



Learning Temporal and Spatial Embedding for Temporal Knowledge Graph Reasoning

Yayao Zuo^(✉), Yang Zhou, Zhengwei Liu, Jiayang Wu, and Minghao Zhan

Guangdong University of Technology, Guangzhou, China
yyzuo@gdut.edu.cn

Abstract. Temporal knowledge graphs store a large number of temporal facts that simulate the dynamic interactions of entities along the timeline. Since existing temporal knowledge graphs often suffer from incompleteness, it is crucial to build time-aware representation learning models that help to infer the missing temporal facts. However, most of the existing models for temporal knowledge graph reasoning focus on mining temporal associations between entities, and do not fully exploit spatial information contained in entities. To this end, we propose spatial-temporal network(ST-Net), a new representation learning model for temporal knowledge graphs, which has both temporal and spatial awareness capabilities. Specifically, ST-Net enriches the hidden features of entities by simultaneously fusing their temporal and spatial information. At the same time, we introduce the core idea of Copy-Generation Networks, which predicts future facts based on either the historical vocabulary or the whole entity vocabulary. We evaluate our proposed method via link prediction at future times on three benchmark datasets. Through extensive experiments, we demonstrate ST-Net has superior performance on the future link prediction tasks.

Keywords: Temporal knowledge graph · Representation learning · Link prediction

1 Introduction

Knowledge graphs (KGs) is widely used in natural language processing applications due to its ability to represent structured knowledge. However, most KGs suffers from incompleteness, which limits the performance and scope of KG applications to a certain extent. Therefore, it is an important task to predict the missing facts by KG reasoning. Recently, inference task has been expanded from KGs to a more challenging field: Temporal Knowledge Graphs(TKGs). The inference task on TKGs can be simply expressed as predicting missing object entity in query (subject entity, relation, ?, timestamp). Most of their researches focus on how to effectively integrate temporal information into the model. It's suggested by some researches that the traditional KG embedding method can be extended to TKGs [1, 2]. However, they only pay attention to learning the potential representation of entity in a snapshot, while ignoring modeling the dynamic evolution

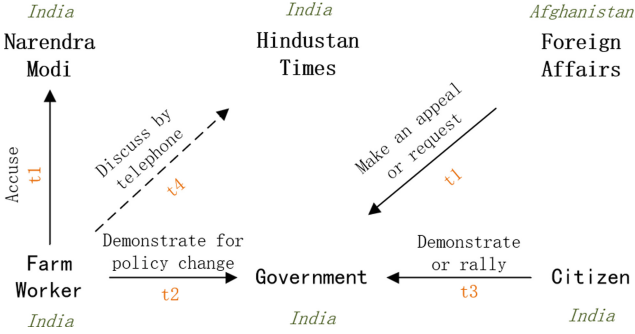


Fig. 1. Illustrates the process of ST-Net leveraging spatial information to predict future facts.

process of entity. Recently, some researchers have also conducted researches on the dynamic evolution of entity and relation [3–6], which combine historical information from previous snapshots. Another part of researchers try to enrich the potential representations of entity by incorporating multiple information [7–9], such as entity descriptions, event descriptions and uncertain information.

Inspired by this, we find that the above methods all ignore the importance of the spatial information (country, city, organization) attached to the entity. In fact, the occurrence of an event cannot be separated from spatial information, and the interaction between two entities also means the interaction between two spaces. Specifically, we found that the probability of entities in same event located in same space is 41.13% according to ICEWS18 dataset, which shows the importance of leveraging the spatial information contained in the entity to predict future facts.

To this end, we propose a innovative TKG representation learning model with both temporal and spatial perception. Mining the spatial information attached to the entity would help in fine-grained modeling of entity representation, and further strengthen the representation ability of entity embedding. At the same time, we introduce two inference modes in the copy generation network [6] to predict future facts from the historical vocabulary and the whole entity vocabulary. As shown in Fig. 1, after applying the spatial information, two nodes without interaction, *Farm Worker* and *Hindustan Times*, can be connected in the same space to pave the way for prediction.

We evaluated our proposed method on three benchmark datasets and the experimental results show that ST-Net is superior to the baseline model in link prediction task. The ablation experiment further proves that spatial information can help the model predict future facts better.

The main contributions of this work are as follows:

1. We propose a innovative TKG representation learning model ST-Net, which implements fine-grained modeling of entity representation by mining the spatial information attached to entity.

2. We introduce two inference modes in the Copy-Generation Networks [6], which predict future facts from the historical vocabulary and the whole entity vocabulary respectively.
3. We conduct extensive experiments on three public TKG datasets and demonstrate the effectiveness of ST-Net in link prediction.

2 Related Work

We discuss three relevant research directions. Since there is massive work in each direction, we can only select representative and closely related ones to elaborate.

2.1 Static Knowledge Graph Embeddings

Without considering temporal facts, researchers have made passable progress in KG embedding, which Ji et al. [10] summarizes. A classic class of models is translation model (TransE and its variants) [11–13], which leverages a distance-based scoring function to measure the reasonableness of facts. The other is the semantic matching model (such as ComplEx [14], DistMult [15], etc.), which uses a similarity-based scoring function and measure the plausibility of facts through semantic matching. Recently, some methods based on deep neural networks (such as R-GCN [16], ConvE [17], RSN [18], etc.) have emerged, which utilize CNN, RNN, and graph neural networks (GCN, R-GCN, etc.) to help models learn embedding of entity and relation. However, these methods cannot capture temporal facts.

2.2 Temporal Knowledge Graph Embeddings

TKG embedding incorporates timestamp of facts into the learning process. Some researchers attempt to extend the static KGs directly to the field of TKGs. TTransE [1] is an extension of TransE, which simply integrates time information into the scoring function; HyTE [2] replaces the unit normal vector of the hyperplane projection in TransH with a time-specific normal vector. Other researchers focus on the dynamic evolution of entity and relation. Know-Evolve [3] models the occurrence of facts as a temporal point process to learn non-linearly evolving entity representation over time; Goel et al. [4] provides the embedding of an entity at any point in time by equipping the static model with a diachronic entity embedding function. However, none of these methods correlate snapshots that have occurred in history.

To solve the problem that the model cannot capture the long-term dependency of facts, the autoregressive model is proposed. Jin et al. [5] modeled the TKGs in the way of autoregressive, that is, the snapshot at T timestamp depends on the historical snapshot before T ; Han et al. [19] leverages continuous temporal embedding to encode the temporal and structure information of historical snapshots; Zhu et al. [6] utilizes the recurrence rule of facts and combines two inferring modes to predict future facts from historical vocabulary and whole entity vocabulary respectively.

2.3 Embedding with Auxiliary Information

More recent attempts have been made to combine multimodal information with structured information in KGs to promote more effective knowledge representation. SSP [8] provides more precise semantic embedding for entity and relation by projecting triples and text descriptions into semantic subspaces. Glean [7] leverages graph neural networks to fuse unstructured event descriptions and structured data from TKGs to enrich the hidden features of event participants. The uncertain KG embedding model proposed by Chen et al. [9] incorporates the confidence scores of uncertain relation facts when learning embedding. Influenced by this, we try to fuse the spatial information contained in the entity with the structured information of the TKGs. As we know, this is the first work to incorporate spatial information into TKG representation learning.

3 Spatial-Temporal Network

In this section, we start with the notations for building our model and problem definition, and then introduce the model architecture as well as its training and inference procedures in detail. Figure 2 illustrates an overview of ST-Net reasoning process. ST-Net can be decomposed into 3 sub-modules: Query Vectorization, Capture History Vocabulary, and Copy-Generation Mode.

3.1 Notations

TKGs consists of temporal facts, each of which can be simply described as subject and object entity $s \in \mathcal{E}$ and $o \in \mathcal{E}$ have a relation $r \in \mathcal{R}$ at timestamp $t \in \mathcal{T}$, denoted as quadruple (s,r,o,t) , where \mathcal{E} , \mathcal{R} represent the vocabularies corresponding to the entities and relations respectively, and \mathcal{T} is the set of timestamps. The boldface $\mathbf{s}, \mathbf{r}, \mathbf{o}, \mathbf{t}$ represent the corresponding embedding vectors. A TKG can be divided into a set of snapshots $\{G_1, G_2, \dots, G_{t_k}\}$ according to the timestamps, where G_t is a snapshot of the TKG at time step t , containing all temporal facts at time step t . For each subject entity and relation pair at timestamp t_k , we define a delimited subset of \mathcal{E} specific to (s, r, t_k) as $\mathbf{H}_{t_k}^{(s,r)}$, namely historical vocabulary for (s, r, t_k) , which contains all object entities in temporal facts with the subject entity s and the relation r in the known snapshots $G_{(t_1, t_{k-1})} = \{G_{t_1}, G_{t_2}, \dots, G_{t_{k-1}}\}$ before t_k , where the historical vocabulary $\mathbf{H}_{t_k}^{(s,r)}$ is an N -dimensional multi-hot indicator vector and N is the cardinality of \mathcal{E} , the value of entities in the historical vocabulary are masked 1 while others are 0.

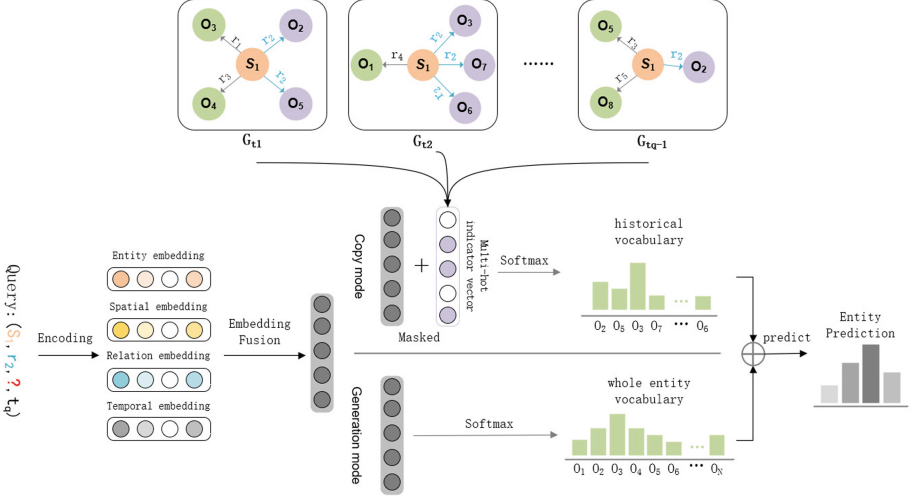


Fig. 2. Overview of ST-Net. ST-Net implement fine-grained modeling of entity representation by mining spatial information of entity, and combines two inference modes to predict missing entity in the query. In the figure, light purple nodes are the candidate object entities in the historical vocabulary for query $(s_1, r_2, ?, t_q)$.

Prediction of a missing temporal fact aims to infer the quadruple $(s, r, ?, t_q)$ or $(?, r, o, t_q)$ from the known snapshots $G_{(t_1, t_{q-1})} = \{G_1, G_2, \dots, G_{t_{q-1}}\}$. Without loss of generality, the task of the model is defined as predicting missing object entity.

3.2 Model Components

Query Vectorization. In encoding phase, the information contained in query needs to be converted into a continuous low-dimensional vector by the embedding layer. ST-Net first randomly initializes entity feature \mathbf{s}_i and relation feature \mathbf{r}_i for all $s \in \mathcal{E}$ and all $r \in \mathcal{R}$. In order for the model to be time-aware, the timestamp in temporal facts need to be encoded. Define the embedding for a unit step of time as \mathbf{t}_u and $\mathbf{t}_1 = \mathbf{t}_u$, so the embedding of timestamp \mathbf{t}_k is represented as follows:

$$\mathbf{t}_k = \mathbf{t}_{k-1} + \mathbf{t}_u \quad (1)$$

ST-Net expands the spatial-aware based on time-aware. Due to the complexity and uncertainty of data, the spatial information attached to entity is hard to learn by neural networks. We found that the text for most entities in TKGs already contains spatial information, like Citizen(India), Government(Pakistan). This is because these information have been summarized and sorted out in the process of temporal fact extraction, but they were not fully utilized. ST-Net obtains spatial information by preprocessing the text of the entity and the text in the raw data. After processing, for few of entities lacking spatial information,

the entity itself can be directly used as its spatial information or use the existing mature pretraining language model to generate these message by prompt learning. After acquiring the spatial information for all entities, ST-Net randomly initializes them to get the spatial feature \mathbf{sp} .

The discovered spatial information can support the model to carry out fine-grained modeling of entity representation, so that the entity has a richer and more favorable embedding to improve the effect of downstream tasks. After query Vectorization, the model needs to capture historical vocabulary of the query in advance to facilitate the next reasoning process.

Capture Historical Vocabulary. First, the training dataset can be divided into a series of snapshots $G_1, G_2, \dots, G_{t_{train}}$ by timestamp. Then obtain the historical vocabulary for each subject entity and relation combination (s, r, t) in each snapshot, i.e. $\{\mathbf{h}_{t_1}^{(s,r)}, \mathbf{h}_{t_2}^{(s,r)}, \dots, \mathbf{h}_{t_{train}}^{(s,r)}\}$. During training process, ST-Net is trained on each snapshot in timestamp order by incrementally maintain the historical vocabulary for all the previous snapshots. When evaluating the performance of our model on the validation set and test set, the maximum historical vocabulary from the whole training set will be used.

Specifically, for each query quadruple $(s, r, ?, t_k)$ at time t_k , during the training process, ST-Net will expand the historical vocabulary that specific to (s, r, t_k) from the snapshot before time t_k , as formalized below:

$$\mathbf{H}_{t_k}^{(s,r)} = \mathbf{h}_{t_1}^{(s,r)} + \mathbf{h}_{t_2}^{(s,r)} + \dots + \mathbf{h}_{t_{k-1}}^{(s,r)} \quad (2)$$

where $\mathbf{H}_{t_k}^{(s,r)}$ is an N-dimensional multi-hot indicator vector where 1 is marked for all entities in the current historical vocabulary. Next two modes of reasoning will be introduced.

Copy-Generation Mode. The Copy mode aims to identify repeated facts and predict future facts by down sampling known facts of the same type in history. Given a query $(s, r, ?, t_k)$ and its corresponding historical vocabulary $\mathbf{H}_{t_k}^{(s,r)}$, the copy mode will increase the probability estimated for the object entity that are selected in the historical vocabulary. The Copy mode generates the query vector v_q with an MLP:

$$\mathbf{v}_q = \tanh(\mathbf{W}_c[\mathbf{s}, \mathbf{sp}, \mathbf{r}, \mathbf{t}_k] + \mathbf{b}_c) \quad (3)$$

where $\mathbf{W}_c \in \mathbb{R}^{4d \times N}$ and $\mathbf{b}_c \in \mathbb{R}^N$ are trainable parameters. The query vector v_q is an N-dimensional vector, where N is the cardinality of the whole entity vocabulary \mathcal{E} .

To minimize the probability of some entities that do not form known facts with s and r in history, we first change the index value for an uninterested entity in $\mathbf{H}_{t_k}^{(s,r)}$ to a small negative number denoted $\mathbf{H}_{t_k}^{\dot{(s,r)}}$. Then, the Copy mode can add the query vector and the changed multi-hot indicator vector $\mathbf{H}_{t_k}^{\dot{(s,r)}}$ to limit

the scope of candidate entities. After applying softmax function, the probability of the uninterested entities will be minimized.

$$\mathbf{c}_q = \mathbf{v}_q + \mathbf{H}_{t_k}^{(s,r)} \quad (4)$$

$$\mathbf{p}(c) = \text{softmax}(\mathbf{c}_q) \quad (5)$$

where \mathbf{c}_q is an N-dimensional index vector. $\mathbf{p}(c)$ is an N-dimensional probability distribution vector representing the prediction probabilities on the historical vocabulary. Finally, we use the entity with the largest probability value in $\mathbf{p}(c)$ to answer the query. The advantage of the Copy mode is that it enable to predict from a more limited candidate entity space than the overall vocabulary. However, facts can also appear in upcoming snapshot, therefore a Generation mode is needed to predict such facts.

Given the same query $(s, r, ?, t_k)$ as the copy mode, the Generation mode is responsible for selecting the object entity from the whole entity vocabulary \mathcal{E} to predict facts. The reasoning of Generation mode is that regards the predicted fact as a completely new fact without reference to the facts that have happened in history. The Generation mode also generates a query vector \mathbf{g}_q and is further normalized using the softmax function for prediction:

$$\mathbf{g}_q = \mathbf{W}_g[\mathbf{s}, \mathbf{sp}, \mathbf{r}, \mathbf{t}_k] + \mathbf{b}_g \quad (6)$$

$$\mathbf{p}(g) = \text{softmax}(\mathbf{g}_q) \quad (7)$$

where $\mathbf{W}_g \in \mathbb{R}^{4d \times N}$ and $\mathbf{b}_g \in \mathbb{R}^N$ are trainable parameters. Similar to $\mathbf{p}(c)$ in the Copy mode, $\mathbf{p}(g)$ represents the predicted probability on the entire entity vocabulary. The maximum value in \mathbf{p}_g denotes the object entity predicted by Generation mode throughout the entity vocabulary. The Generation mode is complementary to the Copy mode, with the ability to predict entirely new facts.

3.3 Parameter Learning and Inference

Given a query $(s, r, ?, t)$ to predict the object entity can be viewed as a multi-class classification task, where each class corresponds to each object entity. The learning objective is to minimize the following cross-entropy loss \mathcal{L} :

$$\mathcal{L} = - \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{E}} \sum_{k=1}^K o_{it} \ln \mathbf{p}(y_{ik} | s, r, t) \quad (8)$$

where o_{it} is the i-th ground truth object entity in the snapshot G_t , $\mathbf{p}(y_{ik} | s, r, t)$ is the combined probability value of the k-th object entity in the snapshot G_t when the i-th ground truth object entity is o_i .

In order to ensure that the sum of the probability equals 1 for all entities in \mathcal{E} , we set a hyperparameter α to adjust the weight between the Copy mode and the Generation mode, which is defined as follows:

$$\mathbf{p}(o | s, r, t) = \alpha * \mathbf{p}(c) + (1 - \alpha) * \mathbf{p}(g) \quad (9)$$

$$o_t = \operatorname{argmax}_{o \in \mathcal{E}} \mathbf{p}(o|s, r, t) \quad (10)$$

where $\alpha \in [0, 1]$, $\mathbf{p}(o|s, r, t)$ is the final predicted probability vector, which contains the probabilities of all entities under the current query.

4 Experiments

In this section, we evaluate our proposed method on the link prediction task on three public TKG datasets. We first introduce experimental settings in detail, including details of datasets and baselines. After that, we compare and analyze the experimental results and conduct an ablation study to evaluate the importance of spatial information.

4.1 Experimental Setup

Table 1. Statistics of the datasets.

Data	Entities	Relation	Training	Validation	Test	Granularity	Time granules
ICEWS18	23,033	256	373,018	45,995	49,545	24 h	304
ICEWS14	7,128	230	74,845	8514	7371	24 h	365
ICEWS05-15	10,488	251	368,868	46,302	46,159	24 h	4,017

Datasets and Evaluation Metrics. We evaluate ST-Net on three benchmark datasets for link prediction: ICEWS18 [20], ICEWS14 [21] and ICEWS05-15 [21]. Table 1 provides a summary of these datasets statistics. We divide each dataset into training set, validation set and testing set into 80%/10%/10% splits in the chronological order, and adopt a filited version of Mean Reciprocal Ranks(MRR) and Hits@1/3/10 to measure the performance of ST-Net.

Baselines. We compare ST-Net with multiple static knowledge graph embedding(SKGE) and temporal knowledge graph embedding(TKGE) models. The former includes TransE [11], DistMult [15], ComplEx [14], RotatE [22], and SimpleE [23], while the latter includes TTransE [1], HyTE [2], TA-DistMult [21], DE-DistMult [4], DE-SimpleE [4], RE-Net [5], CyGNet [6], and ATiSE [24].

Parameter Settings. The value of the hyperparameter α used to tune the weight of Copy mode and Generation mode is determined based on the MRR performance on the validation set of the current dataset. After extensive experiments, we found that the model works best when α is set to 0.8 on ICEWS14 and ICEWS18 and 0.9 on ICEWS05-15. The parameters of the model are initialized with Xavier initialization (Glorot and Bengio 2010) [25], and then optimized using an AMSGrad optimizer with a learning rate of 0.001. The batch size is set to 1024. The hidden layer embedding dimension is set to 200. The training epoch is limited to 30, which is enough for the model to converge in most cases. The baseline results are adopted from CyGNet [6] and ATiSE [24].

4.2 Results

Table 2. Results on ICEWS18, ICEWS14 and ICEWS05-15. The best results are in bold, and the second best ones are underlined.

Method	ICEWS18				ICEWS14				ICEWS05-15			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	17.56	2.48	26.95	43.87	28.00	9.40	–	63.70	29.40	9.00	–	66.30
DistMult	22.16	12.13	26.00	42.18	43.90	32.30	–	67.20	45.60	33.70	–	69.10
ComplEx	30.09	21.88	34.15	45.96	46.70	34.70	52.70	71.60	48.10	36.20	53.50	72.90
RotatE	23.10	14.33	27.61	38.72	41.80	29.10	47.80	69.00	30.40	16.40	35.50	59.50
Simple	–	–	–	–	45.80	34.10	51.60	68.70	47.80	35.90	53.90	70.80
TTransE	8.36	1.94	8.71	21.93	25.50	7.40	–	60.10	27.10	8.40	–	61.60
HyTE	7.31	3.10	7.50	14.95	29.70	10.80	41.60	65.50	31.60	11.60	44.50	68.10
TA-DistMult	28.53	20.30	31.57	44.96	47.70	36.30	–	68.60	47.40	34.60	–	72.80
DE-DistMult	–	–	–	–	50.10	39.20	56.90	70.80	48.40	36.60	54.60	71.80
DE-Simple	–	–	–	–	<u>52.60</u>	41.80	<u>59.20</u>	<u>72.50</u>	51.30	39.20	57.80	<u>74.80</u>
RE-NET	42.93	36.19	45.47	55.80	45.71	38.42	49.06	59.12	–	–	–	–
CyGNet	<u>46.69</u>	<u>40.58</u>	<u>49.82</u>	<u>57.14</u>	49.99	43.51	53.55	61.42	<u>57.22</u>	<u>50.25</u>	<u>61.77</u>	68.58
ATiSE	–	–	–	–	55.00	<u>43.60</u>	62.90	75.00	51.90	37.80	60.60	79.40
ST-Net	47.47	41.07	50.62	58.52	51.84	44.74	55.59	64.57	58.34	51.26	62.85	70.22

Table 2 report the link prediction results of ST-Net and baseline methods on three TKG datasets. We observe that all SKGE methods perform worse than most TKGE methods, because they cannot capture temporal information in facts and cannot model dynamic interactions of entities and relations. However, their performance generally outperforms TTransE and HyTE. The reason we believe is that TTransE and HyTE only learn the representation at this timestamp for each snapshot separately, without linking entity representations at different timestamps, so it lacks the ability to capture time sequence information.

Table 2 also shows that ST-Net significantly outperforms other baselines on ICEWS18 and ICEWS05-15. For further analysis, we calculated the probability of repeated events in ICEWS18 to be 49.24%, which will improve the prediction effect of the Copy mode to a certain extent. And in ICEWS18, as many as 41.3% of the groups where the subject entity and the object entity are in the same space, which also explains that the reasonable use of spatial information can improve the prediction result. We argue that spatial message between entities in a fact can also help the model learn the semantic information implied by the relation, that is, the relation often occurs between those spatial locations.

The experimental results indicate that ST-Net uses the spatial information attached to the entity to implement fine-grained modeling of entity representation, which can effectively enrich entity information and improve the effect for link prediction.

4.3 Ablation Study

Table 3. Results (in percentage) of ablation study on the ICEWS18.

Method	ICEWS18			
	MRR	Hits@1	Hits@3	Hits@10
ST-Net-No-Spatial-Information	46.68	40.72	49.67	56.82
ST-Net-Prompt(bert-base-cased)	47.03	40.86	50.12	57.60
ST-Net	47.47	41.07	50.62	58.52

In order to explore the impact of different acquisition of spatial information on the model, we conducted an ablation study. To do this, we create variants of ST-Net by adjusting how spatial information is obtained in the model, and compare their performance gaps with ST-Net on the ICEWS18 dataset. From the results in Table 3, we can observe the importance of spatial information. After removing the spatial information, all metrics of the model have decreased, which demonstrates that the spatial information contained in entity play a vital part for predicting future facts. In addition, ST-Net-Prompt(bert-base-cased) complete the missing spatial information of entity through the Fill-Mask task in the pre-trained language model (bert-base-cased), where the template of the Fill-Mask task is set to : [Entity text]’s country is [MASK]. Its performance ranks between ST-Net and ST-Net-No-Spatial-Information, which further illustrates the importance of spatial information and reflects the shortcomings of the method we designed to complete the spatial information through the pre-trained model.

5 Conclusion

In this paper, we explore the spatial information attached to the entity and predict the future facts by combining the copy and generation reasoning mode. The experimental results show that ST-Net has promising performance in predicting the future facts in the temporal knowledge graph. In the future work, we plan to mine more accurate and effective information in entity through prompt learning, and combine them to help the model perform better. Meanwhile, further study on the utilization of spatial information is also significant.

Acknowledgements. The research is supported by The Natural Science Foundation of Guangdong Province (No.2018A030313934).

References

1. Leblay, J., Chekol, M.W.: Deriving validity time in knowledge graph. In: Companion Proceedings of the The Web Conference 2018, pp. 1771–1776 (2018)
2. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HyTE: hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2001–2011 (2018)
3. Trivedi, R., Dai, H., Wang, Y., Song, L.: Know-evolve: deep temporal reasoning for dynamic knowledge graphs. In: International Conference on Machine Learning, pp. 3462–3471. PMLR (2017)
4. Goel, R., Kazemi, S.M., Brubaker, M., Poupart, P.: Diachronic embedding for temporal knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3988–3995 (2020)
5. Jin, W., Qu, M., Jin, X., Ren, X.: Recurrent event network: autoregressive structure inference over temporal knowledge graphs. arXiv preprint [arXiv:1904.05530](https://arxiv.org/abs/1904.05530) (2019)
6. Zhu, C., Chen, M., Fan, C., Cheng, G., Zhan, Y.: Learning from history: modeling temporal knowledge graphs with sequential copy-generation networks. arXiv preprint [arXiv:2012.08492](https://arxiv.org/abs/2012.08492) (2020)
7. Deng, S., Rangwala, H., Ning, Y.: Dynamic knowledge graph based multi-event forecasting. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1585–1595 (2020)
8. Xiao, H., Huang, M., Meng, L., Zhu, X.: SSP: semantic space projection for knowledge graph embedding with text descriptions. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
9. Chen, X., Chen, M., Shi, W., Sun, Y., Zaniolo, C.: Embedding uncertain knowledge graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3363–3370 (2019)
10. Ji, S., Pan, S., Cambria, E., Marttinen, P., Philip, S.Y.: A survey on knowledge graphs: representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(2), 494–514 (2021)
11. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26 (2013)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI Conference on Artificial Intelligence (2015)
13. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
14. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080. PMLR (2016)
15. Yang, B., Yih, W.T., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint [arXiv:1412.6575](https://arxiv.org/abs/1412.6575) (2014)
16. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) *ESWC 2018*. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
17. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

18. Guo, L., Sun, Z., Hu, W.: Learning to exploit long-term relational dependencies in knowledge graphs. In: International Conference on Machine Learning, pp. 2505–2514. PMLR (2019)
19. Han, Z., Ding, Z., Ma, Y., Gu, Y., Tresp, V.: Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 8352–8364 (2021)
20. Boschee, E., Lautenschlager, J., O’Brien, S., Shellman, S., Starz, J., Ward, M.: ICEWS coded event data. Harvard Dataverse, vol. 12 (2015)
21. García-Durán, A., Dumančić, S., Niepert, M.: Learning sequence encoders for temporal knowledge graph completion. arXiv preprint [arXiv:1809.03202](https://arxiv.org/abs/1809.03202) (2018)
22. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint [arXiv:1902.10197](https://arxiv.org/abs/1902.10197) (2019)
23. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
24. Xu, C., Nayyeri, M., Alkhoury, F., Yazdi, H.S., Lehmann, J.: Temporal knowledge graph embedding model based on additive time series decomposition. arXiv preprint [arXiv:1911.07893](https://arxiv.org/abs/1911.07893) (2019)
25. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256. JMLR Workshop and Conference Proceedings (2010)