Learning BiLSTM-based Embeddings for Relation Prediction in Temporal Knowledge Graph

Sixia Ma1*, Aiping Li1, Xiaojuan Zhao1 and Yichen Song1

1 College of Computer, National University of Defense Technology, ChangSha, Hunan, China

*Corresponding author’s e-mail: masixia96@163.com

Abstract. At present, as the knowledge graph continues to increase, problems such as incomplete information in the graph also emerge, mainly reflected in the lack of links between entities [1], and the current commonly solution is to learn their representations that can represent more semantic information for the entities and relations in the graph. This article focuses on the temporal knowledge graph (TKG), consistent with the goal in the static knowledge graph, this paper tries to propose a better representation learning method to solve the link prediction problem. According to the time information in the temporal knowledge graph, the paper constructs a bag of words about time, uses the bag of words model to decompose the time in the temporal knowledge graph into sequences, and learns its semantic information together with relations and entities. The embeddings obtained in this way can learn more semantic information. In addition, proving the effectiveness of the model through experiments on the currently commonly used temporal knowledge graphs.

1. Introduction

The knowledge graph is mainly used to organize, manage and retrieve structured information, and collect and organize scattered human knowledge into a structured knowledge system [1]. The knowledge graph is composed of nodes and edges, generally expressed as \( G = (E, R) \), where \( E \) is the node set representing the entity, \( R \) is the edge set representing the relations between the entities, and the facts in the knowledge graph are generally represented by triples, that is \((s, r, o)\). For example, the fact triple: \((NingGaoning, chairmanof, COFCO)\) stated that Ning Gaoning is the chairman of COFCO. For the temporal knowledge graph is composed of nodes, edges and time, it is generally expressed as \( G = (E, R, T) \), where \( T \) is a set of time, and the facts in the temporal knowledge graph are generally represented by quadruples, namely \((s, r, o, t)\), where \( t \) can be time point information or time period information. For example, \((NingGaoning, chairmanof, COFCO, [2004 – 12 – 28, 2016 – 01 – ##])\) or \((NingGaoning, chairmanof, COFCO, 2010 – 12 – 02)\), the quadruple limits the time when the fact occurs based on the triple. Compared with the static knowledge graph, the temporal knowledge graph is closer to the facts in the real world that are only valid in a specific time period. The mainstream knowledge graph embedding method ignores time information when learning the embedding of nodes and relations. Therefore, there are more and more researches on temporal knowledge graphs, but the current research on temporal knowledge graphs usually studies the temporal consistency of the relations in the time dimension (temporal consistency: for an entity, the facts involved are time-constrained, e.g., for a person, he must experience birth first, then experience other events, and finally experience death.) [1] [4] [6] [27]. Or directly portray the time point as a plane, and study the rationality of the triples on a specific plane [2]. We found that most of the previous studies on the representation learning of the TKG

---

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd
took time as a constraint condition for the validation of triples, considering its structural characteristics on the timeline of the entire TKG, and ignored semantic information of time itself, such as quadruple: \( (\text{Ning Gaoming}, \text{chairman of}, \text{COFCO}, 2010 \rightarrow 12 \rightarrow 02) \), most researches focus on the structural information on the timeline involved in the entity (NingGaoming) at the time of (2010 \rightarrow 12 \rightarrow 02), and does not consider the semantic information of (2010 \rightarrow 12 \rightarrow 02) itself.

Compare to other methods, our method: consider the embedded representation of entities and relationships that change over time; consider both the structural characteristics of time and the characteristics of time: Contrast with the existing methods that only consider the semantic characteristics of time or only consider time. This method uses LSTM to mine the sequence characteristics of time, and draws on the word embedding method in natural language processing, and designs a word bag model, considering the semantic characteristics of time.

2. Related Work
At present, most of the knowledge graph representation learning methods are mostly for static knowledge graphs, such as commonly used methods TransE [8], TransH [16]. There are also many studies on the development status of static knowledge graphs [17] [18]. Our research mainly focuses on the TKG. However, we found that the TKG representation learning evolved from the static knowledge graph model at first, so we briefly introduce classic static knowledge graph model: TransE. TransE is inspired by the phenomenon of translation invariance, and regards the relation in the knowledge base as a translation vector between entities. In recent years, the study of temporal knowledge graph representation learning is divided into two categories according to the processing method of time: the first type of processing method is to combine time and relation, and for the second type of processing method, time is processed separately and then integrated with entity and relation.

The combination of time and relation: In [4], the model assumes that two time-ordered relations of the entity can evolve each other, based on this, it is used to separate the order of the relation involved in an entity. In [6], Use three different time processing methods to expand several classic static knowledge graph representations learning models, such as TransE and RESCAL. The time processing methods: relation and time constitute a composite relation code; time is expressed as embedding in the same dimension as entity and relation; time is used as a coefficient to affect the embedding of entity or relation. In [1], time and relation are combined into a new time-predicate for coding, time is first expressed as a sequence, and then joined with relations. In [27], in addition to using fact rationality to optimize embedding in the above methods, it also calculates useful context for each fact to measure time consistency to jointly improve the quality of embedding. But we can find that in this type, the entity representation does not have much sense of participation in the time dimension.

The time with entity and relation: Time is processed separately and then merged with entity and relation. In [2], Using the coordinate projection rule to project the entity and relation to the hyperplane where time is located, and then the TransE score function for optimization. In [7], models the occurrence of facts as a multidimensional point-in-time process. In [5] constructs a probability model to model the time, multiple multi-relations and concurrent interactions between entities to predict the occurrence of future events. In [17], the model uses a multi-dimensional additive time series decomposition composed of trend components (linear functions), seasonal components (sine functions) and random components (Gaussian noise) to capture the evolution of entities or relations.

3. Model
This chapter first introduces the symbols and definitions used in the paper, then introduces the main content of our method, and finally introduces the optimization methods used in the model.

3.1. Notations and definitions
The temporal knowledge graph (TKG) is formalized as \( G = (E, R, T) \), where \( E, R, T \) represent the set of entity, relation, and time respectively. In quadruples \( \text{fact} = (s, r, o, t), s, o \in E, s \) represents the head entity, \( o \) represents the tail entity; \( r \in R \), \( r \) represents the relation between the head entity and the
tail entity, \( t \in T \), \( t \) represents the time when the fact happened. The embedding model tries to learn the effective representation of entities and relations and the scoring function \( f \), so we describe the initial embedding of the head entity, relation and tail entity before training as \( s_{emb}, o_{emb} \) and \( r_{emb} \). After model training, the embedding of head entity, relation and tail entity is expressed as \( s_{tem}, o_{tem} \) and \( r_{tem} \), for the fact quadruples fact, \( f(\text{fact}) \) gives the possibility that fact is a valid quadruple.

3.2. Our model

Compared with the representation learning of the static knowledge graph, we study the representation learning of the TKG, and the most important thing is the processing of time and the fusion of time and entity. This section first proposes a new time processing method, and then explains how to combine the processed time information with the entity to obtain a new embedding with time information.

3.2.1. Time processing.

For the time format like \{####-##-##\}, it is composed of 6 digits, and each digit is a number from 0 to 9. We combined with practical considerations and set the highest digit of the year to 2. In addition, based on the current calendar setting, there are fixed 12 months in a year, with a maximum of 31 days per month.

We decompose the year digit by digit, the month and date are limited due to the number, and the number is not large, and can be directly added to the constructed time dictionary. We design time decomposition as a time dictionary, as shown in Figure 1.

\[
\begin{align*}
\text{year:} & \quad 0000, 000, 00, 4, 12, 28 \\
\text{month:} & \quad 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12 \\
\text{day:} & \quad 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
\end{align*}
\]

Figure 1: Time Dictionary

In Figure1, \( y \) represents the year, the year contains 4 digits, and the highest digit is 2; \( m \) represents the month, a total of 13 identifiers, of which 12 identifiers are the number of months among a year, and the other identifier 00m is the vacancy of the month in the graph; day is 32 identifiers, 31 of which are the days covered by a normal month, and the identifier 00d represents the vacancy of the day. Based on this, the time information obtained by decomposition is converted into the corresponding sequence according to the constructed time dictionary.

\[
\{2004-12-28\} \rightarrow \{2000,00,00,4,12,28\} \\
\{2000,00,00,4,12,28\} \rightarrow \{2000y, 000y, 00y, 4y, 12m, 28d\}
\]

TABLE 1: The form of time is year-month-day, \# indicates the information is missing. The year is decomposed based on its digits. The month and date correspond one by one according to the time dictionary. The last column in the table is the 6 ID numbers corresponding to the time sequence according to the time dictionary.

| Time       | Time Sequence | Corresponding ID |
|------------|---------------|------------------|
| 983-##-##  | 0000y,900y,80y,3y,00m,00d | 0,12,21,26,33,46 |
| 2016-01-## | 2000y,000y,10y,6y,01m,00d | 2,3,14,29,34,46 |
| 2004-12-28 | 2000y,000y,00y,4y,12m,28d | 2,3,13,27,45,74 |

3.2.2. BiLSTM-based temporal embedding model.

In TKG, many studies believe that the change over time is mainly the relations between entities, but as one of the main attributes of the TKG, the change over time should also be considered.

We first target the relation \( r \) and consider its embedding representation. In order to integrate relational embedding \( r_{emb} \) into time information, we concatenate relation embedding \( r_{emb} \) and time sequence embedding \( t_{emb} \) to obtain temporal relation sequence. The composed sequence
contains both relational information and time information, so we use it as an input to the BiLSTM model. BiLSTM is a combination of forward LSTM and backward LSTM. LSTM is a type of recurrent neural network (RNN) and is usually used to model contextual information in natural language processing tasks. The LSTM model can learn the information that needs to be memorized and the information that needs to be forgotten during the training process, and can capture long-distance dependence. Due to its design characteristics, LSTM is very suitable for modeling time series data. However, LSTM has a problem: it cannot encode back-to-front information, so Bi-LSTM can better capture the two-way semantic dependence. LSTM calculation formula:

\[ h = LSTM(x_t) \]

Figure 2: LSTM model calculation process

Forward LSTM:
We take the concatenation of \( r_{emb} \) and \( t_{emb} \) as the input of Forward LSTM.

\[ seq_{forward} = r_{emb} \| t_{emb} \]

where \( \| \) represents concatenate operation.

According to Figure 2, respectively calculate the memory gate, forget gate, cell state, hidden layer and output gate of the forward LSTM, and use the last hidden layer as the output of the forward LSTM. The final output is represented by the last hidden state, that is,

\[ h_L = LSTM(seq_{forward}) \]

Backward LSTM:
Compared with the forward LSTM, we hope that the backward LSTM collects the information that the forward LSTM has forgotten, so the backward input is exactly the opposite of the forward input order, and the backward LSTM first inputs the last id of the time (the day’s id representation), the initial embedding representation of the relation as the final input.

\[ seq_{backward} = t_{emb} \| r_{emb} \]

where \( t_{emb} \) is \( t_{emb} \) in reverse order. The operation is the same as the Forward LSTM, so the output of the Backward LSTM can be obtained as

\[ h_R = LSTM(seq_{backward}) \]

After the above operation, the obtained Forward LSTM output and the Backward LSTM output are concatenated to obtain the final relation embedding of the time information:

\[ r_{tem} = h_L \| h_R \]

Do the same operation for the entities in the fact quadruples to get the entity embedding representation incorporating time information \( s_{tem} \) and \( o_{tem} \).

3.2.3. Training details. In this part, we use the classic static knowledge graph embedding model TransE [8] training method. Input the transformed entities and relation into the scoring function of TransE, and calculate the score of the fact quadruples \( fact = (s, r, o, t) \):

\[ f(fact) = s_{tem} + r_{tem} - o_{tem} \| t_{1}/t_{2} \]
where $\|\cdot\|_{l_1/l_2}$ represents $l_1/l_2$ norm. In order to distinguish between valid and invalid quadruples, a margin-based scoring function is used as the training target, which is defined as follows:

$$
Loss(s, r, o, t) = \sum_{(s', r, o, t) \in T} [\gamma + f(s, r, o, t) - f(s', r, o, t)]_+
$$

where $[x]_+$ takes the positive part of $x$, $\gamma > 0$ is marginal hyperparameter, $(s, r, o, t) \in T$, $T$ is valid quadruples set, $T'$ is the set of invalid quadruples after negative sampling.

$$
T' = \{(s', r, o, t) | s' \in E - s \} \cup \{(s, r, o', t) | o' \in E - o \}
$$

![BiLSTM-based relation embedding representation learning](image)

Figure 3: BiLSTM-based relation embedding representation learning. The left side of the figure is the forward LSTM process, and the right side of the figure is the backward LSTM process. The final result is obtained by the forward and backward directions.

4. Experiment

We evaluate our model and compare with different state-of-the-art baselines based on Link prediction on four different temporal KG data sets.

4.1. Datasets

4.1.1. The most used data set in TKG research is the ICEWS data set. We choose two subsets of ICEWS as the experimental data set. Time annotations of ICEWS are time points.

- ICEWS14: ICEWS is a repository that contains political events with specific time annotations. ICEWS14 is subset of ICEWS, which corresponds to the facts in 2014;
- ICEWS05-15: ICEWS05-15 is subset of ICEWS, which corresponds to the facts between 2005 to 2015.

| Dataset   | #entities | #relations | #timestemps | #train  | #valid  | #test   | #timespan   |
|-----------|-----------|------------|-------------|---------|---------|---------|-------------|
| ICEWS05-15 | 10488     | 251        | 4017        | 368962  | 46275   | 46092   | 2005-2015   |
| ICEWS2014  | 6869      | 230        | 365         | 72826   | 8941    | 8963    | 2014        |

4.2. Experiment design

We use link prediction to evaluate the performance of the model. The task of link prediction is to predict the missing entities, experiment start from the perspective of predicting the head entity and tail entity of the fact quadruples. For each fact quadruples, perform negative sampling. First, perform negative sampling according to $(s, r, ?, t)$ for tail entity link prediction, and then perform negative sampling according to $(?, r, o, t)$ for head Entity link prediction. The negative sampling in this process is consistent with the above description, that is, random replacement of the head and tail entities that need to be predicted, so that the newly generated quadruples is not observed in the graph.

| Dataset   | ICEWS 2014 | ICEWS 2005-15 |
|-----------|------------|---------------|
| ICEWS     |            |               |
We sort the scores of all quadruples, including invalid quadruples and test quadruples, to get the ranking of test quadruples. For ranking, we use three evaluation indicators: MRR (mean reciprocal rank, higher is better), hit@k (higher is better, the proportion of the top k test quadruple rankings) and MR (mean rank, lower is better).

### 4.3. Model compared

In order to compare the performance of the model, we compare it with the classic static knowledge graph models of TransE and the following time-related models:

- **T-TransE [6]**: The role of time and relation is used to separate the order of the relation involved in the same entity. This method is based on the correlation of time-ordered relation-pairs, which can be implemented on the assumption of mutual evolution through the evolution matrix.

- **HyTE [2]**: Use the coordinate projection law to project the entities and relations to the hyperplane where time is located, and then use the TransE score function for optimization.

- **TA-TransE [1]**: Combine time and relation into a new time predicate, enter it into LSTM for encoding to get a new relation representation, and finally use TransE’s scoring function to optimize.

### 4.4. Result

#### 4.4.1. Implementation details.

We use the ADAM optimizer to train the model, fix the mini-batch size to 512, set the initial embedding dimensions of entity, relation, and time to 100, set the learning rate to 0.001, set the number of training to 500, and set the margin to 1.

#### 4.4.2. Result analysis.

Table 3 lists the results of the link prediction task in the TKG. For the selected evaluation indicators, our method has obtained good results in almost all cases.

In ICEWS2014, for the evaluation indicator MRR, our results are comparable to other models or slightly inferior to the HyTE model. We analysis that may be due to related reasons of the data set, because in ICEWS05-15, our method is better than HyTE for the same indicator nearly 4%. At the same time, it can be seen from Table III that our method performs well on Hit@10, for link prediction tasks, our method can provide a relatively high-precision entity candidate set, which can be used as a basis for future work to improve prediction accuracy.

### 5. Conclusion and future work

We propose a representation learning method based on BiLSTM to learn a TKG fact representation with time, which can be used with existing scoring functions for link prediction. The experiment in ICEWS proves the effectiveness of this method.

However, our work is only carried out on the data set where the time information is the time point. For the data set where the time information is the time period like wikidata, it has not been implemented, so our next step is to focus on how to integrate the time period information into the TKG facts.

### Acknowledgments

The work described in this paper is partially supported by the National Key R&D Program of China (No.2017YFB0802204, 2016QY03D0603, 2019QY1406, 2017YFB0803303), the Key R&D Program of Guangdong Province (No.2019B010136003), and the National Natural Science Foundation of China (No. 61732022,61732004, 62072131, 61672020).
References

[1] Alberto G., Sebastijan D. and Mathias N., (2018) Learning sequence encoders for temporal knowledge graph completion. In: EMNLP, pp. 4816-4821.

[2] Dasgupta S.S., Ray S.N., Talukdar P., (2018) Hyte: Hyperplane-based temporally aware knowledge graph embedding. In: EMNLP, pp. 2001-2011.

[3] Goel R., Kazemi S.M., Brubaker M., Poupart P., (2020) Diachronic embedding for temporal knowledge graph completion. In: AAAI 2020.

[4] Jiang T., Liu T., Ge T., Sha L., Chang B., Li S., Sui Z., (2016) Towards time-aware knowledge graph completion. In: COLING, pp. 1715–1724.

[5] Jin W., Jiang H., Qu M., Chen T., Zhang C., Szekely P. and Ren X., (2019) Recurrent event network: Global structure inference over temporal knowledge graph. In: ICLR-RLGM.

[6] Leblay J., Chekol M.W., (2018) Deriving validity time in knowledge graph. In: Companion of the The Web Conference 2018 on The Web Conference, pp. 1771–1776.

[7] Trivedi R., Dai H., Wang Y., Song L., (2017) Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In: ICML.

[8] Antoine B., Nicolas U., Alberto G., Jason W., and Oksana Y., (2013) Translating embeddings for modeling multirelational data.” In: In Advances in neural information processing systems, pp. 2787–2795.

[9] Yoon K., Yacine J., David S., and Alexander M., (2016) Character-aware neural language models. In: AAAI, pp.2741–2749.

[10] Yang B and Mitchell T, (2019) Leveraging knowledge bases in lstms for improving machine reading. In: arXiv preprint arXiv:1902.09091.

[11] C. Esteban, V. Tresp, Y. Yang, S. Baier, and D. Krompa, (2016) Predicting the co-evolution of event and knowledge graphs. In: the 19th International Conference on Information Fusion (FUSION), pp. 98–105.

[12] D. P. Kingma and J. L. Ba, (2015) Adam: A method for stochastic optimization. In: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf.Track Proc., pp. 1–15.

[13] Rudolf K., Ondrej B., and Jan K., (2017) Knowledge base completion: Baselines strike back. In: arXiv preprint arXiv:1705.10744.

[14] Nitish S., Geoffrey H., Alex K., Ilya S., and Ruslan S., (2014) Dropout: A simple way to prevent neural networks from overfitting. In: The Journal of Machine Learning Research, 15(1):1929–1958.

[15] Liu Y., Hua W., Xin K. and Zhou X., (2019) Context-Aware Temporal Knowledge Graph Embedding. In: WISE, pp.583-598.

[16] Wang Z., Zhang J., Feng J., and Chen Z., (2014) Knowledge graph embedding by translating on hyperplanes. In: AAAI, pp. 1112–1119.

[17] Lin Y., Han X., Xie R., Liu Z. and Sun M., (2018) Knowledge Representation Learning: A Quantitative Review. In: arXiv preprint arXiv:1812.10901.

[18] Ji S., Pan s., Cambria E., Marttinen P. and Yu P., (2020) A Survey on Knowledge Graphs: Representation, Acquisition and Applications. In: AAAI.

[19] Xu C., Nayyeri M., Alkhoufy F., Yazdi H., Lehmann J., (2020) Temporal Knowledge Graph Completion Based on Time Series Gaussian Embedding. In: ISWC, pp. 654-671.

[20] Zhao X., Jia Y., Li A., Jiang R. and Song Y., (2020) Multi-source knowledge fusion: a survey. In: World Wide Web23, pp. 2567–2592.