انجمن کامپیوتر ایران
Computer Society Of Iran

# 28th International Computer Conference

## Computer Society of Iran

25rd and 26th Jan. 2023, Sharif University of Technology

Sharif University of Technology

## PRESENTATION CERTIFICATE

This is to certify that the paper with title

# Dynamic knowledge graph completion through time-aware relational message passing

by

**Amirhossein Baqinejadqazvini and Saedeh Tahery and Saeed Farzi**

has been presented in the conference.

Jafar Habibi
General Chair

Alireza Ejlali
Technical Chair

# Dynamic knowledge graph completion through time-aware relational message passing

Amirhossein Baqinejadqazvini
*Department of Software Engineering*
*Faculty of Computer Engineering*
K. N. Toosi University of Technology
Tehran, Iran
baqinejad@email.kntu.ac.ir

Saedeh Tahery
*Department of Artificial Intelligence*
*Faculty of Computer Engineering*
K. N. Toosi University of Technology
Tehran, Iran
saedeh.tahery@email.kntu.ac.ir

Saeed Farzi
[1] *Department of Software Engineering*
*Faculty of Computer Engineering*
K. N. Toosi University of Technology
Tehran, Iran
[2] *School of Computer Science*
Institute for Research in Fundamental Sciences (IPM)
Tehran, Iran
saeedfarzi@kntu.ac.ir

***Abstract*— As the structure of knowledge graphs may vary over time, static knowledge graph completion methods do not deal with time-varying knowledge graphs. However, examining the paths between entities and entities' context information can lead to more accurate completion methods. This paper attempts to complete dynamic (time-varying) knowledge graphs by combining time-aware relational paths and relational context. The proposed model can improve dynamic knowledge graph completion methods by leveraging neural networks. Experimental results conducted on two standard datasets, ICEWS14 and ICEWS05-15, indicate our model's superiority in terms of Mean Reciprocal Rank (MRR) and Hit@k over its well-known counterparts, such as DE-TransE and DE-DistMult.**

***Keywords—knowledge graph completion, dynamic knowledge graphs, time-aware relational context, time-aware relational paths***

## I. Introduction

Knowledge graphs are a particular type of graph storing information in a structured way. A knowledge graph consists of a set of triples, *so-called fact*, where each triple states a relationship between the head and the tail entities through a relation of type *r* [1].

A knowledge graph consists of thousands or millions of entities and relations. Regardless of how large a knowledge graph is and how many facts exist, they are far from perfect and complete. Thus, knowledge graph completion is a critical topic that predicts new connections between entities.
Many researchers so far have employed knowledge graphs in various fields, including social chatbots [2], digital assistants such as XiaIoc [3], recommender systems such as the DKN news recommendation system [4], medical and health [5], education [6], stock price prediction [7], and machine translation model construction [8, 9].

Previous static methods for completing knowledge graphs did not consider the change of these graphs over time, which causes predictions to become less accurate as time passes. Therefore, modeling knowledge graphs in a static and unchanged form cannot address the change in relationships between entities over time. However, dynamic (i.e., temporal) knowledge graphs are time-varying graphs that predict new relationships between entities more accurately [10].

Although static knowledge graphs do not incorporate time, methods proposed to complete them have become appropriate ones during the years [11-13]. One of these methods is to consider the path information in the knowledge graph and the context information of the entities. By using the entities' context information, it is possible to understand the type of entities and their role. For example, entities can represent human characters, objects, places, and more. It is also possible for a person to have a social personality and play different roles at the same time as being a member of a family. Therefore, finding out the entity's type and role can be reached by examining the entities' context information with respect to their neighbor relations. Furthermore, examining the information of the paths in the knowledge graph can help predict new relationships between entities more accurately. Suppose a person works as a cameraman in a moviemaking process and another as a director. While there is no connection between the two mentioned characters in the knowledge graph, the path information between the first and the second character reveals a coworker relationship. Examining the information of entities' context and paths in the knowledge graph has led to better models than when only considering information of the triples in the knowledge graph.

The absence of efficient and innovative methods, such as examining the context and path information in dynamic knowledge graphs and modeling time in static knowledge graphs, has led us to propose a new model to complete dynamic knowledge graphs following time-aware approaches. In this paper, we first study the effect of time in modeling the paths in knowledge graphs and entities' context information. Then, we use vectors representing this information to complete the dynamic knowledge graphs. Our proposed model results in remarkable performance in terms of Mean Reciprocal Rank (MRR) and Hit@k by taking advantage of state-of-the-art methods for completing static knowledge graphs and incorporating time into the model.

To show the performance and evaluate our model, we use two standard datasets named ICEWS14 and ICEWS05-15. In doing so, one of the head or tail entities is masked for each fact, and two questions are arisen:

1- Which entity is the head entity related to given the relation *r*?

2- What entity establishes a relationship with the tail entity given the relation *r*?

In answering these questions, the model determines several candidate entities. Two ranking metrics, MRR and Hit@k, examine the rank of the entity that is the correct answer to the question among the candidate entities. Our model achieves the MRR of 0.638 and 0.708 and the Hit@1 of 52.47 and 62.25 on ICEWS14 and ICEWS05-15 datasets. Comparing the results with the counterparts demonstrates that our model can improve the dynamic knowledge graph

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

completion performance. It is of note that the previous models evaluated on the two mentioned datasets only cared about the information of the triples in the knowledge graph and influencing time in their modeling. However, the improvement achieved by our model is arisen not only from using the information of triples and incorporating time in the model but also from the knowledge graph path and the entities' context information.

The rest of the paper is organized as follows. Related work is reviewed in Section II, and the proposed method is explained in Section III. The experimental studies conducted to evaluate the method are reported in Section IV. Finally, in Section V, a conclusion is drawn.

## II. Related Work

This section reviews different static and dynamic knowledge graph completion methods and their advantages and disadvantages. Various methods so far have been introduced to complete knowledge graphs. Embedding-based is one of the essential methods that consist of two categories of translational distance and semantic matching models. TransE [14], TransH [15], TransR [16], TransD [17], and TransM [18] are some examples for translational distance, while RESCAL [19] and DistMult [20] fall into the semantic matching category. The main focus of these models was on triples' information, but none considered the knowledge graph path and the entities' context information. On the other hand, some models used more information available on knowledge graphs [11-13].

KPRN [13] is a model that developed a solution for reasoning on paths in a knowledge graph to infer user preference. KPRN tried to incorporate knowledge graphs into recommender systems. Although the paths in the knowledge graph were remarked in [13], the entities' context information was not addressed.

CoKE [11] is a model in which edges and paths were formalized as sequences of entities and relations. Given an input sequence, CoKE used a stack of Transformers [21], encoded the input, and obtained contextualized representation for its components. It was suitable for static knowledge graphs and did not consider the entities' context.

PathCon [12] used paths and entities' context information in a knowledge graph to infer new relationships between entities. Using rich information in the entities' context and paths between entities is the main advantage of PathCon. However, it was used for completing static knowledge graphs. We engaged this model in ours for dynamic knowledge graph completion in this paper.

It is of note that instead of triples, there are four factors in dynamic knowledge graphs, including head and tail entities, relation, and time of relation between them. Time causes some differences in static and dynamic knowledge graphs as follows:

- An edge may have specific endpoint nodes which can be changed over time.
- A node's adjacent edges may differ at two time periods.
- Head and tail pairs have different context information over time; hence, the entities' context information should be specified regarding time.
- The paths between the head and tail entities may differ at different periods.

Not all paths between entities are equally significant. Therefore, measuring each path’s importance is necessary according to the time related to the path between head and tail entities.

It is enough in dynamic knowledge graph completion methods to model the time factor like other entities and relations. After modeling the time, one can follow the idea of static knowledge graph completion methods to complete dynamic ones. Two methods are usually used for modeling time [22]. In the first method, it is regarded as an independent factor, and like entities and relations, it has a corresponding vector. In the second one, the effect of this factor is considered in modeling the vectors related to entities and relations. Thus, there is no separate vector for time, and the vectors related to entities and relations reflex static and dynamic characteristics. The diachronic embedding [10] method took each entity's static and dynamic features into account and controlled the contribution of these features to the vector of entities using a parameter. By dynamic modeling of entities and employing previous methods, DE-TransE and DE-DistMult were introduced [10]. The main difference between the mentioned methods and TransE and DistMult was in the way of modeling the entities.

## III. Proposed Method

The lack of novel methods considering the context information and paths in dynamic knowledge graph completion models, as well as the lack of considering the changes of knowledge graphs over time in static knowledge graph completion models, motivates us to propose a new method to complete dynamic knowledge graphs by incorporating time and using static methods. Fig. 1 indicates different components of our model described in this section.

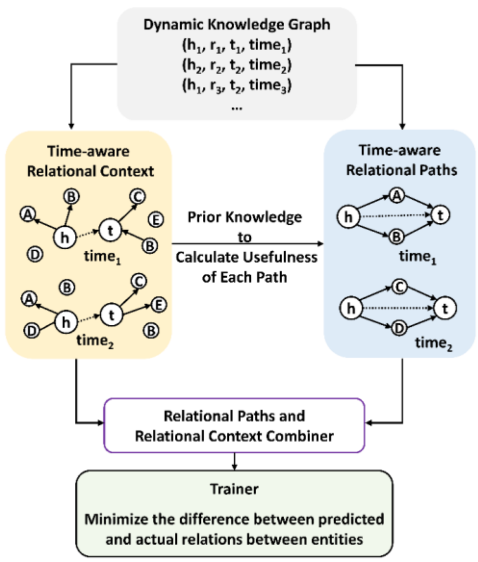


Fig. 1. Components of the proposed model.

Our model receives a set of facts constituting the dynamic knowledge graph as input. Inspired by [12], two components of relational context and relational paths are engaged. The main difference between ours and [12] is incorporating time in the relational context and relational paths modeling. A component is then needed to cope with combining information obtained from the entities' relational context and paths. Once the context and path information are combined, the model is ready for training. Further after training, the model can be applied to completing dynamic knowledge graphs. In the rest, first, the problem is defined. Next, each component in our proposed model is fully described.

### A. Problem definition

Predicting a new relationship between two entities at a specific time is the problem we encounter in completing dynamic knowledge graphs. In other words, the purpose of the model is to predict a relationship given head and tail entities and time.

Reminding the approaches for considering time, mentioned in Section II, time in the former is viewed as an independent factor, whereas it is incorporated in modeling entities and relations in the latter. Since assuming the independence of time makes implementation complicated and requires a separate vector, the latter approach is pursued in this paper. Therefore, the goal is to predict the probability depicted in (1).

$$p(r_{time} \mid h_{time}, t_{time}) \propto p(h_{time}, t_{time} \mid r_{time}) \cdot p(r_{time}) \tag{1}$$

Considering the probability value $p(r_{time})$ is a constant value, for calculating the probability $p(r_{time} \mid h_{time}, t_{time})$, it is enough to compute the probability $p(h_{time}, t_{time} \mid r_{time})$. According to the Bayes theorem, this probability can be obtained by (2).

$$p(h_{time}, t_{time} \mid r_{time}) = \frac{1}{2}((p(h_{time} \mid r_{time}) \cdot p(t_{time} \mid h_{time}, r_{time}) + p(t_{time} \mid r_{time}) \cdot p(h_{time} \mid t_{time}, r_{time})) \tag{2}$$

If the neighboring relations of the entities represented by $C(.)$, *called a local relational subgraph*, are considered as the context information of each entity, $p(h_{time} \mid r_{time})$ and $p(t_{time} \mid r_{time})$ will be respectively turned into $p(C(h_{time}) \mid r_{time})$ and $p(C(t_{time}) \mid r_{time})$. Moreover, $p(h_{time} \mid t_{time}, r_{time})$ and $p(t_{time} \mid h_{time}, r_{time})$ show the probability of reaching from one entity to another, given relation $r_{time}$. Next, Time-aware relational context and Time-aware relational paths are explained. Notations used here are listed in Table I.

TABLE I. NOTATION USED IN THIS PAPER

| Symbol | Description |
|---|---|
| $h_{time}, t_{time}$ | head and tail entities at a specific time |
| $r_{time}$ | relation at a specific time |
| $s^i_{e(time)}$ | hidden state of edge *e* at a specific time at *i-th* iteration |
| $m^i_{v(time)}$ | the message of node *v* at a specific time at *i*-th iteration |
| $\mathcal{N}_{time}(e)$ | endpoint nodes of edge *e* at a specific time |
| $\mathcal{N}_{time}(v)$ | incoming and outgoing edges to node *v* at a specific time |
| $S_{(h,t),time}$ | a representation for the context of entity *h* and the context of entity *t* at a specific time |
| $S_{h\to t,time}$ | a representation for all paths from *h* to *t* at a specific time |
| $\alpha_{P(time)}$ | attention weight of path *P* at a specific time |
| $\mathcal{P}_{h\to t,time}$ | a set that includes all paths from *h* to *t* at a specific time |

### B. Time-aware relational context

In order to obtain time-aware entities' context information, the relational message passing method is utilized. In this regard, a hidden state vector should be considered for each relation around the entity at a specific time. This vector is primarily a one-hot vector representing the type of relation.

According to (3), the messages of each entity at a specific time are obtained from integrating the messages of related relations. Equation (4) shows how the hidden state of relations is updated. For each entity, relations available within *K* order distance from the entity are considered as its neighbor relations. Afterward, the entities' context information is obtained by repeating (3) and (4) for *K* times; hence, information is available through the vectors $m^{K-1}_{h(time)}$ and $m^{K-1}_{t(time)}$.

$$m^i_{v(time)} = \sum_{e(time)\in\mathcal{N}_{time}(v)} s^i_{e(time)} \tag{3}$$

$$s^{i+1}_{e(time)} = \sigma\left(\left[m^i_{v(time)}, m^i_{u(time)}, s^i_{e(time)}\right] \cdot W^i + b^i\right),\quad v(time), u(time) \in \mathcal{N}_{time}(v) \tag{4}$$

### C. Time-aware relational paths

This section discusses the method of modeling paths and the time effect. It is worth noticing that there are some differences in path modeling between our proposed model and [12]. To clarify, suppose the knowledge graph is such that person (i.e., entity) A is an accountant in company C. Person B is also the CEO of company C. A cooperative relationship between A and B can be concluded if the examined knowledge graph is static and does not change over time. In contrast, the cooperative relationship cannot be deduced from a dynamic knowledge graph changing over time. Indeed, the existence of a cooperation relationship between A and B can only be concluded when the relationships "*being an accountant*" and "*being a CEO*" both exist at the same time.

To model a path between entities *h* and *t* at a specific time, consider $v_{i(time)}$ as an intermediate node, $e_{i(time)}$ as the edge connecting node $v_{i(time)}$ to $v_{i+1(time)}$, and $h(v_{0(time)}) \xrightarrow{e_{0(time)}} v_{1(time)} \xrightarrow{e_{1(time)}} v_{2(time)} \cdots v_{L-1(time)} \xrightarrow{e_{L-1(time)}} t(v_{L(time)})$ as the connection between nodes. So, the paths between the entities are presented as $P = (r_{e0(time)}, r_{e1(time)}, r_{e2(time)}, \dots, r_{eL-1(time)})$. If all paths between entities *h* and *t* are in $\mathcal{P}_{h\to t,time}$, a vector $S_{P(time)}$ is then regarded for each path $P_{time}$ in $\mathcal{P}_{h\to t,time}$. Note the $S_{P(time)}$ vector is obtained by integrating the relation vectors of the path $P_{time}$.

Once entities' context and paths information is available, the corresponding components should be combined to create a model for completing dynamic knowledge graphs, described in the following section.

### D. Time-aware relational context and relational paths combiner

Using two vectors, $m^{K-1}_{h(time)}$ and $m^{K-1}_{t(time)}$, mentioned in Section *B*, the vector $s_{(h,t),time}$ represents the time-aware context information of head and tail entities, as formulated by (5).

$$s_{(h,t),time} = \sigma\left(\left[m^{K-1}_{h(time)}, m^{K-1}_{t(time)}\right] \cdot W^{K-1} + b^{K-1}\right) \tag{5}$$

where $\sigma(.)$ is a nonlinear activation function, $W$ is a learnable weighting matrix, and *b* is a bias.

Our model declares the effect of different paths between two entities. According to (6), the entities' context information is used as prior knowledge to measure the importance of each path. Therefore, the entities' context information plays two crucial roles. The first is the direct impact on the proposed model, and the second is to use this information as prior knowledge to determine the importance of each path.

$$\alpha_{P(time)} = \frac{\exp\left(s_{P(time)}^{\top} s_{(h,t),time}\right)}{\sum_{P_{time} \in P_{h\to t,time}} \exp(s_{P(time)}^{\top} s_{(h,t),time})} \quad (6)$$

After calculating the weight for each path, information of all paths between entities $h$ and $t$ at a specific time is integrated using (7), which provides a vector representation for the paths information between two entities.

$$s_{h\to t,time} = \sum_{P_{time} \in P_{h\to t,time}} \alpha_{P(time)} s_{P(time)}^{i} \quad (7)$$

Given the entities' context and paths information, already determined based on time, the probability $p(h_{time}, t_{time} \mid r_{time})$ is computed by (8).

$$p(h_{time}, t_{time} \mid r_{time}) = SoftMax(s_{(h,t),time} + s_{h\to t,time}) \quad (8)$$

By creating the model, the difference between the predicted and the actual relation at a specific time should be minimized during the training, as (9) computes.

$$min\mathcal{L} = \sum_{(h,r,t,time) \in \mathcal{D}_{train}} J(p(r_{time} \mid h_{time}, t_{time}), r_{time}) \quad (9)$$

where $J(.)$ is the cross-entropy loss function and $\mathcal{D}_{train}$ is the training dataset. After training, the model is capable of being used as a new time-aware method for completing dynamic knowledge graphs.

## IV. Experimental Study

The key research questions to be addressed in this paper are as follows:

**RQ1:** In determining the entity of the other side of a relation, how well does the proposed model perform, given one entity and a relation?

**RQ2:** Given the model and several activation functions, which one performs the best?

This section continues by explaining datasets, configuration, and metrics, and answering the research questions.

TABLE II. Statistical report on ICEWS14 and ICEWS05-15

| Dataset | Number of Trainset Facts | Number of Evaluation-Set Facts | Number of Test-Set Facts | Sum of Facts | Number of Entities | Number of Relations | Number of Times |
|---|---|---|---|---|---|---|---|
| **ICEWS14** | 72826 | 8941 | 8963 | 90730 | 7128 | 230 | 365 |
| **ICEWS05-15** | 386962 | 46275 | 46092 | 479329 | 10488 | 251 | 4017 |

### A. Data

In this paper, two publicly available datasets, ICEWS14 and ICEWS05-15 produced by [23], are used as standard datasets for evaluating methods of completing dynamic knowledge graphs. ICEWS14 is a collection of facts in 2014, and ICEWS05-15 is a collection of facts between 2005 and 2015[1]. Table II summarizes some statistics of the datasets.

### B. Configuration and parameters

The experiments are run on an 11th Gen Intel Core i7-11800H at 2.30GHz CPU and 16 GB RAM, and one node with two GPUs. All algorithms are implemented in Python and by using the PyTorch framework. Table III reports the parameter values used for the simulations corresponding to each dataset.

TABLE III. The values of setting parameters

| Parameter | Value |
|---|---|
| learning rate | 5e-3 |
| batch size | 32[a], 16[b] |
| embedding size | 64 |
| maximum length of path | 4 |
| maximum length for collecting message from neighboring edges | 3 |

[a] ICEWS14 dataset, [b] ICEWS05-15 dataset

### C. Evaluation metrics

This paper follows a scenario to evaluate the proposed model so that two queries are created for each fact. The first query, $f = (?, r, t)$, means which entities are related to entity $t$ through the relation $r$. The proposed model determines several candidate entities and ranks them in response to this query. The second query, $f = (h, r, ?)$, also means which entities are related to $h$ by relation $r$. Let $k_{f,h}$ denote the rank of the head entity in the first query and $k_{f,t}$ be the rank of the tail entity in the second query. Accordingly, Mean Reciprocal Rank (MRR) and Hit@k are defined respectively by (10) and (11).

$$MRR = \frac{1}{2 * \mid \mathcal{D}_{test} \mid} \sum_{f=(h,r,t,\mathrm{time}) \in \mathcal{D}_{test}} \left(\frac{1}{k_{f,h}} + \frac{1}{k_{f,t}}\right) \quad (10)$$

where for all facts in the test dataset, $\mathcal{D}_{test}$, two queries are created, and the rank of the correct answer is examined among the candidate entities; thus, MRR measures the model's ability to determine the correct entity.

$$Hit@k = \frac{1}{2*|\mathcal{D}_{test}|} \sum_{f=(h,r,t,\mathrm{time}) \in \mathcal{D}_{test}} \left(1_{k_{f,h} \le k} + 1_{k_{f,t} \le k}\right) \quad (11)$$

where $1_{condition}$ equals one if the condition is satisfied; otherwise, it equals zero. Hence, Hit@k checks the presence of the correct entity in the first $k$ ranks.

### D. Comparison of the proposed method with counterparts

In response to RQ1, we study the performance of our model in terms of MRR and Hit@k for $k \in < 1, 3, 10 >$ in comparison to its counterparts. Here, we consider the effect of context, path, and both in the proposed model. Results reported in Table IV indicate the superiority of our model over the baselines in determining the correct entity.

[1] https://github.com/BorealisAI/DE-SimplE

TABLE IV. THE PERFORMANCE COMPARISON OF DIFFERENT MODELS. BEST RESULTS ARE IN BOLD.

| Model | MRR | Hit@1 | Hit@3 | Hit@10 |
|---|---|---|---|---|
| **TransE [14]** | 0.280* | 9.4* | - | 63.7* |
| | 0.294✦ | 9.0✦ | - | 66.3✦ |
| **DistMult [20]** | 0.439* | 32.3* | - | 67.2* |
| | 0.456✦ | 33.7✦ | -[a] | 69.1✦ |
| **DE-TransE [10]** | 0.326* | 12.4* | 46.7* | 68.6* |
| | 0.314✦ | 10.8✦ | 45.3✦ | 68.5✦ |
| **DE-DistMult [10]** | 0.501* | 39.2* | 56.9* | 70.8* |
| | 0.484✦ | 36.6✦ | 54.6✦ | 71.8✦ |
| **PathConTKG (path)** | 0.552* | 40.77* | 62.73* | 85.40* |
| | 0.641✦ | 55.42✦ | 67.16✦ | 83.64✦ |
| **PathConTKG (context)** | 0.546* | 40.21* | 61.31* | 84.59* |
| | 0.584✦ | 43.60✦ | 67.02✦ | 87.45✦ |
| **PathConTKG (path-context)** | **0.638*** | **52.47*** | **70.37*** | **86.47*** |
| | **0.708✦** | **62.25✦** | **75.94✦** | **88.86✦** |

*. ICEWS14 dataset, ✦. ICEWS05-15 dataset

a. The values for baselines are directly taken from [10]. Hit@3 values for TransE and DistMult were also not reported in [10].

### E. *Comparison of different activation functions*

Inspired by [10], this section addresses RQ2 by investigating different activation functions in the proposed model. Table V shows the experimental results on the ICEWS14 dataset considering both context and path. Accordingly, we apply the sigmoid function as it performs best in our experiments.

TABLE V. RESULTS ON THE ICEWS14 DATASET BASED ON DIFFERENT ACTIVATION FUNCTIONS. BEST RESULTS ARE IN BOLD.

| Model | Activation Function | MRR | Hit@1 | Hit@3 | Hit@10 |
|---|---|---|---|---|---|
| **DE-DistMult [10]** | Tanh | 0.486 | 37.5 | 54.7 | 70.1 |
| | Sigmoid | 0.484 | 37.0 | 54.6 | 70.6 |
| | Leaky ReLU | 0.478 | 36.3 | 54.2 | 70.1 |
| **PathConTKG** | Tanh | 0.617 | 50.75 | 67.82 | 83.94 |
| | **Sigmoid** | **0.638** | **52.47** | **70.37** | **86.47** |
| | Leaky ReLU | 0.544 | 46.98 | 60.46 | 66.12 |

## V. CONCLUSION

Due to the lack of methods that examine the context information and paths in dynamic knowledge graph completion models, and also modeling time in static approaches, this paper has introduced a model for dynamic knowledge graph completion leveraging entities' context and path information. The time factor has also been modeled while combining the relational paths and context. Experimental results on standard datasets, ICEWS14 and ICEWS05-15, have shown the superiority of our proposed model over the well-known baselines in terms of MRR and Hit@k. Note that the entities of the paths have not played a role in modeling the path. The neighboring entities also have not been taken into account in modeling the entities' context information. In future work, we will consider the effect of modeling entities in paths and neighboring entities for modeling context information.

## REFERENCES


[1] H. Wang, H. Ren, and J. Leskovec, "Entity context and relational paths for knowledge graph completion," arXiv preprint arXiv:2002.06757, p. 47, 2020.

[2] X. Zou, "A survey on application of knowledge graph," in Journal of Physics: Conference Series, 2020, vol. 1487, no. 1: IOP Publishing, p. 012016.

[3] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, "The design and implementation of xiaoice, an empathetic social chatbot," Computational Linguistics, vol. 46, no. 1, pp. 53-93, 2020.

[4] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," in Proceedings of the 2018 world wide web conference, 2018, pp. 1835-1844.

[5] L. Shi, S. Li, X. Yang, J. Qi, G. Pan, and B. Zhou, "Semantic health knowledge graph: semantic integration of heterogeneous medical knowledge and services," BioMed research international, vol. 2017, 2017.

[6] P. Chen, Y. Lu, V. W. Zheng, X. Chen, and B. Yang, "Knowedu: A system to construct knowledge graph for education," Ieee Access, vol. 6, pp. 31553-31563, 2018.

[7] J. Liu, Z. Lu, and W. Du, "Combining enterprise knowledge graph and news sentiment analysis for stock price prediction," 2019.

[8] D. Moussallem, M. Wauer, and A.-C. N. Ngomo, "Machine translation using semantic web technologies: A survey," Journal of Web Semantics, vol. 51, pp. 1-19, 2018.

[9] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," arXiv preprint arXiv:1611.03954, 2016.

[10] R. Goel, S. M. Kazemi, M. Brubaker, and P. Poupart, "Diachronic embedding for temporal knowledge graph completion," in Proceedings of the AAAI Conference on Artificial Intelligence, 2020, vol. 34, no. 04, pp. 3988-3995.

[11] Q. Wang, P. Huang, H. Wang, S. Dai, W. Jiang, J. Liu, Y. Lyu, Y. Zhu, and H. Wu, "Coke: Contextualized knowledge graph embedding," arXiv preprint arXiv:1911.02168, 2019.

[12] H. Wang, H. Ren, and J. Leskovec, "Relational message passing for knowledge graph completion," in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1697-1707.

[13] X. Wang, D. Wang, C. Xu, X. He, Y. Cao, and T.-S. Chua, "Explainable reasoning over knowledge graphs for recommendation," in Proceedings of the AAAI conference on artificial intelligence, 2019, vol. 33, no. 01, pp. 5329-5336.

[14] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," Advances in neural information processing systems, vol. 26, 2013.

[15] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in Proceedings of the AAAI conference on artificial intelligence, 2014, vol. 28, no. 1.

[16] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in Twenty-ninth AAAI conference on artificial intelligence, 2015.

[17] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers), 2015, pp. 687-696.

[18] M. Fan, Q. Zhou, E. Chang, and F. Zheng, "Transition-based knowledge graph embedding with relational mapping properties," in Proceedings of the 28th Pacific Asia conference on language, information and computing, 2014, pp. 328-337.

[19] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in Icml, 2011.

[20] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," arXiv preprint arXiv:1412.6575, 2014.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[22] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, "Representation learning for dynamic graphs: A survey," J. Mach. Learn. Res., vol. 21, no. 70, pp. 1-73, 2020.

[23] A. García-Durán, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," arXiv preprint arXiv:1809.03202, 2018.