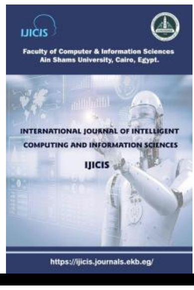
IJICIS, Vol.25, No.2, 88-104
DOI: 10.21608/ijicis.2025.398811.1407

International Journal of Intelligent Computing and Information Sciences

https://ijicis.journals.ekb.eg/



JOINT FEATURES-BASED KNOWLEDGE GRAPH COMPLETION

Maha Farghaly*

Information Systems Department, Faculty of Computers and Information Technology, Future University in Egypt, Cairo, Egypt
maha.farghaly@fue.edu.eg

Mostafa Aref

Computer Science Departement, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt
mostafa.aref@cis.asu.edu.eg

Mahmoud Mounir

Information Systems Departement, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt
mahmoud.mounir@cis.asu.edu.eg

Received 2025-06-28; Revised 2025-06-28; Accepted 2025-07-19

Abstract: *Knowledge graphs (KGs) help in resolving data inconsistencies and redundancies by organizing information in a unified structure, paving the way for building scalable, interpretable AI systems, as they provide a transparent way to trace reasoning paths and conclusions. The continuous evolving nature of information imposes for continuous knowledge graph completion (KGC). Thus, many research efforts are directed into building efficient KGC models. KGC models varied in the utilized knowledge features into the knowledge textual semantics, the graph relational structure, and temporal features, proving the crucial role of each. However, prior models neglect utilizing one or more types of features in the embedding and completion learning, missing the potential advantage of the neglected features for efficient completion learning. Thus, we propose an innovative sequential hybridization-based completion model, named Joint Features-based Knowledge Graph Completion (JF-KGC). The model encodes the latent semantics of knowledge, the structural topology, and the evolution of knowledge over time. This is a novel completion model that utilizes hybrid embedding of deep textual-based semantics, graph structure and temporal features for completion learning. The experiment conducted over YAGO11k and Wikidata12k benchmark datasets proved significant improvement over baselines in predicting missing knowledge with Mean Reciprocal Rank (MRR) equals 39.2% and 51.3% on YAGO11k and Wikidata12k, respectively. Besides, we provide an ablation study to prove and quantify the contribution of each feature type to the performance.*

Keywords: *Knowledge Graph Completion (KGC), Temporal Knowledge Graph (TKG) , Knowledge Semantics, Graph Topology, Language Model (LM).*

1. Introduction

***Corresponding Author**: Maha Farghaly

Information Systems Department, Faculty of Computers and Information Technology, Future University in Egypt, Cairo, Egypt

Email address: maha.farghaly@fue.edu.eg

Recently, Knowledge Graphs (KGs) have become fundamental for various knowledge-based AI systems such as chatbots [1], recommendation systems [2], and search engines [3]. This role is granted due to KGs' resolvability of inconsistencies and the unstructured nature of information in several domains [4]. Knowledge Graph (KG) consists of interconnected knowledge triplets, each of which consists of a relation connecting two entities [5]. The uniform integrity of real-world facts from several data sources is not the only capability of KGs, a KG is capable of holding much knowledge such as entities' attributes, entities' semantic definitions [6], and timestamps that can indicate the time validity of relations in a KG [7]. Recently, Temporal Knowledge Graphs (TKGs) [8] and Dynamic Knowledge Graphs (DKGs) [9] appeared and gained huge attention in the research community. A Temporal knowledge graph (TKG) is a KG in which eventual facts have timestamps and the topological structure of the KG dynamically changes over time [10]. The aforementioned capabilities paved the way for KG to be adopted for many knowledge-based applications in different domains such as medical [11, 12], educational [13, 14], commercial [15, 16], and enterprise domain [17, 18]. Despite the huge capabilities of KGs and their prevalence, the research community believed in the incompleteness of constructed KGs and how it holds back the capabilities of the relying application. Therefore, various efforts are directed to develop methods for exploring and discovering missing facts in a KG [19]. Knowledge Graph Completion (KGC) is the process of reasoning missing paths in a KG to achieve KG completeness by utilizing graph information for completion learning [20]. The prior KGC models varied with respect to the utilized graph information which can be classified into graph-structural features, textual description features, and temporal features [21].

Textual description features can be defined as the derived context from entities and relations co-occurrences or the hidden semantics in the natural language sentences of the definitions, and descriptions of entities and relations [22]. Textual-based completion methods, (e.g., KG-BERT [23]) transform the completion task into a Natural Language Processing (NLP) task in which a pre-trained Language Model (LM), or NLP techniques are applied for completion. Graph-structural features related to the topology and connectivity structure of knowledge triplets and the relations between entity pairs in the graph [24]. Structural-based completion methods (e.g., RotatE [25]) leverage the structural features to learn determine the structural trustworthiness of a candidate knowledge triplet, in order to complete the graph. Temporal features refer to the timestamps of relations in knowledge triplets, indicating the specific time in which an eventual fact occurred [26]. Temporal-based completion methods, (e.g., TELS [27]) employ the timestamps of knowledge triplets to learn the graph evolution over time and hence, detect missing knowledge. Although the results in the literature proved the crucial role of each of the aforementioned knowledge features for KGC, prior conventional models have a major drawback which is ignoring one or more types of features in the learning process, missing the neglected features' potential capability of discovering missing knowledge. Therefore, the incorporation of the three types of features to train one completion model is vital to reach the completeness of the KG efficiently. The combination of textual, structural, and temporal features to jointly train a completion model is challenging due to the different modalities of the features, and direct concatenation might not effectively capture interactions between features, which requires an adaptive fusion approach.

In this work, we propose Joint Features-based Knowledge Graph Completion (JF-KGC), which follows a proposed sequential hybridization approach to learn KGC with semantic, structural and temporal features of TKG. The model splits KG into sequential snapshots of sub-KGs based on the timestamps, then obtains a semantic-aware representation of knowledge triplets in each snapshot using Bidirectional Encoder Representations from Transformers (BERT). Subsequently, the encoded semantics of entities and relations are refined with neighboring structural features on each temporal snapshot using a Relational Graph Convolutional Network (R-GCN). Finally, a Gated Recurrent Unit (GRU) captures the evolution of knowledge and the sequential patterns through the consecutive snapshots' representations, to attain final representations of knowledge that hold semantic, structural and temporal features, to learn completing the KG.

The incorporation of features in the proposed model provides an enhanced representation of real-world data, and a generalization in completion learning across different graph types with the ability to fill the gap of insufficiency of a specific knowledge feature with other features in the graph.
In this work, we intend to prove the necessity of the incorporation of features. Our contribution can be summarized as follows:

- A review of the related baseline KGC models, focusing on novel perspective, which is employed and missed knowledge features in the training process of the models one by one.
- As far as we know, our model achieves novelty in KGC by hybridizing the completion learning with the three types of knowledge features.
- The novelty of the methodological pipeline which leverage KG's semantics guided by contextual pre-trained LM, jointly with the topologies in the graph and the temporal evolution of knowledge for KG completion.
- Model evaluation on YAGO11k and Wikidata12k datasets and results analysis against baseline KGC models proved a significant improvement achieved in KGC.
- Ablation studies are conducted to prove the necessity and quantify the contribution of each type of knowledge feature to the completion of TKG.

The paper is organized into: Section 2 exhibits the KGC process and investigates the recent KGC models related to our work along with their drawbacks with respect to the utilized knowledge features, emphasizing the difference and clarifying the position of our work. In Section 3, we elaborate JF-KGC model phase by phase. In Section 4, we pose our experimental details and the evaluation results of the JF-KGC model on benchmark datasets along with an extensive analysis of JF-KGC. In Section 5, remarks and insights are concluded and followed by suggested future work.

2. KGC Background and Related Models Review

2.1. KGC Background

KGs popularized lately [28] and many efforts directed for KG construction and completion [29] in various domains [30]. The incompleteness in KGs happens due to inaccurate or incomplete data sources, or inefficiency of the KG construction model. To predict missing knowledge, candidate facts need to be generated from $\mathcal{E}$, and $\mathcal{R}$; the entities and relations existing in the graph, to be scored further by the KGC model, whether they are correct missing facts or not. For KGC model construction, a preprocessing step over the original KG needs to be done. Negative samples $\mathcal{F}'$ generated out from the existing set of facts $\mathcal{F}$, in the KG, by corrupting a number of facts randomly. Then, both $\mathcal{F}$ and $\mathcal{F}'$ combined to be used in the construction of the KGC model. Subsequently, the KG is embedded to obtain representation vectors that take into consideration the knowledge features. The features-aware embedded vectors are employed to train a factual scoring function capable of discriminating against the true facts from the false ones out of the candidate facts. In the literature, KGC models varied in the considered knowledge features in KG embedding. Following, we are going to exhibit the related KGC models according to the levered feature variations, along with the benchmark datasets employed for model evaluation.

2.2. Related KGC Models

In this sub-section, KGC models' variations in feature types, utilization strategy, and evaluation benchmarks are discussed. For semantics leveraging, authors of [23] employed pre-trained LM to learn the textual semantics of knowledge triplet by processing the definitions of the triplet's components as a textual sequence. The textual sequences of each knowledge triplet fed into the pre-trained LM to train the loss function to learn triplet plausibility. Similarly, LASS model [31] employed pre-trained LM to capture the latent semantics in each knowledge triplet, for each component in isolation, to obtain a single embedding vector for each entity in the triplet and for the relation. The authors proposed a scoring function based on translation property to model triplet components. However, the topological features and temporal dynamicity of facts were neglected in [23] and [31].

Meanwhile, the authors of RotatE [25] capture only the triplet structural features by projecting KG entities into a vector space and KG relations as a rotation in the same space. In [32], the authors maintain the triplets' structure pattern while embedding the KG's entities and relations into a complex space to consider the relations hierarchy in training the completion model. Nevertheless, the last two aforementioned models utilized only graph structural topology.

For temporals leveraging, Hyte model [33] projects the entities and relations embeddings into a temporal hyperplane to capture the evolution through the temporal hyperplane. However, only the temporal feature is utilized. The same limitation exists in TTransE [34], where the translation property is considered in representation with respect to the time, to model the changes of facts representations at different time stamps. In [35], the TA-DistMult model advances DistMult by augmenting time-dependent representations for entities and relations. Temporal features were also utilized in ATiSE [36], by capturing the uncertainty and temporal variations using Gaussian distributions and time-aware attention. LGRe [37] employed a Convolutional Neural Network (CNN) to utilize temporals to learn the changing patterns of entities and relations through different time periods, considering time at periodical levels, not as a static stamp. The models [33] through [37] missed leveraging structural and textual semantic features.

In [38], both the structural and temporal features are utilized for learning KGC. The structural patterns were captured using R-GCN and the temporal features were utilized using time gate and GRU. The same features are utilized for KGC in ATGIE [39]. The model integrates Euclidean, hyperbolic, and hyperspherical geometric spaces to model the time-varying structures inherent in TKGs. R-GCN was also utilized in THOR [40] with three modifications to capture the topological patterns of relations, entities' temporal dependencies, and entities' non-temporal dependencies to train the completion model. In [27], TELS model captured the changing topology representation of entities over time, by employing dual quaternion algebra. This mathematical framework allows for the simultaneous modelling of rotational and translational transformations, capturing intricate temporal evolutions and latent relations between entities. However, the advantage of latent semantics in textual descriptions is unexploited in [38] through [27], losing its capability in the completion learning.

The textual-based and graph structural-based completion models employed for evaluation benchmark dataset(s) derived from WordNet (i.e. WN18, and WN18RR), Freebase (i.e. FB15k, and FB15k-237), Unified Medical Language System (UMLS), and YAGO (i.e. YAGO3-10, and YAGO11k). The temporal-based completion models employed for evaluation benchmark dataset(s) derived from YAGO (i.e. YAGO3-10, and YAGO11k), Wikidata, Integrated Crisis Early Warning System (ICEWS), and Global Database of Events, Language, and Tone (GDLET). The employed knowledge features in the reviewed KGC models and the origin of the employed datasets are encapsulated in Table 1.

2.3. Research Gap

The reviewed KGC models and the considered perspective reveal a research gap which is exploring the impact of cooperating the three types of features to train a completion model with hybridized embeddings, as the baseline models neglect utilizing one or more types of knowledge features which can be noticed clearly in Table 1, missing the potential advantage of the neglected feature over efficient KGC. This sets our proposed model apart from the existing KGC models and highlights our contribution.

3. JF-KGC Model

In general, TKG can be remodeled into a sequence of temporal consecutive KG snapshots, that is TKG= {KG1, KG2, …, KGt}. Each temporal snapshot mainly consists of a set of interrelated facts occurred at time t, which denoted as $\mathcal{F}_t$. $f_t \in \mathcal{F}_t$, is a fact occurred at time t, which consists of a knowledge triplet that can be denoted as (e_s, r, e_o), where e_s is the subject node, e_o is the object node, and r is the relation edge that connects the two entity nodes. e_s, $e_o \in \mathcal{E}$, and $r \in \mathcal{R}$, where $\mathcal{E}$ and $\mathcal{R}$ are the set of all entity nodes and relation edges in TKG, respectively.

The completion of a KG is concerned with predicting missing links in the graph. We propose JF-KGC model which aims to solve this problem by training the model with embedding vectors that capture the textual semantics, structural graph topology, and historical evolution in TKG. All symbols mentioned in the proposed model are summarized in Table 2, along with their definitions. JF-KGC model is composed of four main phases which are the semantic phase, the graph structure phase, the temporal phase, and the scoring phase, as illustrated in Figure. 1. Each phase is explained in a separate sub-section.

Table 1 The Employed Features in Related KGC Models and The Origin of The Employed Benchmarks for Evaluation

Model Ref.	Employed Feature			Employed Benchmark						
	Textual description features	Graph structural features	Graph temporal features	WordNet	Freebase	UMLS	YAGO	Wikidata	ICEWS	GDLET
[23]	•			•	•	•				
[31]	•			•	•	•				
[25]		•		•	•					
[32]		•		•	•		•			
[33]			•				•	•		
[34]			•				•	•	•	
[35]			•				•	•	•	
[36]			•				•	•	•	
[37]			•				•	•	•	
[38]		•	•				•	•	•	•
[39]		•	•				•		•	•
[40]		•	•				•		•	
[27]		•	•				•	•	•	

Table 2 Definitions of Symbols

Symbol	Definition
KGt	Snapshot of KG at time t
$\mathcal{F}_t$	Set of all facts occurred at time t
f_t	A temporal fact where $f_t \in \mathcal{F}_t$
e_s, e_o, r	Subject entity, object entity and a relation in a knowledge triplet, respectively
$\mathcal{E}$	The set of all entities in TKG
$\mathcal{R}$	The set of all relations in TKG
v_s, v_o	The subject and object semantic representation, respectively
v_r	The relation semantic representation
$\mathcal{E}'_t$	A matrix of the structural-refined semantic representation of all entities occurred at time t
$\mathcal{R}'_t$	A matrix of the structural-refined semantic representation of all relations occurred at time t
$\mathbf{E_i}, \mathbf{R_i}$	Evolved representation of ith entity and ith relation in TKG

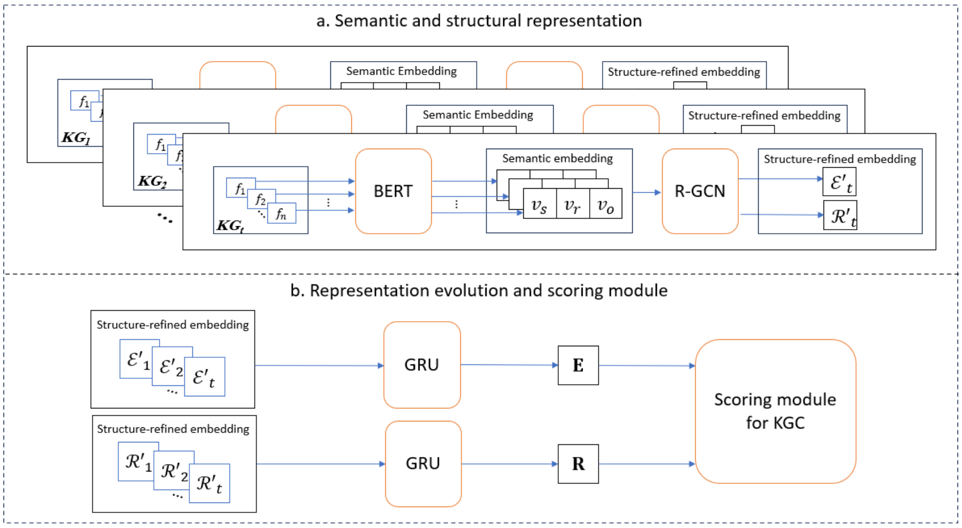


Figure. 1: An overview of JF-KGC, a. The semantic and structure-refined representations; b. The representation evolution

3.1. The Semantic Phase

The hidden semantics in textual descriptions of entities and relations proved to be significant for KGC process due to the semantic similarity among the components of a knowledge triplet [31]. Recently, pre-trained LMs played a vital role in many NLP tasks and employed in the context of KG [41]. The powerful capability of the pre-trained LMs in capturing language semantics from natural language text motivated us to exploit a pre-trained LM to capture the hidden semantics in textual descriptions of the components of knowledge triplets. In this phase, we employed for encoding the pre-trained LM, BERT. For each temporal knowledge triplet f_t in the TKG, the textual definition of its e_s, r, and e_o, acquired.

For each of the definitions, the word sequence is tokenized into a sequence of tokens, to be fed into BERT model, following the format in Figure. 2, where [CLS] is placed to mark the beginning of f_t sequence of tokens, and [SEP] is placed to separate the definition tokens of e_s, r, and e_o. Subsequently, for each token, the sum of three embeddings is obtained (i.e., token, segment, and position), and the final embedding produced through the pre-trained BERT. To obtain an aggregated embedding vector for each element in a knowledge triplet, mean pooling applied to its tokens embeddings which produced out from BERT. The semantic features of e_s, r, and e_o are finally captured in v_s, v_r, and v_o as defined in Eqs. (1-3), respectively:

$$v_s = Mean\,pooling(T_1^s, T_2^s, \dots, T_{ns}^s) \tag{1}$$

$$v_r = Mean\,pooling(T_1^r, T_2^r, \dots, T_{nr}^r) \tag{2}$$

$$v_o = Mean\,pooling(T_1^o, T_2^o, \dots, T_{no}^o) \tag{3}$$

where T is the final token embedding produced by BERT. v_s, v_r, and $v_o \in \mathbb{R}^d$, where d is the dimension of the produced vector. However, the semantic vectors hold the semantic features of each fact in isolation, without considering the structure topology of entities in the graph. Thus, a succeeding structural phase is needed in order to refine the encoded semantic embeddings with structural features.

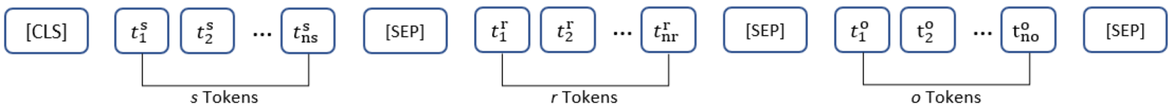


Figure. 2: Triplet formatting to be fed into BERT

3.2. The Graph Structure Phase

Relational structure features are crucial for TKG completion since the topologies of the graph evolves over time and the potential information in neighboring nodes [42]. Thus, we refine the semantic embeddings with structural features by utilizing R-GCN [43] with multi-layers, to aggregate information from neighboring nodes. At layer $l+1$, an object embedding is refined with the object embedding at layer l, and information from neighboring subjects' embeddings and relations' embeddings at layer l, through message passing formula, w.r.t. the translation property between subject and object, as defined in Eq. (4):

$$v_o{}^{l+1} = \sigma\left(\sum_{(e_s,r),\exists\,(e_s,r,e_o)\in\mathcal{F}_t} \frac{1}{c} W_1^l\,(v_s{}^l + v_r) + W_2^l v_o{}^l\right) \tag{4}$$

where $\sigma(\cdot)$ is ReLU activation function; c is a constant equals to the number of the connected relations to e_o; $v_o{}^l$, $v_s{}^l$, and v_r are the embeddings at layer l of object, subject and relation, respectively; W_1^l, W_2^l are the aggregated neighboring information and self-loop weight matrices, respectively. Note that the R-GCN's initial input embeddings are the pre-computed BERT embeddings; v_s, v_r, and v_o. The output of R-GCN at the last layer holds the semantic features of the entities refined with the topological structure of the entities' neighbors in the graph at time t. For all entities occurred at time t, the refined representation vectors grouped into a matrix, $\mathcal{E}'_t$, and the relation representation vectors into $\mathcal{R}'_t$.

However, the semantics and dependency patterns among each graph snapshot are encoded independently. Thus, a temporal phase needed to learn the evolved representations over the sequence of the snapshots.

3.3. The Temporal Phase

According to the real-world evolving nature, knowledge evolves over time where the relationships between entities may change over time [44]. This dynamic topology calls for learning the evolved characteristics of knowledge over time and aggregate information of the nodes and relations from their past states for KGC.

To obtain the evolved representation along the temporal sequence of snapshots, GRU [45] utilized. For each entity $e_i \in \mathcal{E}$, the evolved temporal representation $\mathbf{E_i}$ obtained as described in Eq. (5):

$$\mathbf{E_i} = GRU(\mathbf{E_i}^t, \mathbf{E_i}^{t-1}) \tag{5}$$

where $\mathbf{E_i}^{\mathrm{t}} \in \mathcal{E}'_t$, is the semantic-structural representation of e_i at t. The evolved representations of relations (i.e., $\mathbf{R}$) are also obtained using GRU.

3.4. The Scoring Phase

The concern was predicting missing links in TKG. In this sub-section, we define the score function employed in our model to score candidate entities, which supervises the training of the loss function. The score function for candidate entities to be an object in a knowledge triplet is Se, defined in Eq. (6):

$$Se = sigmoid(\mathbf{E}\, dec(v'_s, v'_r)) \tag{6}$$

where v'_s, v'_r are e_s and r representation in $\mathbf{E}$ and $\mathbf{R}$, respectively; $dec(\cdot)$ is ConvTransE [46] decoder which inspired by [38]; Se $\in \mathbb{R}^{\mathcal{E}\times d}$; and Se$_i \in [0,1]$. We computed the cross-entropy loss, supervised by Se for entity prediction in Eq. (7):

$$\mathcal{L}_e = \sum_{(e_s,r,e_o)\in\mathcal{F}_t} \sum_{i=1}^{|\mathcal{E}|} y_{(e_s,r,e_o),i} log(Se_i) + ((1 - y_{(e_s,r,e_o),i}) log(1 - Se_i)) \tag{7}$$

where $y_{(e_s,r,e_o),i}$ in Eq. (7) is the i$^{\mathrm{th}}$ entity label; having $y_{(e_s,r,e_o),i}$ equals 1 if the entity belongs to the triplet and equals 0 if not.

4. Experimental Details and Results Analysis

The evaluation of JF-KGC for KGC is conducted through deducing incomplete facts with missing entities. Subsequently, the missing part in the fact substituted with every entity in $\mathcal{E}$ to compute their scores using Eq. (6). In the following subsections, we pose the evaluation setup including employed datasets description, the metrics of evaluation, and the parameters setting details, the evaluation results along with a detailed analysis for the results compared to the baseline completion models, analysis of the model performance, and ablation studies for quantifying the contribution of each feature type.

4.1. Evaluation Setup

4.1.1. Dataset

The necessity of the encapsulation of semantic definitions, rich structures, and timestamps forces our experiment to be conducted upon YAGO11k and Wikidata12k datasets. Both datasets are commonly employed for evaluating graph structural-based and temporal-based completion models in literature, as can be clarified from Table 1. For the semantics of entities and relations, the definitions are acquired from Wikipedia summaries. YAGO11k is derived from the benchmark YAGO3 [47], with 10,623 entity, 10 relations, and 20,507 facts. Wikidata12k is derived from Wikidata [48] with 12,544 entity, 24 relation, and 40,621 facts. Both entities' names and definitions are considered. Only the year value is considered in the time stamps. For the facts with interval time stamps, only the start dates were considered. The datasets are divided into temporal snapshots of one year time interval.

Through the experiments, the datasets divided into 80%, 20% and 20% as shown in Table 3. The statistical description of the two datasets is also summarized in Table 3. The distributions of facts over time of YAGO11k and Wikidata12k shown in Figure. 3 reveal a long-tail problem in which most facts are distributed in fewer timestamps and most stamps contain fewer facts.

4.1.2. Evaluation Metrics

JF-KGC evaluated using the two popular evaluation metrics in KGC scope:

- Mean Reciprocal Rank (MRR): It is the average of the reciprocal rank of correct predicted fact among the set of corrupted candidates in the test set.
- Hits@k: It is the proportion of times in which the correct predicted fact appears within the k-top ranked predictions.

Table 3 A Summary of Datasets Description

Dataset	Triplets	Entities	Relations	Training	Validation	Testing
YAGO11k	20,507	10,623	10	16,406	2050	2051
Wikidata12k	40,621	12,554	24	32,497	4062	4062

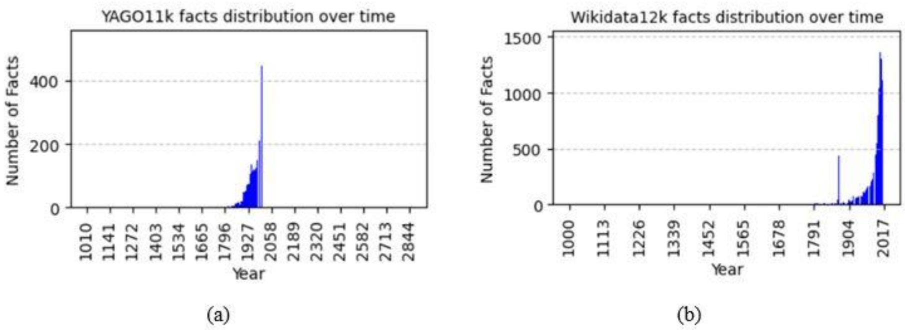


Figure. 3: Facts distribution over time in (a) YAGO11k , and (b) Wikidata12k based on facts begin

4.1.3. Parameters Setting

For semantic embedding using BERT; $\text{BERT}_{\text{BASE}}$ adopted; Sequence length=512; batch size=32; and d=768. For R-GCN, layers count=2; dropout rate=0.2. For ConvTransE, number of filters=50; and dropout rate=0.2. Adaptive Moment Estimation (Adam) optimizer employed. The learning rate is set to 0.001. The parameters were selected experimentally.

4.2. Evaluation Results and Comparative Discussion

We ran our model on YAGO11k and Wikidata12k for evaluation and assessment. The evaluation results presented in Table 4 in terms of MRR and Hits@k are the average results of predicting missing head entity and missing tail entity. The dashes indicate that the results are not reported and cannot be obtained. The **bold** values represent the best overall, and the underlined represent the runner up. We compare our model against the completion baselines that are oriented with link prediction task, with similar experimental setup:

(i) One feature-based KGC models, which are: RotatE [25], HyTE [33], TTransE [34], TA-DistMult [35], ATiSE [36], and LGRe [37], presented in Table 4.

(ii) Two feature-based KGC models, which are: ATGIE [39], THOR [40], and TELS [27], presented in Table 4.

All models are reviewed in Section 2. As shown in Table 4, JF-KGC accomplishes improvements over the baseline completion models for both datasets. In terms of MRR, JF-KGC outperforms the best model in (i) family by 13.4% on YAGO11k and 7.3% on Wikidata12k outperforms the best model in (ii) family by 2.8% and 2.7% on YAGO11k and Wikidata12k, respectively. Our model surpasses the average MRR of (i) family by 22.35%, and the average MRR of (ii) family by 11.43% on YAGO11k. It can be observed from Figure. 4 that (ii) family models, which utilize a combination of two types of features (i.e., either textual semantics and temporal features or graph topology and temporal features) in the embedding and completion learning outperforms with a margin (i) family models, in which only one feature type considered in the embedding and completion learning, in terms of MRR and Hits@1. However, JF-KGC which utilizes the three types of features, outperforms both families, on both datasets. Also, it can be clearly noticed in Figure. 4 the existence of positive correlation between the count of utilized knowledge features in completion process and the evaluation results, proving the crucial role of the utilization of the textual semantics, graph structural, and temporal features. Moreover, the results achieved on YAGO11k and Wikidata12k despite the long-tail effect shown in Figure. 3, prove the efficiency of hybrid representation by JF-KGC.

Table 4 Evaluation on YAGO11k and Wikidata12k in Terms of The Percentage of Hits@1, Hits@3, Hits@10, and MRR

Dataset	YAGO11k				Wikidata12k			
Metric	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
RotatE [25]	0.103	0.167	0.305	0.167	-	-	-	-
HyTE [33]	0.033	-	0.298	0.136	0.147	-	0.483	0.253
TTransE [34]	0.020	0.150	0.251	0.108	0.096	0.184	0.329	0.172
TA-DistMult [35]	0.098	-	0.267	0.155	0.130	-	0.461	0.230
ATiSE [36]	0.121	0.197	0.319	0.187	0.198	0.329	0.507	0.299
LGRe [37]	0.188	0.274	0.402	0.258	0.341	0.491	0.642	0.440
ATGIE [39]	0.155	0.232	0.357	0.223	-	-	-	-
THOR [40]	0.186	0.263	0.359	0.246	-	-	-	-
TELS [27]	0.280	0.404	0.525	0.364	0.370	0.543	0.674	0.486
JF-KGC	**0.346**	**0.417**	**0.564**	**0.392**	**0.462**	**0.588**	**0.693**	**0.513**

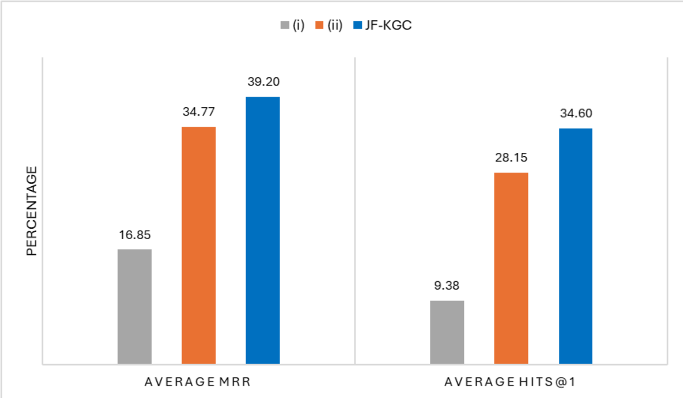


Figure. 4: The average MRR and Hits@1 (percentage) of (i) one feature-based KGC models, (ii) two feature-based KGC models, and JF-KGC on YAGO11k

4.3. Analysis

In this subsection, we provide deep analysis to evaluate JF-KGC performance. A comparative analysis of JF-KGC complexity is conducted and the generalization and robustness of JF-KGC is evaluated. Moreover, the sequential fusion strategy employed in JF-KGC is empirically analyzed.

4.3.1. Complexity Analysis

To further assess JF-KGC, a comparative analysis is conducted of the space complexity required to train JF-KGC against competitors. Table 5 shows the obtainable space computational complexity of KGC baselines and JF-KGC, where d is the embedding dimension, $n_{\mathcal{E}}$, $n_{\mathcal{R}}$, and n_t are the number of entities, relations and timestamps in the KG. Comparative analysis shows that while our model incorporates multiple feature types (i.e., semantic, structural, and temporal), space complexity remains competitive due to the sequential refinement strategy.

4.3.2. Generalization and Robustness Analysis

The evaluation of KGC model performance to unseen data is necessary to judge the generalizability of the proposed model to other graph types, while the evaluation of model performance against noise and sparsity is necessary to judge the model robustness. To evaluate the generalization and robustness of JF-KGC, YAGO11k is employed. For generalization evaluation, facts are transferred into two equal parts from the training set into testing and validation sets following [49]; to be unseen during the model training. The percentages of the randomly removed facts during the experiments are 10%, 20%, and 30%.

Table 5 Comparative Summary of JF-KGC Space Complexity Against Baseline Models

Model	Complexity
RotatE [25]	$O(d(n_{\mathcal{E}} + n_{\mathcal{R}}))$
HyTE [33]	$O(d(n_{\mathcal{E}} + n_{\mathcal{R}} + n_t))$
TTransE [34]	$O(d(n_{\mathcal{E}} + n_{\mathcal{R}} + n_t))$
TA-DistMult [35]	$O(d(n_{\mathcal{E}} + n_{\mathcal{R}} + n_t))$
ATiSE [36]	$O(d(n_{\mathcal{E}} + n_{\mathcal{R}}))$
LGRe [37]	$O(d(n_{\mathcal{E}} + n_{\mathcal{R}} + n_t))$

THOR [40]	$O(d(n_{\mathcal{E}}n_t + n_{\mathcal{R}}))$
JF-KGC	$O(d(n_{\mathcal{E}} + n_{\mathcal{R}} + n_t))$

For evaluation of robustness against sparsity, 25%, 50, and 75% of the facts are removed through the experiments to create sparsity, while the same proportion of facts are inserted to create noise for robustness against noise evaluations. The evaluation results of generalization and robustness are shown in Figure. 5 and Figure. 6, respectively. The model's performance gradually decreases, showing slight sensitivity to unseen facts, noise, and sparsity. However, JF-KGC maintains its superiority over KGC models in (i) family and most of the models in (ii) family even under extreme noise, sparsity and unseen facts. We attribute this to the ability of multi-feature fusion to compensate for the lack of information.

4.3.3. Fusion Analysis

In JF-KGC, we followed a sequential refinement-based strategy to fuse semantic, structure, and temporal features. To verify the efficiency of the applied fusion strategy, an experiment is conducted on YAGO11k and Wikidata12k to explore JF-KGC with simple concatenation fusion. The results are illustrated in Table 6, where JF-KGC$_{SC}$ stands for JF-KGC with simple concatenation for features fusion and JF-KGC$_{SR}$ stands for JF-KGC with sequential refinement for features fusion. As shown in Table 6, the JF-KGC$_{SR}$ surpasses JF-KGCSC with margin.

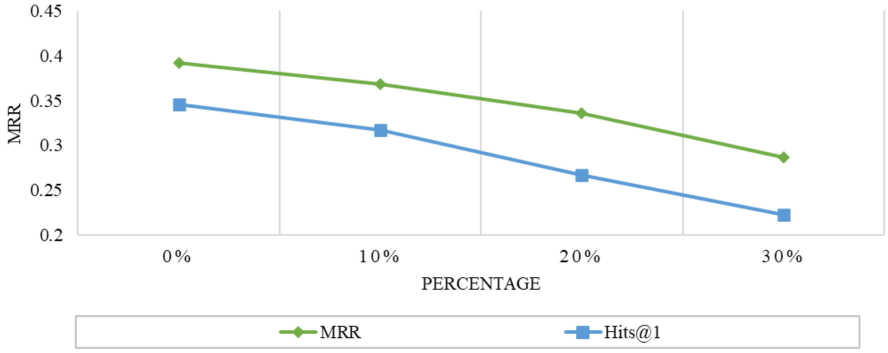


Figure. 5: The achieved MRR and Hits@1 across the evaluation of generalization on YAGO11k

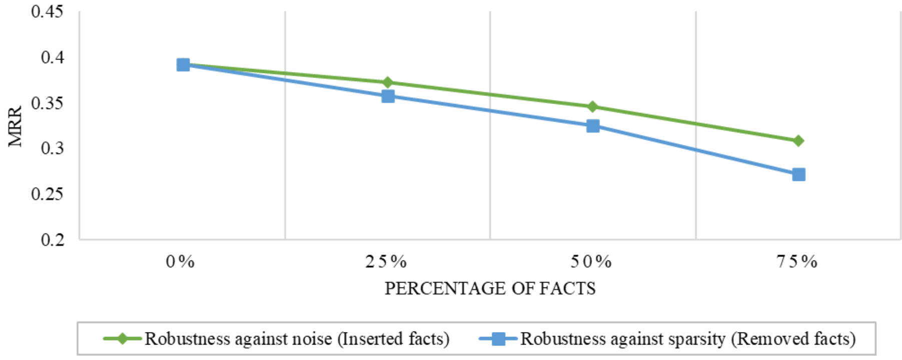


Figure. 6: The achieved MRR across the evaluation of robustness against noise and sparsity on YAGO11k

Table 6 Evaluation of Fusion Strategy on YAGO11k and Wikidata12k

Dataset	YAGO11k				Wikidata12k			
Metric	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR
JF-KGC$_{SC}$	0.198	0.235	0.269	0.203	0.244	0.333	0.462	0.324
JF-KGC$_{SR}$	0.346	0.417	0.564	0.392	0.462	0.588	0.693	0.513

Table 7 Ablation Studies on YAGO11k Benchmark

Variation	Hits@1	Hits@3	Hits@10	MRR
w/o-Descriptions	0.237	0.296	0.364	0.289
w/o- Structures	0.187	0.227	0.293	0.244
w/o- Temporals	0.213	0.248	0.322	0.263
JF-KGC	0.346	0.417	0.564	0.392

4.4. Ablation Study

In order to demonstrate and quantify the impact of each feature type to the model's performance, we conducted ablation studies on YAGO11k dataset. The ablation studies involve eliminating each of the features discussed in Section 1, at once. We demonstrate the elimination results in Table 7 and discuss the contribution of each.

4.4.1. Elimination of Textual Description Features

The textual semantics were captured from the description sentences of entities and relations using pre-trained BERT to be fed further for structural refinement. Understanding the language semantics and contextual meaning were beneficial to predict missing knowledge, achieving improvements over the results by 10.3% in terms of MRR, 10.9% Hits@1, 12.1% Hits@3, and 20.0% Hits@10.

4.4.2. Elimination of Graph Structural Features

The graph topology of the occurring facts at each time stamp captured using R-GCN, which was initialized with the semantic embedding of entities and relations to be refined with neighboring information. Understanding the connectivity patterns of facts at each time stamp improved the results by 14.8%, 15.9%, 19.0%, and 27.1% in terms of MRR, Hits@1, Hits@3, and Hits@10, respectively.

4.4.3. Elimination of Temporal Features

The evolution of knowledge representation over time captured using GRU by feeding GRU with the sequential historical representation of knowledge through the historical occurrences over time. Considering the temporal variation in knowledge representation improved the results by 12.9% in terms of MRR, 13.3% Hits@1, 16.9% Hits@3, and 24.2% Hits@10.

5. Conclusion

Joint Features-based Knowledge Graph Completion (JF-KGC) model is novel proposed. The model acquires the definitions of entities and relations to encode their semantics, on each temporal snapshot in a TKG, using a pre-trained LM (i.e. BERT). For each snapshot, the semantic embeddings fed into multilayer R-GCN to refine the semantic embeddings with potential structural embeddings. Finally, the refined embeddings on each snapshot fed into GRU to obtain the evolved representations over the snapshots. ConvTransE decoder employed to compute the score of candidates and the computed score supervises a cross-entropy loss for link prediction. Experiments were conducted over two temporal benchmark KGs, YAGO11k and Wikidata12k. The acquired semantic definitions gathered from Wikipedia. The model achieved MRR equals to 39.2%, Hits@1 equals to 34.6%, Hits@3 equals to 41.7%, and Hits@10 equals to 56.4% on YAGO11k dataset. The model achieved MRR equals to 51.3%, Hits@1 equals to 46.2%, Hits@3 equals to 58.8%, and Hits@10 equals to 69.3% on Wikidata12k dataset. The experiments show that JF-KGC outperforms the baseline models, proving the crucial rule of leveraging textual semantics, topological, and temporal features of KGs. In future, the outcome of other techniques and methods in the followed sequential hybridization approach over the performance of the model are yet to be investigated, as well as other features' hybridization approaches such as weighted embeddings' combination, and parallel training-based approaches. We plan to investigate the possibility and scenarios of domination or redundancy of specific features. We aspire to handle non-eventual facts and to advance the handling of facts with temporal intervals. Moreover, further investigation of the model performance under long-tail and sparse relations are yet to be carried out.

References

1. P. Ni, R. Okhrati, S. Guan and V. Chang, "Knowledge graph and deep learning-based text-to-GraphQL model for intelligent medical consultation chatbot," Information Systems Frontiers, vol. 26, no. 1, pp. 137-156, 2024.
2. J.-C. Zhang, A. M. Zain, K.-Q. Zhou, X. Chen and R.-M. Zhang, "A review of recommender systems based on knowledge graph embedding," Expert Systems with Applications, 2024.
3. N. Kokash, M. Romanello, E. Suyver and G. Colavizza, "From books to knowledge graphs," Journal of Data Mining & Digital Humanities, 2023.
4. E. Rajabi and K. Etminani, "Knowledge-graph-based explainable AI: A systematic review," Journal of information science, vol. 50, no. 4, pp. 1019-1029, 2024.
5. Y. Chen, X. Ge, S. Yang, L. Hu, J. Li and J. Zhang, "A survey on multimodal knowledge graphs: Construction, completion and applications," Mathematics, vol. 11, no. 8, p. 1815, 2023.
6. C. Peng, F. Xia, M. Naseriparsa and F. Osborne, "Knowledge graphs: Opportunities and challenges," Artificial Intelligence Review, p. 1–32, 2023.
7. B. Zhou, X. Shen, Y. Lu, X. Li, B. Hua, T. Liu and J. Bao, "Semantic-aware event link reasoning over industrial knowledge graph embedding time series data," International Journal of Production Research, vol. 61, no. 12, p. 4117–4134, 2023.
8. J. Zhang, B. Hui, X. Zhu, L. Tian and F. Hua, "Temporal knowledge graph reasoning based on discriminative neighboring semantic learning," Pattern Recognition, 2025.
9. S. Wan, Y. Wu, Y. Liu, L. Xiao and M. Guo, "A recommendation approach based on heterogeneous network and dynamic knowledge graph," International Journal of Intelligent Systems, vol. 2024, no. 1, p. 4169402, 2024.
10. L. Bai, M. Chen, L. Zhu and X. Meng, "Multi-hop temporal knowledge graph reasoning with temporal path rules guidance," Expert Systems with Applications, vol. 223, p. 119804, 2023.
11. M. Elhefny, M. Elmogy, A. Elfetouh and F. Badria, "Foorc: A fuzzy ontology-based representation for obesity related cancer knowledge," International Journal of Intelligent Computing and Information Sciences, vol. 16, no. 3, pp. 15--36, 2016.

12. Y. Gao, R. Li, E. Croxford, J. Caskey, B. W. Patterson, M. Churpek, T. Miller, D. Dligach and M. Afshar, "Leveraging Medical Knowledge Graphs Into Large Language Models for Diagnosis Prediction: Design and Application Study," JMIR AI, vol. 4, p. e58670, 2025.
13. B. Abu-Salih and S. Alotaibi, "A systematic literature review of knowledge graph construction and application in education," Heliyon, vol. 10, no. 3, 2024.
14. K. Qu, K. C. Li, B. T. Wong, M. M. Wu and M. Liu, "A survey of knowledge graph approaches and applications in education," Electronics, vol. 13, no. 13, p. 2537, 2024.
15. E. E. Kosasih, F. Margaroli, S. Gelli, A. Aziz, N. Wildgoose and A. Brintrup, "Towards knowledge graph reasoning for supply chain risk management using graph neural networks," International Journal of Production Research, vol. 62, no. 15, pp. 5596-5612, 2024.
16. A. G. Regino, R. O. Caus, V. Hochgreb and J. C. d. Reis, "Leveraging knowledge graphs for e-commerce product recommendations," SN Computer Science, vol. 4, no. 5, p. 689, 2023.
17. P. Li, Q. Zhao, Y. Liu, C. Zhong, J. Wang and Z. Lyu, "Survey and prospect for applying knowledge graph in enterprise risk management," Computers, Materials and Continua, vol. 78, no. 3, pp. 3825-3865, 2024.
18. A. Chis, A.-M. Ghiran and S. Alter, "Informing enterprise knowledge graphs with a work system perspective," Enterprise Modelling and Information Systems Architectures (EMISAJ), vol. 19, 2024.
19. C. Meilicke, M. W. Chekol, P. Betz, M. Fink and H. Stuckeschmidt, "Anytime bottom-up rule learning for large-scale knowledge graph completion," The VLDB Journal, vol. 33, no. 1, pp. 131-161, 2024.
20. C. Chen, F. Zheng, J. Cui, Y. Cao, G. Liu, J. Wu and J. Zhou, "Survey and open problems in privacy-preserving knowledge graph: merging, query, representation, completion, and applications," International Journal of Machine Learning and Cybernetics, vol. 15, no. 8, pp. 3513-3532, 2024.
21. M. Farghaly, M. Mounir, M. Aref and S. M. Moussa, "Investigating the challenges and prospects of construction models for dynamic knowledge graphs," IEEE Access, vol. 12, pp. 40973--40988, 2024.
22. Y. Wu, X. Pan, J. Li, S. Dou, J. Dong and D. Wei, "Knowledge graph-based hierarchical text semantic representation," International Journal of Intelligent Systems, vol. 2024, no. 1, p. 5583270, 2024.
23. L. Yao, C. Mao and Y. Luo, "KG-BERT: BERT for knowledge graph completion," arXiv preprint arXiv:1909.03193, 2019.
24. J. Lin, L. Wang, X. Lu, Z. Hu, W. Zhang and W. Lu, "Improving knowledge graph completion with structure-aware supervised contrastive learning," in Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024.
25. Z. Sun, Z.-H. Deng, J.-Y. Nie and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," ICLR, pp. 1-18, 2019.
26. H. Dong, P. Wang, M. Xiao, Z. Ning, P. Wang and Y. Zhou, "Temporal inductive path neural network for temporal knowledge graph reasoning," Artificial Intelligence, vol. 329, p. 104085, 2024.
27. J. Guo, J. Yu, M. Zhao, M. Yu, R. Yu, L. Xu, Y. Pan and X. Li, "TELS: Learning time-evolving information and latent semantics using dual quaternion for temporal knowledge graph completion," Knowledge-Based Systems, vol. 301, 2024.
28. X. Liu, T. Mao, Y. Shi and Y. Ren, "Overview of knowledge reasoning for knowledge graph," Neurocomputing, p. 127571, 2024.
29. L. Zhong, J. Wu, Q. Li, H. Peng and X. Wu, "A comprehensive survey on automatic knowledge graph construction," ACM Computing Surveys, vol. 56, no. 4, pp. 1--62, 2023.

30. D. FaDl, S. Abas and M. Aref, "An Approach for Automatic Arabic Ontology Generation," International Journal of Intelligent Computing and Information Sciences, vol. 17, no. 3, pp. 1--10, 2017.
31. J. Shen, C. Wang, L. Gong and D. Song, "Joint language semantic and structure embedding for knowledge graph completion," arXiv preprint arXiv:2209.08721, 2022.
32. I. Chami, A. Wolf, D.-C. Juan, F. Sala, S. Ravi and C. Ré, "Low-dimensional hyperbolic knowledge graph embeddings," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.
33. S. S. Dasgupta, S. N. Ray and P. Talukdar, "Hyte: Hyperplane-based temporally aware knowledge graph embedding," in Proceedings of the 2018 conference on empirical methods in natural language processing, 2018.
34. J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in Companion proceedings of the the web conference, 2018.
35. A. García-Durán, S. Dumančić and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in The 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018.
36. C. Xu, M. Nayyeri, F. Alkhoury, H. S. Yazdi and J. Lehmann, "Temporal knowledge graph embedding model based on additive time series decomposition," arXiv preprint arXiv:1911.07893, 2019.
37. J. Zhang, T. Wan, C. Mu, G. Lu and L. Tian, "Learning granularity representation for temporal knowledge graph completion," arXiv preprint arXiv:2408.15293, 2024.
38. Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang and X. Cheng, "Temporal knowledge graph reasoning based on evolutional representation learning," in Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, 2021.
39. X. Zhao, J. Miao, F. Yang and S. Pang, "Geometry Interaction Embeddings for Interpolation Temporal Knowledge Graph Completion," Mathematics, vol. 12, no. 13, 2024.
40. Y.-C. Lee, J. Lee, D. Lee and S.-W. Kim, "Learning to compensate for lack of information: Extracting latent knowledge for effective temporal knowledge graph completion," Information Sciences, vol. 654, 2024.
41. J. Z. Pan, S. Razniewski, J.-C. Kalo, S. Singhania, J. Chen, S. Dietze, H. Jabeen, J. Omeliyanenko, W. Zhang, M. Lissandrini and others, "Large language models and knowledge graphs: Opportunities and challenges," Transactions on Graph Data and Knowledge (TGDK), vol. 1, no. 1, 2023.
42. J. Zhang, B. Hui, C. Mu and L. Tian, "Learning multi-graph structure for Temporal Knowledge Graph reasoning," Expert Systems with Applications, vol. 255, p. 124561, 2024.
43. M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov and M. Welling, "Modeling relational data with graph convolutional networks," in The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3--7, 2018, proceedings 15, 2018.
44. X. Wang, S. Lyu, X. Wang, X. Wu and H. Chen, "Temporal knowledge graph embedding via sparse transfer matrix," Information Sciences, vol. 623, pp. 56-69, 2023.
45. K. Cho, "On the properties of neural machine translation: encoder-decoder approaches," arXiv preprint arXiv:1409.1259, 2014.
46. C. Shang, Y. Tang, J. Huang, J. Bi, X. He and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," in Proceedings of the AAAI conference on artificial intelligence, 2019.
47. F. Mahdisoltani, J. Biega and F. M. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in CIDR, 2013.

48. F. Erxleben, M. Günther, M. Krötzsch, J. Mendez and D. Vrandečić, "Introducing wikidata to the linked data web," in The Semantic Web--ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, Springer, 2014, pp. 50--65.
49. Y. Li, X. Zhang, Y. Cui and S. Ma, "Hyperbolic graph neural network for temporal knowledge graph completion," in Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024.