization term. Specially, temporal order of relations in time-sensitive facts should affect KB representation. If r_i and r_j share the same head entity e_i, and r_i occurs before r_j, then prior relation's vector r_i could evolve into subsequent relation's vector r_j in the temporal dimension.

To encode the transition between time-sensitive relations, we define a transition matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ between pair-wise temporal ordering relation pair (r_i, r_j). Our optimization requires that positive temporal ordering relation pairs should have lower scores (energies) than negative pairs, so we define a

temporal order score function as

$$g(r_i, r_j) = \|\mathbf{r}_i \mathbf{M} - \mathbf{r}_j\|_1, \tag{1}$$

which is expected to be a low score when the relation pair is in chronological order, and high otherwise.

To make the embedding space compatible with the observed triples, we make use of the triple set Δ and follow the same strategy adopted in previous methods such as TransE.

$$f(e_i, r, e_j) = \|\mathbf{e}_i + \mathbf{r} - \mathbf{e}_j\|_1. \tag{2}$$

For each candidate triple, it requires positive triples to have lower scores than negative triples.

The optimization is to minimize the **joint** score function,

$$
\begin{aligned}
L = \sum_{x^+ \in \Delta} \Big[\sum_{x^- \in \Delta'} [\gamma_1 + f(x^+) - f(x^-)]_+ \\
+ \lambda \sum_{y^+ \in \Omega_{e_i}, y^- \in \Omega'_{e_i}} [\gamma_2 + g(y^+) - g(y^-)]_+ \Big]
\end{aligned}
\tag{3}
$$

where $x^+ = (e_i, r_i, e_j) \in \Delta$ is the positive triple (quad), $x^- = (e'_i, r_i, e'_j) \in \Delta'$ is corresponding the negative triple. $y^+ = (r_i, r_j) \in \Omega_{e_i}$ is the positive relation ordering pair with respect to (e_i, r_i, e_j, t_x). It's defined as

$$
\begin{aligned}
\Omega_{e_i} = \{(r_i, r_j)|(e_i, r_i, e_j, t_x) \in \Delta_\tau, \\
(e_i, r_j, e_k, t_y) \in \Delta_\tau, t_x < t_y\},
\end{aligned}
\tag{4}
$$

where r_i and r_j share the same head entity e_i, and $y^- = (r_j, r_i) \in \Omega'_{e_i}$ are the corresponding negative relation order pairs by inverse. In experiments, we enforce constrains as $\|\mathbf{e}_i\|_2 \leq 1$, $\|\mathbf{r}_i\|_2 \leq 1$, $\|\mathbf{r}_j\| \leq 1$ and $\|\mathbf{r}_i \mathbf{M}\|_2 \leq 1$.

The first term in Equation (3) enforces the resultant embedding space compatible with all the observed triples, and the second term further requires the space to be temporally consistent and more accurate. Hyperparameter λ makes a trade-off between the two cases. Stochastic gradient descent (in mini-batch mode) is adopted to solve the minimization problem.

3 Experiments

We adopt the same evaluation metrics for time-aware KB embedding in two tasks: link prediction (Bordes et al., 2013) and triple classification (Socher et al., 2013).

Dataset	#Rel	#Ent	#Train/#Valid/#Test	#Trip.	#Quad
YG15k	10	9513	13345/1320/1249	15914	15914
YG36k	10	9513	29757/3252/3058	36067	15914

Table 1: Statistics of data sets.

3.1 Datasets

We create two temporal datasets from YAGO2 (Hoffart et al., 2013), consisting of time-sensitive facts. In YAGO2, *MetaFacts* contains all happening time for facts. *DateFacts* contains all creation time for entities. First, to make a pure time-sensitive dataset where all facts have time annotations, we selected the subset of entities that have at least 2 mentions in *MetaFacts* and *DateFacts*. This resulted in 15,914 triples (quadruples) which were randomly split with the ratio shown in Table 1. This dataset is denoted *YG15k*. Second, to make a mixed dataset, we created *YG36k* where 50% facts have time annotations and 50% do not. We will release the data upon request.

3.2 Link Prediction

Link prediction is to complete the triple (h, r, t) when h, r or t is missing.

3.2.1 Entity Prediction

Evaluation protocol. For each test triple with missing head or tail entity, various methods are used to compute the scores for all candidate entities and rank them in descending order. We use two metrics for our evaluation as in (Bordes et al., 2013): the mean of correct entity ranks (Mean Rank) and the proportion of valid entities ranked in top-10 (Hits@10). As mentioned in (Bordes et al., 2013), the metrics are desirable but flawed when a corrupted triple exists in the KB. As a countermeasure, we filter out all these valid triples in the KB before ranking. We name the first evaluation set as *Raw* and the second as *Filter*.

Baseline methods. For comparison, we select translating methods such as TransE (Bordes et al., 2013), TransH (Wang et al., 2014b) and TransR (Lin et al., 2015b) as our baselines. We then use time-aware embedding based on these methods to obtain the corresponding time-aware embedding models. A model with time-aware embedding is denoted as "tTransE" for example.

Implementation details. For all methods, we create 100 mini-batches on each data set. The di-

	Mean Rank		Hits@1 (%)	
Metric	Raw	Filter	Raw	Filter
TransE	1.53	1.48	69.4	73.0
tTransE	**1.42**	**1.35**	**71.1**	**75.7**
TransH	1.51	1.37	70.5	72.2
tTransH	**1.38**	**1.30**	**74.6**	**76.9**
TransR	1.40	1.28	71.1	74.3
tTransR	**1.27**	**1.12**	**74.5**	**78.9**

Table 3: Evaluation results on relation prediction.

mension of the embedding n is set in the range of $\{20,50,100\}$, the margin γ_1 and γ_2 are set in the range $\{1,2,4,10\}$. The learning rate is set in the range $\{0.1, 0.01, 0.001\}$. The regularization hyperparameter λ is tuned in $\{10^{-1},10^{-2},10^{-3},10^{-4}\}$. The best configuration is determined according to the mean rank in validation set. The optimal configurations are n=100,γ_1=γ_2=4,λ=10^{-2}, learning rate is 0.001 and taking ℓ_1-norm.

Results. Table 2 reports the results on the test set. From the results, we can see that time-aware embedding methods outperform all the baselines on all the data sets and with all the metrics. The improvements are usually quite significant. The Mean Rank drops by about 75%, and Hits@10 rises about 19% to 30%. This demonstrates the superiority and generality of our method. When dealing with sparse data YG15k, all the temporal information is utilized to model temporal associations and make the embeddings more accurate, so it obtains better improvement than mixing the time-unknown triples in YG36k.

3.2.2 Relation Prediction

Relation prediction aims to predict relations given two entities. Evaluation results are shown in Table 3 on only YG15K due to limited space, where we report Hits@1 instead of Hits@10. Example prediction results for TransE and tTransE are compared in Table 4. For example, when testing *(Billy_Hughes,?,London,1862)*, it's easy for TransE to mix relations *wasBornIn* and *diedIn* because they act similarly for a person and a place. But known that *(Billy_Hughes, isAffiliatedTo, National_Labor_Party)* happened in 1916, and tTransE have learnt temporal order that *wasBornIn→isAffiliatedTo→diedIn*, so the regularization term $|\mathbf{r}_{born}\mathbf{T} - \mathbf{r}_{affiliated}|$ is smaller than $|\mathbf{r}_{died}\mathbf{T} - \mathbf{r}_{affiliated}|$, so correct answer *wasBornIn* ranks higher than *diedIn*.

Dataset	YG15k				YG36k			
Metric	MeanRank		Hits@10(%)		MeanRank		Hits@10(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TransE	990	971	26.6	29.5	179	163	65.7	75.6
tTransE	**235**	**233**	**35.4**	**36.1**	**60**	**55**	**76.1**	**82.8**
TransH	986	966	25.7	28.0	174	158	65.3	77.8
tTransH	**232**	**230**	**36.1**	**37.2**	**61**	**54**	**76.6**	**82.9**
TransR	976	955	29.5	30.2	175	153	68.3	80.1
tTransR	**228**	**221**	**37.3**	**38.4**	**55**	**46**	**79.5**	**84.2**

Table 2: Evaluation results on link prediction.

Testing quad	TransE predictions	tTransE predictions
(Billy_Hughes,?,London,1862)	diedIn,**wasBornIn**	**wasBornIn**,diedIn
(John_Schoenherr,?,Caldecott_Medal,1988)	owns,**hasWonPrize**	**hasWonPrize**,owns
(John_G._Thompson,?,University_of_Cambridge,1961)	graduatedFrom,**worksAt**	**worksAt**,graduatedFrom
(Tommy_Douglas,?,New_Democratic_Party,1961)	isMarriedTo,**isAffiliatedTo**	**isAffiliatedTo**,worksAt

Table 4: Example results of relation prediction in descending order. Correct predictions are in **bold**.

3.3 Triple Classification

Triple classification aims to judge whether an unseen triple is correct or not.

Evaluation protocol. We follow the same evaluation protocol used in Socher et al. (2013). To create labeled data for classification, for each triple in the test and validation sets, we construct a corresponding negative triple by randomly corrupting the entities. To corrupt a position (head or tail), only entities that have appeared in that position are allowed. During triple classification, a triple is predicted as positive if the score is below a relation-specific threshold δ_r; otherwise as negative. We report averaged accuracy on the test sets.

Implementation details. We use the same hyperparameter settings as in the link prediction task. The relation-specific threshold δ_r is determined by maximizing averaged accuracy on the validation sets.

Results. Table 5 reports the results on the test sets. The results indicate that time-aware embedding outperforms all the baselines consistently. Temporal order information may help to distinguish positive and negative triples as different head entities may have different temporally associated relations. If the temporal order is the same with most facts, the regularization term helps it get lower energies and vice versa.

4 Related Work

Many models have been proposed for KB embedding (Nickel et al., 2011; Bordes et al., 2013; Socher et al., 2013). External information is employed to improve KB embedding such as text (Riedel et al.,

Datasets	YG15K	YG36K
TransE	63.9	71.9
tTransE	**75.0**	**82.7**
TransH	63.4	72.1
tTransH	**75.1**	**82.3**
TransR	64.5	74.9
tTransR	**78.5**	**83.9**

Table 5: Evaluation results on triple classification (%).

2013; Wang et al., 2014a; Zhao et al., 2015), entity type and relationship domain (Guo et al., 2015; Chang et al., 2014), and relation path (Lin et al., 2015a; Gu et al., 2015). However, these methods solely rely on triple facts but neglect temporal order constraints between facts. Temporal information such as relation ordering in text has been explored (Talukdar et al., 2012; Chambers et al., 2014; Bethard, 2013; Cassidy et al., 2014; Chambers et al., 2007; Chambers and Jurafsky, 2008). This paper proposes a time-aware embedding approach that employs temporal order constraints to improve KB embedding.

5 Conclusion and Future Work

In this paper, we propose a general time-aware KB embedding, which incorporates creation time of entities and imposes temporal order constraints on the geometric structure of the embedding space and enforce it to be temporally consistent and accurate. As future work: (1) We will incorporate the valid time of facts. (2) Some time-sensitive facts lack temporal information in YAGO2, we will mine such temporal information from texts.

Acknowledgments

References

Steven Bethard. 2013. Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, number EPFL-CONF-192344.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *ACL*.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. *ACL*, 94305:789–797.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 173–176. Association for Computational Linguistics.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *EMNLP*, pages 1568–1579.

MS Fabian, K Gjergji, and W Gerhard. 2007. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *16th International World Wide Web Conference, WWW*, pages 697–706.

Kelvin Gu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*.

Shu Guo, Quan Wang, Bin Wang, Lihong Wang, and Li Guo. 2015. Semantically smooth knowledge graph embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 84–94.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2015a. Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015b. Learning entity and relation embeddings for knowledge graph completion.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934.

Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. 2012. Acquiring temporal constraints between relations. In *CIKM*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014a. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).*, pages 1591–1601.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014b. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119.

Yu Zhao, Zhiyuan Liu, and Maosong Sun. 2015. Representation learning for measuring entity relatedness with rich information. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1412–1418. AAAI Press.