Neural Computing and Applications (2021) 33:16533–16555
https://doi.org/10.1007/s00521-021-06252-8

ORIGINAL ARTICLE

W-KG2Vec: a weighted text-enhanced meta-path-based knowledge graph embedding for similarity search

Phuc Do[1] · Phu Pham[1]

Received: 29 September 2020 / Accepted: 17 June 2021 / Published online: 2 July 2021

Abstract

Recently, similar entity searching over knowledge graph (KG) has gained much attentions by researchers. However, in rich-semantic KGs with multi-typed entities and relations, also known as heterogeneous information network, relevant entity search is considered as a challenging task due to the ambiguity as well as complexity of user's queries in realistic applications, such as QA chatbot and KG-based information retrieval. In this paper, we propose a novel approach, called W-KG2Vec which enables to automatically learn the semantic representations of entities in KG by applying the meta-path. The proposed W-KG2Vec is a meta-path-specific model which supports to evaluate both semantic relations as well as the text-based similarity between entities. The combination of text- and structure-based embedding mechanism of W-KG2Vec is promising to achieve better representations of entities in given KGs for handling complex user's queries. To effectively learn the sequential textual representations of entities' descriptions, we propose a combination of BERT pre-trained model with LTSM encoder, called BERT-Text2Vec. Then, the text-based similarity between entities is used to leverage our weighted meta-path-based random walk mechanism in W-KG2Vec model. Extensive experiences on real-world KGs (YAGO and Freebase) demonstrate the effectiveness of our proposed model against recent state-of-the-art KG embedding baselines.

Keywords Meta-path · Similarity search · Knowledge graph embedding · Document similarity · BERT · Bi-LSTM encoder

1 Introduction

Go along with the development of Internet, we are witnessing the tremendous growth of multidisciplinary information resources in the pattern of knowledge graphs (KG) [1, 9, 17, 32], such as Wikipedia, WordNet, YAGO, Freebase and DBpedia recently. In fact, KG plays an important role as indispensable auxiliary expert knowledge sources for constructing AI-based systems [4, 8]. A KG is a multi-relational graph [5] which is composed by large number of facts which can be denoted as triples in the form of <head relation tail>, e.g., <Donald_Trump, President_Of, USA>, <Elon_Musk, Founder_Of, SpaceX>. In recent years, many organizations and researchers have much concentrated on how to intensively extract latent features in given KGs by preserving and learning their structures. This approach is called KG embedding or KG representation learning. KG embedding [33, 37] has quickly gained massive attentions due to their wide applications in different domains, such as information retrieval [11], QA chatbot [2, 4] and recommendation [34]. In general, KG embedding can be used to compress high-dimensional complex data structure of entities and their associated relations of KGs into fixed low-dimensional and continuous data structure in the pattern of vectors. Then, these transformed vectors which are represented for entities and their relations will be used for multiple tasks, such as similar entities searching, entities clustering/classification and relation extraction. Moreover, KG embedding can also be used to achieve new unknown facts which do not exist in current KGs yet, such as relations/links predictions between unconnected entities or entity (head/tail) predictions with given relations, commonly called KG

✉ Phuc Do
phucdo@uit.edu.vn

Phu Pham
phamtheanhphu@gmail.com

[1] University of Information Technology (UIT), VNU-HCM, Ho Chi Minh City, Vietnam

completion. Relevant entities searching/querying [1, 3, 18] is considered as a primitive task for most of common applications of KG embedding. Recently, similar search in KG encounters many challenges due to the complexity and ambiguity of user's queries. For example, in realistic application such as QA chatbot [4], we usually encounter complex similar entity-based queries like as "*Which are places in Paris that are similar to the Louvre museum?,*" "*What are similar places of Vọng_Cảnh hill in Huế?*" The reasonable outputs for this type of query are more complicated than just finding top-k similar embedded entities which are closest to "*place/ museum:Louvre*" and "*place/ hill:Vọng_Cảnh*" entities in a given KG. In fact, there are multiple searching criteria must be fulfilled before returning the searching results to the end-users, e.g., the top returned results must be place-/museum-typed entities within the Paris city of France for the first query. A common technique for solving these complex searching tasks is to modelling given queries as meta-path-based patterns. A meta-path is a symmetric sequential order of entities and relations which indicates specific semantic meaning of interconnections between KG's entities. Back to previous example of finding similar entities of "*Louvre,*" we can formulate this query as a meta-path, as [place/museum]$\overset{\text{containedInPlace}}{\rightarrow}$[city]$\overset{\text{containedInPlace}}{\leftarrow}$ [place/museum] (as shown in Fig. 1a). Similar to the first example, the second query also can be modelled as a meta-path [14], as [place]$\overset{\text{containedInPlace}}{\rightarrow}$[city]$\overset{\text{containedInPlace}}{\leftarrow}$ [place] (Fig. 1-B).

However, the second query can be considered as complicated than previous one due to the ambiguity in multiple user's searching purposes. "*Vọng_Cảnh*" is a hill which is a common place for enjoying sightseeing by tourists, so similar entities of "*Vọng_Cảnh*" hill must be places which are suitable for enjoying sightseeing. In this example, "Ngự_Bình" mountain is considered as a top candidate for queries like this: "*Which place is similar to Ngự_Bình?*" The best answers for this type of query must satisfy two aspects: the place must be in "Huế" (same as "*Ngự_Bình*") and should be a mountain or a hill ("*Ngự_Bình*" is a mountain). Therefore, thorough evaluation of entity's concept/description is also necessary while computing the similarity between two entities in KGs.

1.1 Problem definition

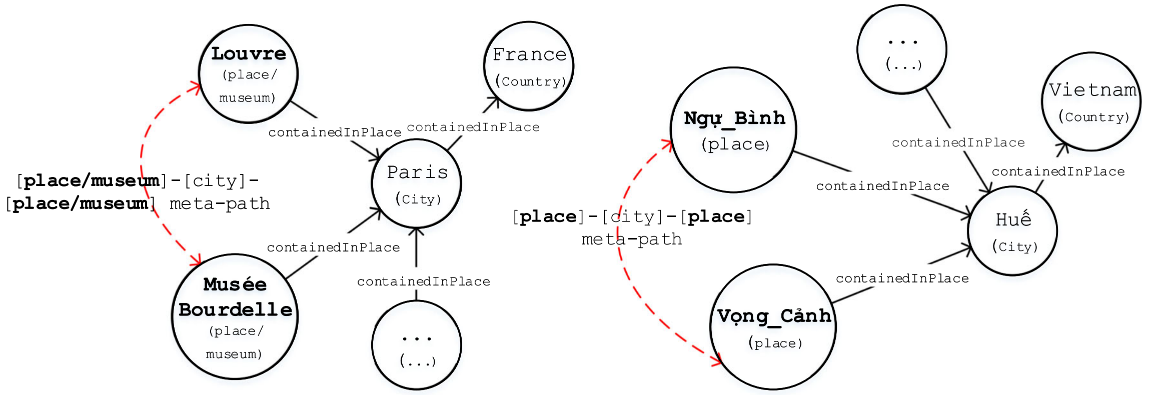
Definition 1 *Knowledge Graph (KG) as Heterogeneous Information Network (HIN)* is a directed labelled graph, denoted as $G = V, E, \phi, \psi$, where

- V stands for a set of entities/nodes in the given KG.
- E stands for a set of relations/links between entities/nodes in given KG. These relations might be binary (1 for existing relation and otherwise 0) or weighted relations.
- ϕ and ψ are two mapping functions, where

  o Node's type mapping function: $\phi(V) \mapsto \mathcal{A}$—with $V = \{v_1, v_2, \ldots, v_n\}$, a specific node ($v$) belongs to a specific type: a,$a \in \mathcal{A}$, we have: $\phi(v) = a$.

  p Edge's type mapping function: $\psi(E) \mapsto \mathcal{R}, \psi(e) \in \mathcal{R}$—with $E = \{e_1, e_2, \ldots, e_m\}$, a specific edge ($e$) belongs to a specific type: $\nabla, \nabla \in \mathcal{R}$, we have: $\psi(e) = r$.

- Traditionally, a knowledge graph is normally d as a set of triple notations, denoted as h, p, t, where$h$, t and p present for the head, tail objects and the predicate/link, respectively. A RDF triple is considered as a direct relation between two entities, e.g., h:Hà



A Similar places/museums in Paris, France

B Similar places for sightseeing in Huế, Vietnam

Fig. 1 Illustrations of modelling user's queries as meta-path-based patterns

Nội $\xrightarrow{\text{r: capital\_of}}$ tViệt_Nam. In previous studies, KG (also known as ontology) is specified as RDF (Resource Description Framework) graph which contains RDF triples, as$\{h, p, t\}$. These RDF triples are used to model direct relationships between entities in a given KG. However, this traditional representation of KG is unable to sufficiently model real-world KGs as heterogeneous networks which contain multi-typed entities and relations.

Definition 2 *Network Representation Learning (NRL)* [15]: given an information network, denoted as $G = (V, E)$, where (V) and (E) present for sets of network's nodes and edges, respectively. The ultimate goal of a NRL model is to find a mapping function, denoted as f for transforming the given set of network's nodes into d-dimensional vectors, as $f : V \rightarrow \mathbb{R}^{|V| \times d}$.

Definition 3 *Similarity search* via *NRL approach*: depending on a specific similarity searching purpose, NRL model is defined to capture specific relevant information between network's nodes in order to, respectively, map these features into similar vector spaces. Therefore, similar nodes with common distinctive features will be represented as similar vectors. To meet a specific relevant node search task, the mapping function (f) is designed accordingly for capture desired latent features of given network's nodes.

Similarity search is considered as the most common problem of knowledge graph (KG) mining [14, 18]. Similarity search in KG supports to find most relevant entities with the given user's queries. To solve similar search task in KG, knowledge graph embedding is considered the subarea of network representation learning (NRL) (Definition 2) which is recently a most well-known technique which supports to preserve and represent the structure of KG (entities and relations) into low-dimensional vectors [2, 8, 9, 17]. Then, we can simply measure the relevance entities by calculating the distance between their vectors. This approach is called similarity search over NRL approach (Definition 3). From the past, most of KG embedding models is considered as homogeneous embedding approach which considers all KG's entities and relations as the same type. However, in practical implementation, the ambiguity in user's queries which are represented as sequential relations within the complex structure of KG, also considered as heterogeneous information networks (HIN) [5] (Definition 1) with multi-typed entities and relations. In the past, to effectively achieve the representation of entities and their associated relations in a given KG, many embedding methods have been proposed recently. The most common approach for KG embedding is the distance translation-based approach with the most well-known Trans-family models (TransE [33], TransH [37], TransR [11]). The famous distance translation-based TransE [5] model is aimed to embed entities and relations in KG into the same fixed $|d|$-dimensional continuous latent space, denoted as $R^{|V| \times d}$, where $|V|$ is number of entities in a given KG. In the translation-based approach, TransE is designed to exploit the translation from head entity, (denoted as a vector $\boldsymbol{h}$) to tail entity $(\boldsymbol{t})$ regarding with their associated relation $(\boldsymbol{r})$ within a specific fact. The model is trained to achieve the objective that: $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$ (as illustrated in Fig. 2-A). In next improvement for resolving multiple relations between same head/tail entities, TransH is proposed the relation-specific hyper-plane projection mechanism for differentiating the roles of same entities with different relations in given facts (as illustrated in Fig. 2b). Similar to TransH, the TransE model employs the relation-specific spaces (instead of hyper-plane-based projection in TransH model) to separate same head/tail entities in different facts according to their corresponding relation. However, distance translation-based embedding techniques only focused on the direct relations/triples (which are occurred in facts) between entities rather than paths. Therefore, these distance translation-based KG embedding techniques are unable to handle complex querying tasks which are required evaluations on indirect interconnections between entities. Table 1 presents common notations which are used in our paper.

1.2 Existing challenges and motivations

In realistic requirements for knowledge extraction from KG, only using direct relations/triples to learn the representation of entities is insufficient. Due to the ignore of evaluating on paths/sequential relations between entities in KG, the representation output is unable to use for complex querying task. Recently, there are multiple studies which are focused on exploiting the sequential relations/paths between entities to leverage the knowledge representation outputs, such as PTransE [11] and RPE [2]. These models consider path-specific evaluation while learning the representation of entities which enable to solve complex querying task in KGs. However, recent proposed models of Lin et al. (PTransE and RPE) still paid less attention on the sequential order as well as relation's type within paths between entities in KGs. In fact, different orders of paths between entities might carry out different semantic meanings. For example, different paths between two entities "France" and "Eiffel_Tower" carry out different meaning like as France $\xrightarrow{\text{Contain}}$ Paris $\xrightarrow{\text{Contain}}$ Eiffel_Tower and France $\xrightarrow{\text{Capital}}$ Paris $\xrightarrow{\text{Has}}$ Eiffel_Tower. Therefore, different semantic

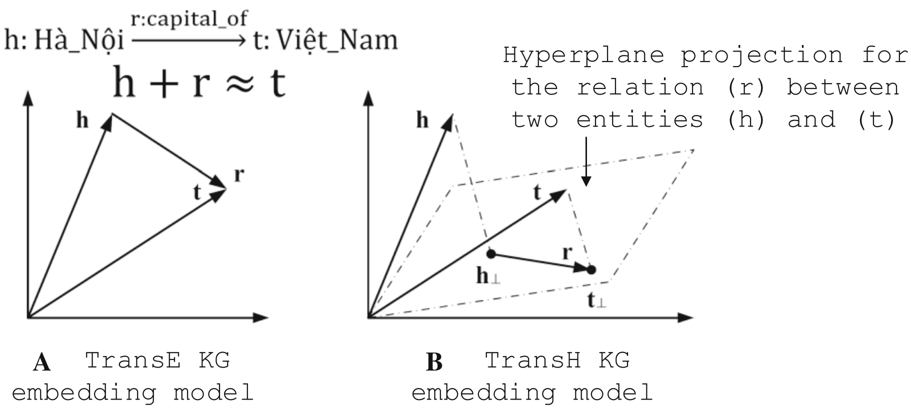

h: Hà_Nội
r:capital_of
t: Việt_Nam
h + r ≈ t
h
t
r
Hyperplane projection for the relation (r) between two entities (h) and (t)
h
t
r
h⊥
t⊥
A TransE KG embedding model
B TransH KG embedding model

Fig. 2 Illustrations of TransE and TransH KG embedding models

Table 1 List of notations which are used in this paper

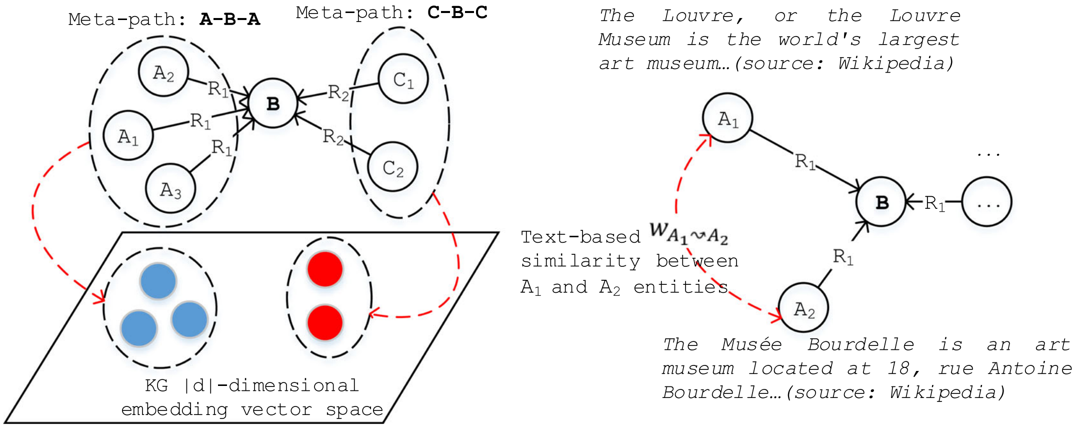
Notations	Descriptions
G	Information network/knowledge graph as a graph-based structure
V and $\mathcal{A}$	The sets of nodes and node's types of a(n) information network/knowledge graph, respectively
E and $\mathcal{R}$	The sets of edges and edge's types of a(n) information network/knowledge graph, respectively
$\mathcal{P}$	A meta-path
$\mathcal{P}_{x \rightsquigarrow y}$	A set of path instances between two nodes: (x) and (y) via meta-path: $\mathcal{P}$
$\phi(.)$ and $\psi(.)$	The mapping functions of nodes and edges in a heterogeneous information network
$\mathbb{R}^{\lvert V \rvert \times d}$	An embedding matrix with size: $\lvert V \rvert \times d$, where each vector row presents for an embedding vector of a node
BERT(.)	The BERT-based embedding layer
LSTM(.)	A single LSTM neural network cell
$\vec{w}$	The embedding vector of a word
$\vec{s}$	The embedding vector of a sentence
$\vec{h}$	The vector present for each hidden state in LSTM or BERT architectural layer
$w_{x \rightsquigarrow y}$	The weight of edge between two network's nodes (x) and (y)
$\pi_{x \rightsquigarrow y}$	The transitional probability between two network's nodes (x) and (y)

paths between same-typed entities should be embedded as different vectors.

Moreover, most of traditional KG embedding techniques only focus on the structural information of knowledge graph (relations between entities) and ignore the textual information which are tightly associated with entities. In fact, plain text which associated with entities in KG can help to provides abundant value information as we as support for entity and relation disambiguation while learning the representation of the given KGs. It is undeniable that textual data could play as a supplement for leveraging knowledge graph embedding task with both structural and contextual aspects. Recently, the joint of textual information and structure representation learning in KG has gained a lot of interests from researchers with multiple proposals [15, 32]. Recent researches [18, 35] focused on the combinations of textual information with structural information of KG to improve the representation outputs. However, joined text-based KG embedding models are considered as lack of thorough evaluations the sequential relations between entities in KG.

1.3 Our contributions

To fully incorporate between textual information and KG's structure in representation learning task in this paper, we propose a novel approach of text-enhanced meta-path-based embedding model, called W-KG2Vec. To properly capture the rich-semantic structure of given KGs, we apply the meta-path-based random walk mechanism to generate contextual entities for each given entity via different defined meta-paths which is inspired from our previous works [24–27]. Our principal assumption of applying meta-path-based guided representation learning in KG is same-typed entities which are interconnected via defined specific paths must be transformed into similar vectors in the given


Meta-path: A-B-A
Meta-path: C-B-C
A₂
A₁
A₃
R₁
R₁
R₁
B
R₂
R₂
C₁
C₂
KG |d|-dimensional embedding vector space
The Louvre, or the Louvre Museum is the world's largest art museum…(source: Wikipedia)
A₁
R₁
B
R₁
...
...
Text-based similarity between A₁ and A₂ entities
$w_{A_1 \rightsquigarrow A_2}$
R₁
A₂
The Musée Bourdelle is an art museum located at 18, rue Antoine Bourdelle…(source: Wikipedia)

A Meta-path-based KG embedding strategy of W-KG2Vec model

B Text-based similarity evaluation between entities following meta-path: **A-B-A**

Fig. 3 Illustrations of KG embedding strategies of proposed W-KG2Vec model

KG embedding space (as illustrated in Fig. 3a). Moreover, the random walk is guided by the transitional weight of text-based similarity between entities. The textual similarity measures between entities are identified by applying the collaborative self-attention of BERT [15] pre-trained model and sequential encoding to effectively learn the representation of textual data.

In this paper, we apply BERT pre-trained model with the bidirectional LSTM encoder to achieve the embedding of textual descriptions of entities in the given KGs. Then, these representations of descriptions are used to compute the text-based similarity between entities (as illustrated in Fig. 3b). Then, these computed text-based similarity scores are used to guide the meta-path-based random walk mechanism. The jointly representation learning of both textual information and KG's structure via meta-path-based random walk is promising to improve the quality of KG representation learning output. The main difference between our proposed model with other KG embedding model is the capability of capturing both semantic and local structural latent features of entities in the given KG to effectively fulfill the similarity search task. To sum up, our main contributions in this paper can be summarized as the following:

- The introduction of novel combination of BERT pre-trained model with Bi-LSTM encoder to support for learning the sequential representation of textual descriptions which are associated with entities in given KGs, called BERT-Text2Vec.
- The application of meta-path-based random walk mechanism in proposed W-KG2Vec model for generating contextual entities for each target entity in KG via defined meta-paths. Meta-path-based walks on KG are guided by the textual similarity weight between entities which are calculated by Bert-Text2Vec. Then, the extracted contextual entities are used to train the KG representation learning model.
- Extensive experiments on benchmark datasets with complex similar entity searching tasks demonstrate the effectiveness of our proposed model in comparing with recent state-of-the-art baselines.

In Fig. 4, we present an overall architecture of our proposed W-KG2Vec model. The rest of this paper has four main sections. In the second section, we review the related and discussing about the advantages/disadvantages of recent KG embedding techniques. In the third section, we briefly introduce about the approach of Bert-Text2Vec and W-KG2Vec models for text-enhanced meta-path-based KG embedding approach. Next, in Sect. 4, we present the extensive experiments and comparative studies on performance of proposed W-KG2Vec with recent KG embedding techniques. Finally, we conclude our works and present future improvement in Sect. 5.

2 Related works and motivations

In recent years, the use of KG for supporting AI-based systems has growth quickly. KG embedding has been proved to benefit multiple tasks such as information retrieval, question-answering system and relation extraction in different knowledge domains. KG embedding is designed to transform multi-typed connected data, in form of entities and their relations into a continuous fixed low-dimensional vector space. There are several popular KG

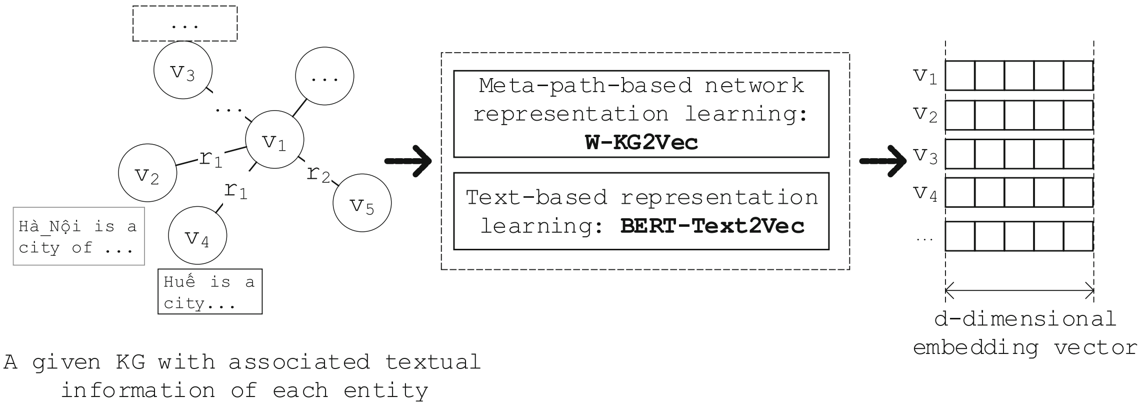

...
v3
...
...
v1
r1
v2
r1
r2
v5
Hà_Nội is a city of ...
v4
Huế is a city...
A given KG with associated textual information of each entity
Meta-path-based network representation learning: W-KG2Vec
Text-based representation learning: BERT-Text2Vec
v1
v2
v3
v4
...
d-dimensional embedding vector

Fig. 4 Overall architecture of our proposed W-KG2Vec model for KG embedding task

embedding techniques which can been categorized as two main groups, as the following.

2.1 Translational distance KG embedding approach

In the translational distance approach, proposed embedding techniques mainly depend on the structural information of the KGs, specifically the directed relationship between entities in form of triple: $\langle h, r, t \rangle$ in KGs. The most traditional well-known KG embedding technique is TransE [1] model. TransE is considered as a simple and effective method which support to learn the vector representations of both entities in relations in a given KG. TransE depends on a basic idea that a relation between head and tail entities are supposed to correspond to a distance translation between the representations of two given entities, denoted as $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$. We also know about the Unstructured Model (UM) [3] is an earlier version of TransE with the elimination of relations between entities in training the embedding model and the structured embedding (SE) [5] apply matrix projections to differentiate relations between same pairwise entities in KGs. However, the TransE model is only capable to present 1-to-1 relation between target entities which leads to the failure in translating 1-to-N, N-to-1 and N-to-N relations. Therefore, several improvements such as TransH, TransR and TransA have been proposed to overcome this problem by applying the relation transformation into different hyper-planes/subspaces. Beside Trans-family models, there are some proposals which can be considered as belong to translation-based KG embedding approach such as Gaussian-based KG embedding techniques (KG2E [13], TransG [38]) which mainly depend on multivariate Gaussian distributions for learning the representation of entities and relations. Similar to the approach of SE model, RESCAL [22] is considered as a bilinear-based model which represents relations between entities in KG as matrices. However, most of translational distance KG embedding techniques are considered as less-informative embedding approach due to the ignore of textual data, such as descriptions and concepts of entities while learning the representations. Textual information which is associated with KG's entities is now taken in consideration while training the KG embedding model in order to improve the output quality of entities and relations. This text-enhanced KG embedding trend has led to the proposals of some improved models recently. Recently, the proposed ConvE [31] is designed to perform a global 2-D convolution operation on the subject entity and relation embedding vectors. These embedding vectors are reshaped as the matrices and then concatenated. Similar to previous approach, the proposed RotateE [39] leverages the KG embedding model with multiple relations/entities by defining each relation as a rotation in the complex embedding spaces. However, these recent approaches are all considered as link/triple-based embedding approach without considering text-based semantic similarity between entities.

2.2 Jointly text-enhanced KG embedding approach

In recent times, researchers have gained much interests on jointly learning the representation of both structural information and textual information of KGs. There are multiple proposed techniques applying embedded textual data to leverage the quality of entities representations in KGs such as the use of average word representation [16, 31] in entities' names for identifying similar entities while training the KG embedding model. Inspiring from previous work [31] on jointly text-enhanced KG embedding, the next improvements [27, 39] proposed an extended embedding approach for learning the representation of entities' textual descriptions which help to enrich the embedding quality of entities in KG. In fact, in previous, the textual information and KG's structure embedding are learnt separately with different objective functions which leads to the sparsity of KG's entity representation. To overcome the separation in structure-based and text-based representations in KG embedding, a proposal [24] of using convolutional neural network (CNN) architecture, which enable to utilize both structure-based and text-based embedding aspects, can be called as J-CNN for short. Recently, with successes of auto-encoding approach, such as GPT [36] and BERT [27], in natural textual data processing, several studies have been adopted these techniques to leverage the performance of KG embedding task in the rich-textual context, such as KG-BERT [42] and K-BERT [41]. Through successes of previous jointly text-enhanced models in leveraging the overall output of KG embedding task, the textual information is proved as effective way for improving the quality of entity representations. However, for the textual data, mainly descriptions of given entities in KGs are existed in form of long-text documents separately learning the embedding of each word in these documents (continuous bag-of-words approach) might cause the information lost due to the sequential complex of textual data which are composed in natural language form.

3 Methodology

In this section, we introduce our proposed W-KG2Vec model for text-enhanced knowledge graph embedding by applying the jointly learning of textual descriptions of entities in KGs and meta-path-based random walk for generating contextual entities of each source entity with defined semantic sequential relations in form of meta-paths.

3.1 Preliminaries and definitions

Formulating a KG as a heterogeneous information network (HIN) (Definition 1), we denote $\mathcal{A}$ and $\mathcal{R}$ stand for the set of entity/node types and relation/link types, respectively. In context of HIN, a knowledge graph is considered as a directed labelled graph with V is a set of multi-typed entities which are connected by a set of multi-typed relations, denoted as E.

A KG is considered as rich-semantic if it has large number of entity types and relation types such as YAGO or Freebase contain thousands of entity's types and relation's types. In order to have an overview about the complexity of a KG, we need to look at its network schema KGNS (Definition 4). In fact, most of real-world KGs such as YAGO, Freebase and DBpedia have complex structure with number of entity types and relation types might be up to thousands. These KGs are considered rich-schematic KGs with complicated KGNS. For rich-schematic KGs, KGNS is really necessary for understanding possible occurred direct relation types between multiple entity types as well as defining semantic paths between pairwise entities. With the KGNS of a given KG, we can easily identify set of direct relation types can be occurred between two entity types as well as interconnected paths/sequence of relations. In a KG, two entities might be connected by not only direct relations but also via indirect sequential relations which carry out rich-semantic meanings.

Definition 4 *Knowledge Graph Network Schema (KGNS):* For a given KG, denoted as $G = V, E, \phi, \psi$ a KGNS is formally defined as a tuple, with: $\mathcal{A}_G, \mathcal{R}_G, \mathcal{E}_G, \mathcal{P}_G$ where $\mathcal{A}_G$,$\mathcal{A}_G = \bigcup_{v \in V} \phi(v)$ and $\mathcal{R}_G$, $\mathcal{R}_G = \bigcup_{e \in E} \psi(e)$ are entity types and relation types which appear in the given KG (G).$\mathcal{E}_G$ and $\mathcal{P}_G$ present for sets of direct relations and indirect relations in form of meta-paths (Definition 5) between entities in a given KG (G)

Definition 5 *Meta-path* ($\mathcal{P}$) [30]*:* It is defined as sequential relations between two entity types, normally a meta-path is defined as symmetric with same source and target entity type. A meta-path with (l)-length is defined in form of $\mathcal{P} = \mathcal{A}_1 \overset{\mathcal{R}_1}{\rightarrow} \mathcal{A}_2 \overset{\mathcal{R}_2}{\rightarrow} \ldots \overset{\mathcal{R}_l}{\rightarrow} \mathcal{A}_{l+1}$, where $\mathcal{A}_1, \mathcal{A}_2 \ldots \mathcal{A}_{l+1} \in \mathcal{A}$ and $\mathcal{R}_1, \mathcal{R}_2 \ldots \mathcal{R}_{l+1} \in \mathcal{R}$ are entity types and relation types which are occurred in given meta-path $\mathcal{P}$, respectively.

In a KG, the indirect sequential relations/paths between entities which are written as $v_1 \overset{e_1}{\rightarrow} v_2 \ldots \overset{e_l}{\rightarrow} v_{l+1}$. These indirect sequential paths connect two entities can be formulated as "meta-paths" (Definition 5). Beside the aspect of carrying rich-semantic meanings of relations between entities, the number of possible meta-paths also corresponds to distinctive features. In order words, existing meta-paths carry the real-world structural complexity of a given KG. There are some rich-schematic KGs, such as YAGO and Freebase, which the number of meta-paths which are possibly defined and might be much larger than simple-schematic KGs, such as DBLP and MovieLens. Between two same-typed entities, we might have multiple meta-paths with different length. In approach of heterogeneous network analysis and mining tasks, such as similarity search task, given meta-paths between entities are mostly defined by users in order to achieve different outputs. Or in other words, meta-paths are patterns of depending on querying purposes of the users. By applying user-specified meta-paths in KG embedding, we can flexibly utilize the entity representation outputs following the needs of retrieval tasks. In our W-KG2Vec model, instead of using all directed relation between entities in all KG's triples, the user-specified meta-paths will be used to train the KG's entity representation model.

In this paper, we propose a meta-path-based KG embedding technique which contains two modules. The first module is in charged for learning the representation of entities' textual descriptions by applying collaborative self-attention of BERT to learn the sentence-level representation then combining with LSTM encoder to produce the final representations of given entities' textual descriptions. This module is called BERT-Text2Vec. Next, for each entity in the given KG, we used the meta-path-based random walk mechanism to generate set of contextual entities which will be used for embedding model training process. The random walks on each entity will be controlled by the calculated similarity weight of that entity with its neighbors. Finally, the representation model is optimized by applying heterogeneous negative sampling with SGD.

3.2 BERT-Text2Vec: sequential textual data representation learning approach

The main goal of proposed BERT-Text2Vec in this paper is to learn the textual description representation of entities in a given KG. The textual description of each entity can provide supplementary information for entity's concept disambiguation (e.g., "JFK$_{\text{AirPort}}$" with "JFK$_{\text{Person}}$," "blackberry$_{\text{company}}$" with "blackberry$_{\text{fruit}}$") as well as text-based similarity evaluation (e.g., "Bengal_tiger$_{\text{animal}}$" with "Sumatran_tiger$_{\text{animal}}$," "Paris$_{\text{Location\_City}}$" with "Lyon$_{\text{Location\_City}}$" etc.). In fact, textual descriptions of entities in KGs are in form of long-text documents with multiple long sentences. Unfortunately, recently BERT pre-trained models have not yet been fine-tuned for long-text documents (larger than 512 words/tokens), so we need to propose a new approach for learning the representation of textual descriptions of entities in KGs, called BERT-Text2Vec. The BERT-Text2Vec is a combination of BERT pre-trained model with bidirectional LSTM encoder to fulfill long-text representation learning task.

3.2.1 Sentence representation learning with BERT pre-trained model

At first, we split each textual description (d) of entities in KGs into multiple (n) sentences, denoted as s, $d = \{s_1, s_2, \ldots, s_n\}$. Then, we apply the BERT pre-trained model to learn the representation of each word/token in the given sentence. Assuming that, for a sentence, we have list (m), (with $m < 512$) tokenized words (w), denoted as $s = \{w_1, w_2, \ldots, w_m\}$. We apply BERT pre-trained (BERT$_{\text{base}}$) model to learn the representation of each word in each sentence. For the original BERT pre-trained model contains 12 hidden layers, and 768 hidden units. We will use these output hidden units as the embedding vectors for words in each sentence. Each word is now represented as 768-dimensional vector, denoted as $\left\{\overrightarrow{w_1}, \overrightarrow{w_2}, \ldots, \overrightarrow{w_m}\right\} \in \mathbb{R}^{768}$. Then, we apply the Bi-LSTM architecture with global average pool to form representation of given sentence, denoted as $\vec{s}$. We construct the Bi-LSTM with the different parameters ($\theta_{\text{forward}}, \theta_{\text{backward}}$) to reflect the asymmetry of sentence processing. After that the hidden states of both forward ($\vec{h}$) and backward ($\overleftarrow{h}$) processes are concatenated and apply global average pool to form the final representation of a given sentence. The overall sentence representation learning can be described as following, (see Eq. 1):

$$\begin{aligned} \vec{h} &= \text{LSTM}(\overrightarrow{w_1}, \overrightarrow{w_2} \ldots \overrightarrow{w_m} | \theta_{\text{forward}}) \\ \overleftarrow{h} &= \text{LSTM}(\overrightarrow{w_m}, \overrightarrow{w_{m-1}} \ldots \overrightarrow{w_1} | \theta_{\text{backward}}) \\ \vec{s} &= \text{AvgPoo}l\left(\left[\vec{h}; \overleftarrow{h}\right]\right) \end{aligned} \tag{1}$$

where

- w is 768-dimensional vector which represents for each word/token in a given sentence (s) which is achieve from BERT.
- $\vec{s}$, is d-dimensional vector which represents for the given sentence, where d is number of LSTM cells which is used.
- $\vec{h}$ and $\overleftarrow{h}$ are hidden states of forward and backward processes, respectively.

Our objective of applying BERT for extracting word embedding in each sentence and forming the sentence embedding by applying Bi-LSTM technique is to capture the implicit discourse relations between words in each sentence. Given (d_s) is the initial size of embedding vector for sentence which is also number setup LSTM cells in forward and backward flows, the inputs of Bi-LSTM model are set of 768-dimensional embedded vectors which are represented for words in the given sentence (s).

Taking 768-dimensional embedded vector of each word ($\boldsymbol{w}_i$), the Bi-LSTM is used to learn the sequential orders of words' representations in both forward and backward processes. Finally, we concatenate hidden forward and backward states and apply AvgPool to form a final $|d_s|$-dimensional representation of given sentence (s). Through our experimental studies, the use of average pooling can help to achieve better performance than other vector combination strategy such as max pool or min pooling. Through careful evaluations of textual document representation via Bi-LSTM encoder, we figured out that the output latent hidden vectors are quite synthetic, and to softly aligned and combine the latent representations of these two hidden state vectors into a final document representation vector, average pooling strategy is considered as a suitable strategy. We conducted extra experiments to compare the performance of max, min and average pooling strategies on the overall W-KG2Vec model performance in Sect. 4.3.2.

The overall process of our sentence representation learning strategy is described in Fig. 5. Our approach of sentence representation learning by utilizing the sequential encoding mechanism of Bi-LSTM is inspired from previous approaches [7, 23]. However, a major difference of our sentence representation learning strategy in comparing with previous models is the application of BERT pre-trained model for achieving the bidirectional word embedding.

3.2.2 LSTM encoder for long-text document model

In order to capture sequential representation of sentence in each description of KG's entity, denoted as $\vec{d}$, we propose a technique of performing the textual embedding exchange between sentences through the state transition of recurrent neural network (RNN) architecture, resulting in a sequence of sentence states. From the sentence representations which are achieved by taking average output layer of BERT embedding, we apply the LSTM encoder to learn the final representation of given textual descriptions of KG's entities. Taking each sentence as each time-step input for LSTM encoder, the gated state transition operation for the hidden state h_j of (j)-th sentence, denoted as s_j can be defined as the following (see Eq. 2):

$$\begin{aligned} i_j &= \sigma\left(W_i \vec{s_j} + U_i h_j + b_i\right) \\ o_j &= \sigma\left(W_o \vec{s_j} + U_o h_j + b_o\right) \\ f_j &= \sigma\left(W_f \vec{s_j} + U_f h_{j-1} + b_f\right) \\ u_j &= \sigma\left(W_u \vec{s_j} + U_u h_i + b_u\right) \\ c_j &= i_j \odot u_j + f_j \odot c_{j-1} \\ h_j &= o_j \odot \tanh\left(c_j\right) \end{aligned} \tag{2}$$

where

- i_j, o_j and f_j present for input, output and forget gates, respectively.
- W, U and b are model parameters which are optimized during the training process.

Each embedded sentence will be passed to the given LSTM encoder and update the model the state transition process. In particular, the state transition between embedded vectors also consists state transition for each sentence within the given textual description of entity. In fact, these state transitions carry out information exchange between a sentence with all previous sentences that are composed within an entity's description. Finally, we take the last output hidden state of given LSTM encoder as the final representation for entities' descriptions in a given KG. The size of embedded document vectors equal to number of setup gated LSMT cells.

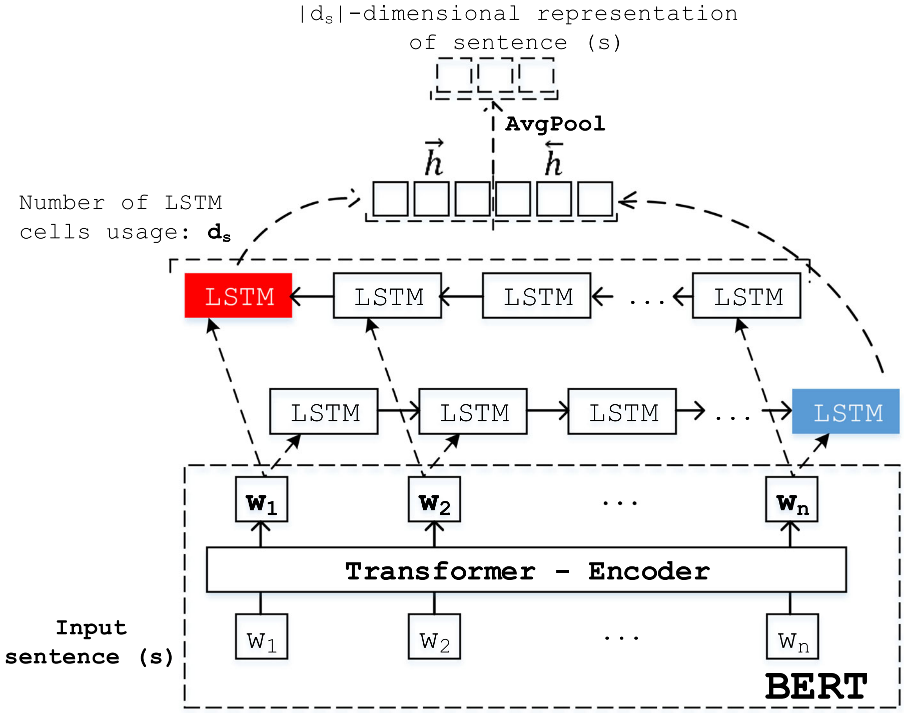


Fig. 5 Overall strategy of sentence representation learning of proposed BERT-Text2Vec model by applying BERT pre-trained model with Bi-LSTM encoder

In fact, textual data are frequently considered as a complex structure data where traditional sequential representation learning method such as GRU/Bi-GRU seems unfordable to fully capture latent features of textual document. In fact, the use of GRU/Bi-GRU can help to reduce the training computational effort and time-consuming due to the smaller number of model's parameters. However, our studies in this paper are majorly focused on how to improve the accuracy performance of KG embedding for similarity search task, therefore we designed to use the LSTM as the main textual sequential encoder in our approach. In Sect. 4.3.2, we also present experiments for comparing the difference in model's accuracy performance between the uses of Bi-GRU and Bi-LSTM.

3.3 W-KG2Vec: text-enhanced meta-path-based KG embedding approach

3.3.1 Text-based similarity weight between entities

From the representations of textual descriptions of entities which have been learnt in previous steps, we apply to cosine similarity to compute the text-based similarity between entities in a given KG. For any given source entity (v_s) and target entity (v_t), the text-based similarity weight, denoted as $w_{v_s \rightsquigarrow v_t}$, is calculated by following equation (see Eq. 3):

$$w_{v_s \rightsquigarrow v_t} = \frac{\overrightarrow{\mathrm{d}v_s} \,.\, \overrightarrow{\mathrm{d}v_t}}{\overrightarrow{\mathrm{d}v_s} \,.\, \overrightarrow{\mathrm{d}v_t}} \tag{3}$$

where

- $w_{v_s \rightsquigarrow v_t}$, is text-based similarity weight between source entity (v_s) and target entity (v_t).
- $\overrightarrow{\mathrm{d}v_s}$ and $\overrightarrow{\mathrm{d}v_t}$ are textual description representations of source entity (v_s) and target entity (v_t) in a given KG, respectively.

3.3.2 Meta-path-based random walk on KG

After learning the representation of textual descriptions in given KGs, we apply the meta-path-based random walk to generate contextual entities for each given entity. Given a KG, denoted as $G = V, E, \phi, \psi$, with a defined meta-path ($\mathcal{P}$). For any starting entity in KG, denoted as v_s and next target node: $\mathrm{v_t}$, where $\mathrm{v_s}$ and $\mathrm{v_t}$ is in same type, $\phi(v_s) = \phi(v_t)$, the transitional probability between source ($\mathrm{v_s}$) and target ($\mathrm{v_t}$) entities, following the given meta-path ($\mathcal{P}$) which is denoted as $\pi_{\mathrm{v_s} \rightsquigarrow \mathrm{v_t}}, \mathcal{P}$. This transitional probability is formulated by the following equation (see Eq. 4):

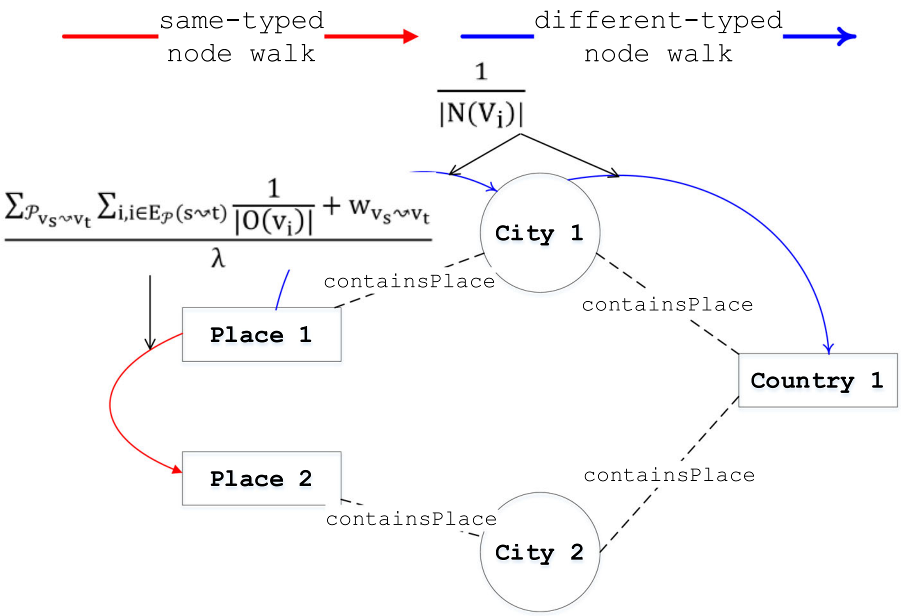
$$\pi_{v_s \rightsquigarrow v_t}, \mathcal{P} = \begin{cases} \text{with}\, e(v_s, v_t) \notin E \begin{cases} \dfrac{\sum_{\mathcal{P}_{v_s \rightsquigarrow v_t}} \sum_{i,i \in E_{\mathcal{P}}(s \rightsquigarrow t)} \dfrac{1}{|N(v_i)|} + w_{v_s \rightsquigarrow v_t}}{\lambda}, \text{ with } \phi(v_s) = \phi(v_t) (4a) \\ 0, \text{ with : } \phi(v_s) \neq \phi(v_t)\,(4b) \end{cases} \\ \dfrac{1}{|N(v_s)|}, \text{with}\, e(v_s, v_t) \notin E_{\mathcal{P}} (4c) \end{cases} \tag{4}$$

where

- $\sum_{\mathrm{i,i} \in \mathrm{E}_{\mathcal{P}}(\mathrm{s} \rightsquigarrow \mathrm{t})} \frac{1}{|\mathrm{N}(\mathrm{v_i})|}$, is the sum of transitional probabilities for meta-path-based walker to travel through all entities (v_i, with $O(v_i)$ is out-degree neighbors of v_i) (within given meta-path $\mathcal{P}$) between entity (s) and entity (t).
- $w_{v_s \rightsquigarrow v_t}$, is the text-based similarity weight between source entity ($\mathrm{v_s}$) and target entity (v_t) which is calculated by Eq. 3.
- λ, the global normalizing constant which help to normalize the value of transitional probability (π) within range [0, 1]. Normally, the value of normalizing constant is calculated for each meta-path-based walk by taking total transitional probability of all walks via different path instances of each meta-path.

Semantic-aware meta-path-based random walk (RW) over KG Our proposed defined random walk mechanism which are adopted in our previous works [14, 15] is mainly designed for generating contextual entities of each KG's entity which are then used for the network representation learning process. Our proposed meta-path-based RW mechanism contains two main types of walks, which are same-typed walk and different-typed walk. Let's take a meta-path: P(place)-C(city)-Ci(country)-C-P in YAGO knowledge as an example in this case (as illustrated in

Fig. 6 Illustration of meta-path-based random walk mechanism in W-KG2Vec model for generating contextual entities for each entity in a given KG


same-typed node walk
different-typed node walk
$\frac{1}{|N(V_i)|}$
$\frac{\sum_{\mathcal{P}_{v_s \rightsquigarrow v_t}} \sum_{i, i\in E_{\mathcal{P}}(s\rightsquigarrow t)} \frac{1}{|O(v_i)|} + w_{v_s\rightsquigarrow v_t}}{\lambda}$
City 1
containsPlace
containsPlace
Place 1
Country 1
Place 2
containsPlace
containsPlace
City 2

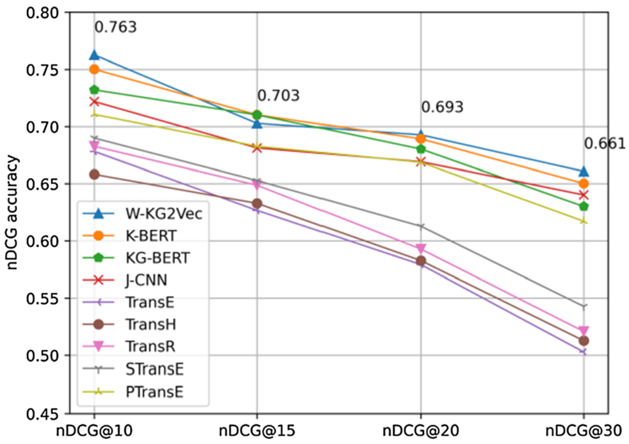


Fig. 7 Similar entity search task with different KG embedding techniques in YAGO-small dataset

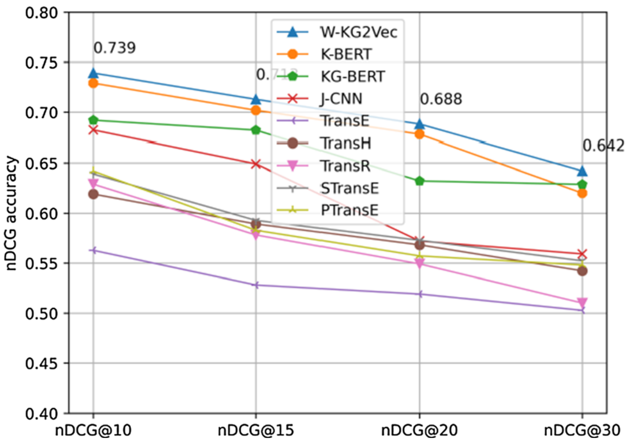


Fig. 9 Similar entity search task with different KG embedding techniques in YAGO-large dataset

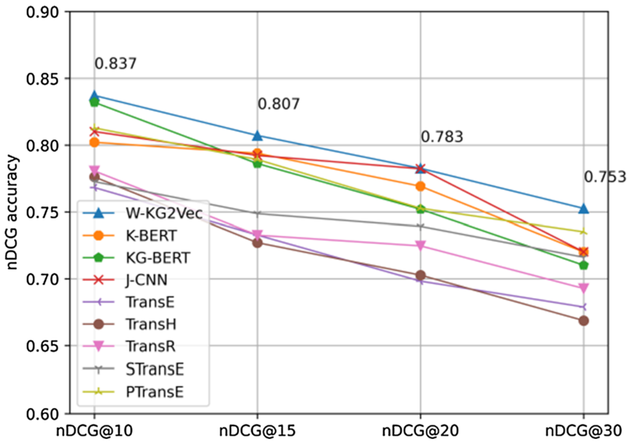


Fig. 8 Similar entity search task with different KG embedding techniques in Freebase-small dataset

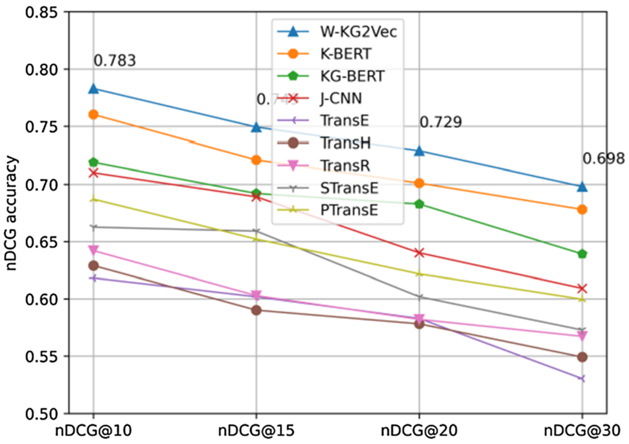


Fig. 10 Similar entity search task with different KG embedding techniques in Freebase-large dataset

Fig. 6). For walks between [place]-[city] and [city]-[-country] entities, we randomly select a next neighborhood node with a distribution of $\frac{1}{|N(v_s)|}$ (as shown in Eq. 4c). At the end of each path instance following the given meta-path, $\mathcal{P}$, we need to re-calculate all transitional probability between two target-typed entities (in this case is place) to identify the next target-typed entity for the next move of our walk via selecting the target-typed node with the maximum transitional probability (as identified by Eq. 4a) (Figs. 7, 8, 9, 10). The overall semantic-aware meta-path-based RW are controlled by predefined walk's length (l) and number of walks per node (w) which are applied in previous studies [22, 29].

Advantages of applying meta-path-based random walk on KG embedding For traditional approach of KG embedding, all relations between entities (in forms of triples) will be scanned and taken in consideration during the embedding process. This is considered as time-consuming task and needs more computing resources for the large-scaled KG representation learning process. On the other hand, random walk is considered a computational efficient approach for large-scaled KGs in terms of both computing resource and time requirements. The complexity of storing immediate same-typed neighbors of each entity is about $O(|E|)$. For meta-path-based random walk, it is useful to store the meta-path-based interconnections between next same-typed neighbors of every entity with: $O(\alpha^2|E|)$, where α is average out-degree of all entities in the given KG. For each entity, with the (k) number of contextual samples which are needed for each entity, we can choose a longer walk-length value, (k) with: $l > k$, which only needs effective computing complexity about: $O\left(\frac{l}{k(l-k)}\right)$ for each contextual sample. After the meta-path-based random walk process in a given KG, we will obtain a set of contextual entities for each KG's entity in form of $\{v, c_t\}$, with c_t presents for a set of same-typed (t) entities of a given entity (v). Then, similar to the approach of previous heterogeneous network representation learning approach such as Node2Vec [22] and Metapath2Vec [29], we applied the Skip-gram architecture of the well-known Word2Vec [28] model to generate training set for our proposed W-KG2Vec model. Specifically, considering a KG as a heterogeneous network with different-typed entities, our proposed KG embedding model is designed to learn embedding of different-typed entities over multi-typed generated entities via meta-paths, therefore we adopted the previous heterogeneous Skip-gram approach of Dong et al. in Metapath2Vec [29] to facilitate the heterogeneity of KG representation learning process.

Application of heterogeneous Skip-gram architecture From sets of generated contextual entities of each entity in the given KGs which are extracted by the meta-path-based random walk mechanism. We apply the heterogeneous skip-gram sampling technique to learn the representation of entity. In order to learn the representation of entities in the given KG, the model is aimed to maximize the probability of having a set of same-typed contextual entities, denoted as (c) for specific given entity (v) as following equation (see Eq. 5):

$$\arg\max_{\theta} \sum_{v \in V} \sum_{t \in T_V} \sum_{c_t \in N_t(v)} \text{Prob}(c_t|v;\theta) \tag{5}$$

where

- $N(v)$ and $N_t(v)$, are the set of neighborhood entities of (v) and set of neighborhood entities (v) with t-th type, respectively.
- $\text{Prob}(c_t|v;\theta)$, is the conditional probability of having context entities (c) which belong to t-th type with given entity (v).

The given probability of having a set of contextual entities (c_t) goes along with target same-typed entity (v) is normally defined as a softmax function, with:$\text{Prob}(c_t|v;\theta) = \frac{e^{X_{c_t} \times X_v}}{\sum_{u \in V, \phi(v)=\phi(c_t)} e^{X_{c_t} \times X_v}}$, where X_{c_t} and X_v are the row embedding vector of the given entity (v) and contextual entities (c_t), respectively. Then, the sampling distributions of contextual entities over each given entity are formulated by the given objective function (as shown in Eq. 6):

$$\mathcal{O}_{c_t,v_i} = log\sigma(X_{c_t} \times X_v) + \sum_{k=1}^{K} log\sigma\left(-X_{u_t^k} \times X_v\right) \tag{6}$$

where

- X_{c_t} and $X_{u_t^k}$, stand for the matrix rows of contextual entities (c_t) and set of negative sample entities (u_t^k), respectively.
- u_t^k, is defined as the k-th negative node which is sampled for context entities (c_t), in heterogeneous sampling approach the sampling entities: (u_t^k) and (c_t) are in the same type, or: $(\phi\left(u_t^k\right) = \phi(c_t))$.

Finally, the overall model's parameters are estimated by applying stochastic gradient descent (SGD) with gradients are updated by the following: $X_v = X_v - \eta \frac{\partial \mathcal{O}_{c_t,v_i}}{\partial X_v}; X_{u_t^k} = X_{u_t^k} - \eta \frac{\partial \mathcal{O}_{c_t,v_i}}{\partial X_{u_t^k}}$ with η is the setup learning rate. In more details, at the beginning, our model will iterate through all entities in a given KG to generated the corresponding contextual entities (c_t) for each target entity via our proposed semantic-aware meta-path-based random walk. Next, we applied the Skip-gram and negative sampling technique to optimize the probability of occurring same-typed contextual entities (c_t) for each target entity (v) (as shown in Eq. 5). Then, we applied the defined learning objective function (Eq. 6) to retrieve the representations of entities in a given KG with SGD.

3.3.3 Challenges of optimal meta-path's length and ambiguity in KG embedding

Normally, in KG as HIN-based embedding task, we might encounter challenges related to the long-length representation of semantic relations between entities in user' queries. The complexity of user's queries along with existing KG's relations might lead to common problems of infinite meta-path's length selection as well as ambiguity in the semantic representation between entities. Considering KG as a heterogeneous network with different-typed relations between entities, such as relations between common entities like as person ("Emmanuel_Macron," "Donald Trump") and location ("France," "USA") and there is no clue for which relation is important than the others to select for appropriated meta-path forming. In fact, most of recent HIN is rich in schematic with hundreds of relation's types which leads to difficulties in selecting proper relations for meta-paths which are used in the embedding process. Moreover, similar relations between entities also lead to ambiguity in different formed meta-paths which carried out different meanings and only some of them can be suitable for answering a specific user's query. If the length of formed meta-paths is too long, it might lead to problems related to the time-consuming of overall embedding process. Currently, to prevent these problems, we combined a previous approach of Changping M. et al. [22] for automatically discovering potential meta-paths between specific entity's types in KG with human-based knowledge expert to select proper meta-paths which is considered as a semi-supervised technique to obtain potential meta-paths for fulfilling user's queries. For the practical implementation of W-KG2Vec model, automatic discovered meta-paths between all types of entities will be showed for users to select which are the targeted semantic relations could be suitable for their queries.

4 Experiments and discussions

In this section, we conduct thorough experiments to demonstrate the effectiveness of our proposed W-KG2Vec model. Two well-known benchmark datasets are used in our experiments, including Freebase and YAGO. We implement W-KG2Vec with recent state-of-the-art KG embedding models for solving problem of similarity search task in KGs. The extensive comparative studies of proposed W-KG2Vec model with well-known KG embedding baselines show the effectiveness and scalability of our W-KG2Vec model performance in solving content-rich KG embedding task.

4.1 Dataset usage

To evaluate the performance of W-KG2Vec model with different KG embedding baselines, we use two main standard datasets, which are YAGO-{small, large} and Freebase-{small, large}. In these two KGs, we collect main entity types which are used for similar locations/places searching task, including (see Tables 2 and 3):

For experiments, we used to main datasets and each dataset has two different versions, as small and large. The main purposes of using different sizes of each KG in order to evaluate the influence of KG's size on the accuracy performance of each KG embedding model. As shown in Table 3, the number of extracted entities and relations for Freebase is quite smaller than YAGO. For the larger dataset of YAGO, we mainly use for extensive evaluations the scalability comparison between our proposed W-KG2Vec and other state-of-the-art KG embedding models.

4.2 Experimental setups

For textual description of each entity, we collected from multiple Internet resources, mainly from Wikipedia, DBpedia (content of "dbo:abstract" and "dbo:comments" fields). For W-KG2Vec model, we apply the BERT-Text2Vec model to learn the representation of textual descriptions, then these representations will be used for computing the text-based similarity weight $w_{v_s \rightsquigarrow v_t}$ in next processing steps. The numbers of vector's size for sentence and full-text description representation (number of LSTM cells) are both established as 128 for all experiments.

For locations/places in each knowledge graph, we intuitively labeled the level of similarity depending on the 12 tourist purpose aspects, which are: "amusement park," "beach," "historical," "lake," "market," "mountain," "museum," "national parks," "pagoda," "street," "temple" and "villages." One tourist location/place might be matched with multiple aspects, such as "Stonehenge" (Great Britain) can be labelled as {"historical/prehistoric," "mountain"}, "Hội_An" (Vietnam): {"village," "historical", "temple"}, or "Disneyland" (USA): {"amusement park," "museum"}, etc. Depending on labelled set of matching tourist aspects of each place we will score the similarity level each two pairwise locations/places as following (see Table 4):

The range of similarity scores as shown in Table 4 is adopted from previous studies of network's node similarity search task [12, 16, 22]. On two given KGs (YAGO and

Table 2 Selected entity and relation types for similar locations/places searching task in YAGO and Freebase *knowledge graph*

KG Usage	Selected entity types	Relation types
YAGO	schema:Country, schema:Place, schema:TouristAttraction, schema:GovernmentBuilding, schema: Museum, schema:AmusementPark	containedInPlace, containsPlace
Freebase	"Country," "Location," "City/Town/Village," "Amusement Park," "Tourist Attraction"	Contained_by, Contained_by (reversed)

Table 3 Number of extracted entities and relations which are used in experiments

KG Usage	Number of entities	Number of relations	Number of entity types	Number of relation types
YAGO-Small	3,846,790	5,187,731	6	2
YAGO-Large	6,582,779	11,093,421	6	2
Freebase-Small	624,698	1,633,240	5	2
Freebase-Large	1,235,492	3,639,482	5	2

Table 4 Scores for similarity level of two entities

Score	Description
0	Non-relevant
1	Quite relevant
2	Closely relevant
3	Very/highly relevant

Freebase), we conducted experiments on similar location/place-typed entities searching with different KG embedding models. The embedding vectors which represented for entities in two KGs are used to calculate the similarity scores (via cosine similarity) between location-/place-typed entities in queries and other same-typed entities in KGs. To evaluate the results of similar entities searching task in the given KGs, we use the nDCG (normalized Discounted Cumulative Gain) metric [19]. The average nDCG@10 (top-10 returned entities), nDCG@15, nDCG@20 and nDCG@30 of 100 queries for random 100 entities in each KG will be taken as the final results for comparisons. The use of sliding [k] range value from 10 to 30 in this paper is majorly inherited from previous network's node similarity search studies which follows the searching behavior of user in common search engine such as Google where the returned results in the first three pages (10 results for each page) are mainly focused. For the W-KG2Vec model, we implemented an experimental environment with the following model's configurations:

For other KG embedding baselines, we applied the golden configurations of each model from their original published works, such as STransE [30], PTransE [34], and RPE [6] (as shown in Table 5). To learn the representation of given KGs, we apply multiple meta-paths (as shown in Table 6) to capture the semantic meanings of interconnected relations of similar location/place-typed entities. We used the default configurations of Word2Vec [20] and Node2Vec [10] models for our neural network-based training process with learning rate is 0.025, and number of training epochs is about 300 for all datasets. For comparative studies with recent KG embedding techniques, we also implemented directed triple/relation-based KG embedding (including TransE [1], TransH, TransR,

Table 5 Configurations for other KG embedding baselines

Parameter	Value			
	Trans-family models [17] (TransE, TransH and TransR)	STransE [30]	PTransE [34]	RPE [6]
The vector dimension (d) for entity representation	100	100	100	100
Model's hyper-parameters (λ)	0.01	0.0005	0.005	0.8
Model's hyper-parameters (γ)	2	50	2	5

Table 6 Used meta-paths for KGs representation learning via W-KG2Vec model

KG Usage	Meta-paths		Semantic meanings
	ID	Details	
YAGO	Y-1	$\text{Place}\xrightarrow{\text{containedInPlace}}\text{Country}\xleftarrow{\text{containedInPlace}}\text{Place}$	Same places in a specific country (e.g., "*Paris,*" "*Lyon,*" etc. in France)
	Y-2	$\text{Place}\xrightarrow{\text{containedInPlace}}\text{Place}\xleftarrow{\text{containedInPlace}}\text{Place}$	Same places in a specific place (e.g., "*Yosemite national park,*" "*Golden Gate bridge,*" etc. in California, USA)
	Y-3	$\text{Museum}\xrightarrow{\text{containedInPlace}}\text{Place}\xleftarrow{\text{containedInPlace}}\text{Museum}$	Museums in a specific place (e.g., "*Vietnam History Museum,*" "*Áo Dài Museum*" in HCM city, Vietnam)
	Y-4	$\text{TouristAttraction}\xrightarrow{\text{containedInPlace}}\text{Place}\xleftarrow{\text{containedInPlace}}\text{TouristAttraction}$	Tourist attractions in a specific place (e.g., "*Bạch Mã national park,*" "*Vọng Cảnh hill,*" etc. in Huế, Vietnam)
	Y-5	$\text{AmusementPark}\xrightarrow{\text{containedInPlace}}\text{Place}\xleftarrow{\text{containedInPlace}}\text{AmusementPark}$	Amusement parks in a same specific place ("*Disneyland,*" "*Universal Studios Hollywood,*" etc. in California, USA)
	Y-6	$\text{GovernmentBuilding}\xrightarrow{\text{containedInPlace}}\text{Place}\xleftarrow{\text{containedInPlace}}\text{GovernmentBuilding}$	Government buildings in a same place (e.g., "*Houses of Parliament,*" "*Palace of Westminster,*" etc. in London, UK)
Freebase	F-1	$\text{City/Town/Village}\xrightarrow{\text{Contained\_by(reversed)}}\text{Country}\xleftarrow{\text{Contained}_{\text{by(reversed)}}}\text{City/Town/Village}$	Same cities, towns or villages in a specific country (e.g., "*Vernazza,*" "*Positano,*" "*San Gimignano*" in Italia)
	F-2	$\text{Location}\xrightarrow{\text{Contained}_{\text{by(reversed)}}}\text{Country}\xleftarrow{\text{Contained}_{\text{by(reversed)}}}\text{Location}$	Similar to meta-path [Y-1] and [F-1]
	F-4	$\text{Tourist Attraction}\xrightarrow{\text{Contained\_by(reversed)}}\text{Location}\xleftarrow{\text{Contained\_by(reversed)}}\text{Tourist Attraction}$	Similar to meta-path [Y-4]
	F-3	$\text{Amusement Park}\xrightarrow{\text{Contained\_by(reversed)}}\text{Location}\xleftarrow{\text{Contained\_by(reversed)}}\text{Amusement Park}$	Similar to meta-path [Y-5]

STransE [21]) and path-based KG embedding (including PTransE and RPE) techniques, the joint textual deep learning-based technique such as J-CNN [12] and BERT-based KG embedding models: KG-BERT [40] and K-BERT [19] for solving the same KG embedding tasks in the same datasets (YAGO and Freebase). For direct relation-based KG embedding techniques (Trans-family models and STransE), we apply the direct triples in two given datasets for training the entities representations. With the path-based KG embedding models (PTransE and RPE), we use the same meta-paths which are used in the W-KG2Vec model as the main training paths—$\mathrm{p} = (\mathrm{r}_1, \mathrm{r}_2 \ldots \mathrm{r}_l)$-between entities.

4.3 Experimental results & discussions

4.3.1 Similar entity searching on KG

For similar location-/place-typed entities searching task in both YAGO and Freebase, we randomly picked up location-/place-typed 100 entities and conducted the similarity searches. The returned entities for each query are sorted by top-10, top-15, top-20 and top-30 depending on the similarity weights between entities which calculated by cosine similarity on the embedded vectors of given pairwise entities. Then, these top-k returned entities were evaluated and ranked by level of relevance (see Table 4) before applying the nDCG metric to calculate the query accuracy score. Finally, the average accuracy scores of 100 queries were taken as the final results for each embedding technique. Tables 7 and 8 show the average top-k nDCG accuracy results for similar location-/place-typed entities searching task with different embedding models in YAGO and Freebase KGs, respectively.

For the small version of both two datasets, the experimental results show the outperformance of our proposed W-KG2Vec model (averagely 70.47% in YAGO and 79.49% in Freebase) in comparing with other state-of-the-art KG embedding techniques. In general, path-based methods gain better performance about 10.18% in term of nDCG metric than direct triple-based KG embedding techniques. The results implicitly indicate the advantages of applying paths in better capturing sematic meanings of relations between entities during the KG embedding task.

In fact, W-KG2Vec model achieves the better accuracy performance in Freebase which is considered as a smaller KG (<1 M entities) than YAGO (>3.8 M entities). In YAGO dataset, W-KG2Vec reasonably outperforms about 16.08% in comparing with direct triple-based KG embedding methods (TransE—18.09%, TransH—18.11%, TransR—15.29% and STransE—12.83%) and average 3.65% in comparing with path-based KG embedding methods (PTransE—5.21% and RPE—2.09%) and J-CNN (3.92%). For experiments on Freebase dataset, our proposed W-KG2Vec model also slightly improves the accuracy in term of nDCG with both direct triple-based (about 9.09%) and path-based KG (2.13%) embedding techniques and 2.4% for J-CNN technique.

Differently with large version of two datasets, where the number of entities is more than 2 times the small versions, the experimental outputs demonstrate a significant improvement of our proposed W-KG2Vec model in comparing with previous KG embedding models (as shown in Tables 7 and 8). In more details, the W-KG2Vec model outperforms Trans-family models (TransE, TransH and TransE) averagely about 21.76%, STransE (15%), PTransE (12.1%) and RPE (10.56%). Experimental outputs in the large version of two dataset demonstrate the effectiveness of our proposed model which can effectively capture richer semantic of relations between entities in context of large-scaled KGs.

Furthermore, in comparing with recent BERT-based KG embedding approaches, such as KG-BERT and K-BERT, our proposed W-KG2Vec also slightly outperforms averagely 1.31% (K-BERT)—3.98%(KG-BERT) and 3.25% (K-BERT)—4.97% (KG-BERT) in YAGO and Freebase datasets, respectively.

4.3.2 Experimental studies on representation learning approaches for KG similarity search task

The combination of textual and meta-path-based representation learning for KG embedding In this section, we demonstrate experiments related to the comparison between the use of sequential textual representation learning with the combined text-enhanced meta-path-based approach of W-KG2Vec model. Our proposed W-KG2Vec model is a combination between two modules, which are BERT-Text2Vec and the meta-path-based network embedding (MP2Vec) approach which is majorly inspired from the previous Metapath2Vec model [30]. Figures 11 and 12 demonstrate the separated performance evaluations of each embedding module in comparing with the completed W-KG2Vec model for the KG's entity similarity search task in YAGO-large and Freebase-large datasets.

As shown from the experiments, the combination between two embedding modules (BERT-Text2Vec and MP2Vec) of our proposed W-KG2Vec model can achieve better performance than separated usage of each embedding approach in the KG's entity similarity searching task.

Comparison on the uses of GRU/Bi-GRU and LSTM/Bi-LSTM for sequential textual encoding In Sect. 3.2, we

Table 7 Average nDCG@k accuracy for different embedding techniques on YAGO dataset

Dataset	Model	nDCG@10	nDCG@15	nDCG@20	nDCG@30
YAGO-small	TransE	0.67822	0.62672	0.57921	0.50281
	TransH	0.65821	0.63291	0.58271	0.51271
	TransR	0.68271	0.64872	0.59281	0.52082
	STransE	0.68992	0.65281	0.61281	0.54278
	PTransE	0.71062	0.68271	0.66872	0.61721
	RPE	0.74261	0.69982	0.67796	0.64072
	J-CNN	0.72193	0.68123	0.66921	0.64021
	KG-BERT	0.73213	0.71023	0.68029	0.63012
	K-BERT	0.75021	**0.71023**	0.68921	0.65021
	W-KG2Vec	**0.76261**	0.70281	**0.69271**	**0.66082**
YAGO-large	TransE	0.56281	0.52823	0.51923	0.50291
	TransH	0.61923	0.58921	0.56829	0.54231
	TransR	0.62912	0.57821	0.54921	0.51023
	STransE	0.63921	0.59281	0.57281	0.55231
	PTransE	0.64212	0.58291	0.55721	0.54812
	RPE	0.65821	0.62913	0.59821	0.56271
	J-CNN	0.68232	0.64912	0.57213	0.55921
	KG-BERT	0.69201	0.68212	0.63231	0.62892
	K-BERT	0.72912	0.70192	0.67812	0.62012
	W-KG2Vec	**0.73923**	**0.71291**	**0.68821**	**0.64213**

Table 8 Average nDCG@k accuracy for different embedding techniques on Freebase dataset

Dataset	Model	nDCG@10	nDCG@15	nDCG@20	nDCG@30
Freebase-small	TransE	0.76828	0.73281	0.69829	0.67891
	TransH	0.77621	0.72712	0.70281	0.66873
	TransR	0.78068	0.73271	0.72472	0.69281
	STransE	0.77271	0.74872	0.73928	0.71621
	PTransE	0.81293	0.78927	0.75271	0.73521
	RPE	0.82687	0.79562	0.77056	0.74392
	J-CNN	0.81023	0.79231	0.78231	0.72023
	KG-BERT	0.83213	0.78621	0.75201	0.71023
	K-BERT	0.80219	0.79392	0.76921	0.72012
	W-KG2Vec	**0.83727**	**0.80726**	**0.78261**	**0.75261**
Freebase-large	TransE	0.61823	0.60192	0.58291	0.53021
	TransH	0.62912	0.59021	0.57821	0.54912
	TransR	0.64213	0.60291	0.58213	0.56728
	STransE	0.66271	0.65921	0.60192	0.57281
	PTransE	0.68721	0.65221	0.62182	0.59972
	RPE	0.69281	0.67281	0.63821	0.59281
	J-CNN	0.71023	0.68923	0.64021	0.60923
	KG-BERT	0.71923	0.69212	0.68291	0.63921
	K-BERT	0.76021	0.72123	0.70123	0.67821
	W-KG2Vec	**0.78291**	**0.74921**	**0.72921**	**0.69819**

present the use of LSTM/Bi-LSTM sequential textual encoder to learn representations of words/sentences in each textual document which is associated with each KG's entity. However, there is a question of which type of sequential textual encoder can suitable for our proposed model in order to each the highest accuracy performance in similarity search task. We implemented our W-KG2Vec with two types of sequential textual encoders which are W-KG2Vec-LSTM and W-KG2Vec-GRU.

Experimental outputs (as shown in Figs. 11 and 12) demonstrate the use of LSTM can achieve better performance than GRU approximately 18.45% in terms of nDCG@k metric for both YAGO and Freebase dataset. This output proves that LSTM could be the best sequential textual encoder for our proposed W-KG2Vec model (Figs. 13, 14).

Comparison on the uses of different vector combination strategies In this section, we study the influence of different vector pooling strategies, which are denoted as max pooling, min pooling and average pooling on the overall W-KG2Vec model accuracy performance. Three versions of W-KG2Vec model are implemented corresponding to different pool strategies (W-KG2Vec-AvgPool, W-KG2Vec-MaxPool and W-KG2Vec-MinPool) to demonstrate the differences of model's accuracy performance in similar entity search task in both YAGO and Freebase. Figures 15 and 16 present that the use of the average pool strategy in the sequential textual representation learning process can help our proposed W-KG2Vec model can achieve the highest accuracy performance in both YAGO and Freebase datasets.

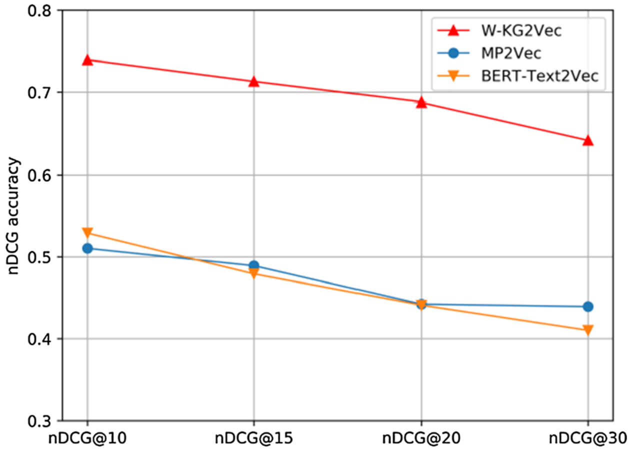


Fig. 11 Comparative studies between W-KG2Vec model with separated embedding modules (BERT-Text2Vec and MP2Vec) in YAGO-large dataset

4.3.3 Text-based similarity weight studies via different textual embedding techniques

The W-KG2Vec model mainly applied the textual representation of BERT-Text2Vec which is a combination of BERT pre-trained model and LSTM encoder. To demonstrate the outperformance of this combination in comparing with recent common textual representation learning

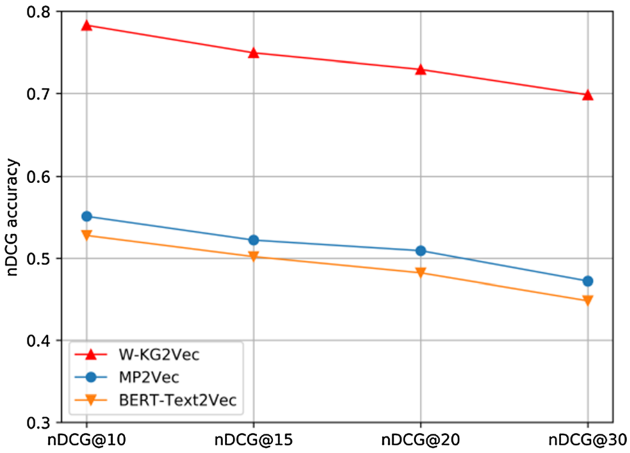


Fig. 12 Comparative studies between W-KG2Vec model with separated embedding modules (BERT-Text2Vec and MP2Vec) in Freebase-large dataset

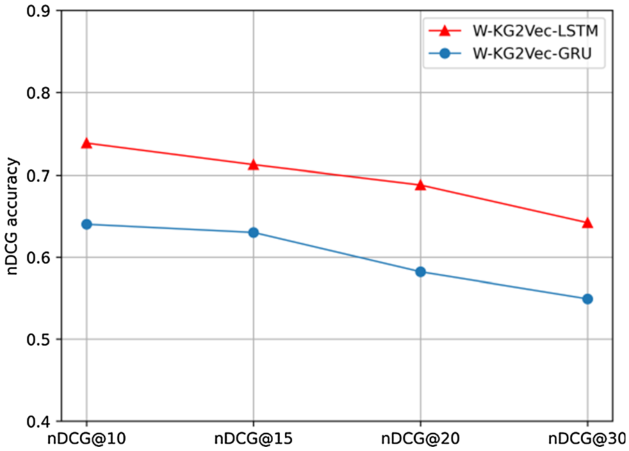


Fig. 13 Comparative studies between different types of sequential textual embedding techniques for W-KG2Vec model in YAGO-large dataset

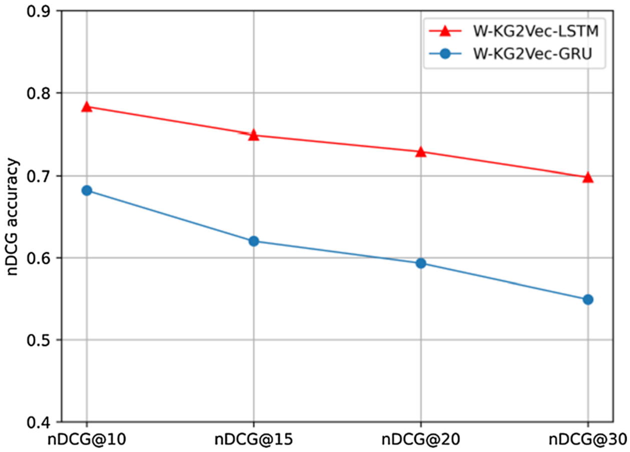


Fig. 14 Comparative studies between different types of sequential textual embedding techniques for W-KG2Vec model in Freebase-large dataset

techniques, such as topic modeling—Latent Dirichlet Allocation (LDA), Word2Vec and Doc2Vec, we implemented W-KG2Vec with different textual representation learning mechanisms to solve the same similar entities search task. The details of different textual representation learning implementations for W-KG2Vec are described as the following (as shown in Table 9):

To evaluate the performance of each textual representation learning techniques, we varied the size of two given KGs (YAGO and Freebase) from 10 to 100%. Each modified implementation of W-KG2Vec (described in Table 9) is applied to solve similar entities search task and reported the accuracy performance in terms of nDCG@30. Tables 10 and 11 present the accuracy outputs for each textual representation learning implementation of W-KG2Vec in terms of nDCG@30 on YAGO-small and Freebase-small datasets.

The experimental results (Figs. 17 and 18) demonstrate the effectiveness of applying our proposed BERT-Text2Vec for textual representation learning in comparing with recent well-known embedding techniques (LDA, Word2-Vec and Doc2Vec). In fact, previous textual representation learning techniques are lack of evaluations on the sequential relations between words in different document's contexts, therefore it leads to the significant decrease in the quality of textual representation outputs. In overall, the original W-KG2Vec implement with BERT-Text2Vec based textual representation learning outperforms about 5.63% (YAGO) and 5.78% (Freebase) other textual embedding techniques.

4.3.4 Parameter sensitivity studies

Model's parameters of network representation learning process In this section, we demonstrate experiments on the influence of model's parameters, including the walk length

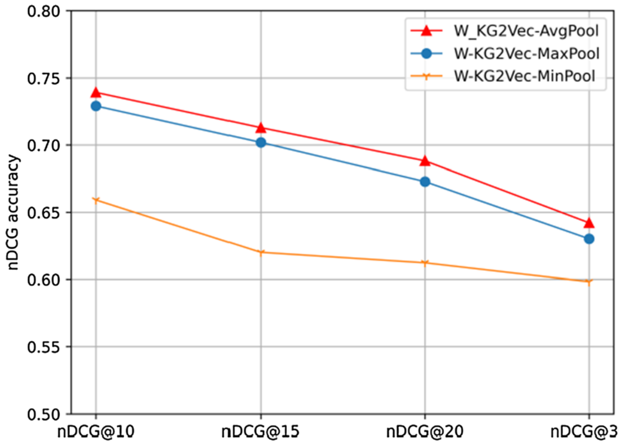


Fig. 15 Comparative studies between different types of vector combination strategies for W-KG2Vec model in YAGO-large dataset

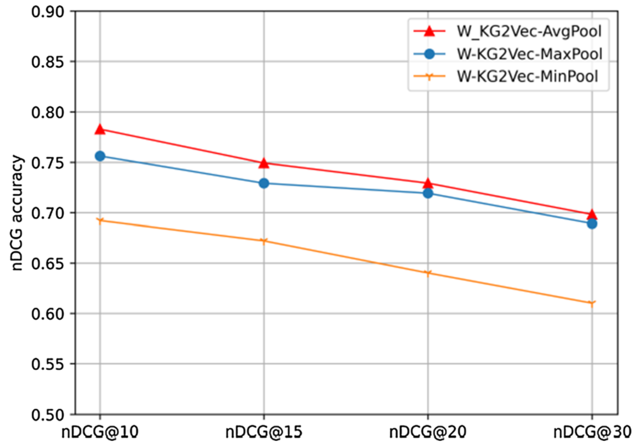


Fig. 16 Comparative studies between different types of vector combination strategies for W-KG2Vec model in Freebase-large dataset

(l), number walk per node (w) and the dimension of embedding vector (d) on KG’s embedding task in YAGO-small dataset. Following the same experimental procedure on similar entities searching task in YAGO-small, we varied the values of walk length (l), number walk per node (w) and embedding vector dimension (d) and reported the changes on proposed W-KG2vec’s accuracy performance in terms of nDCG@30 metric.

We conducted multiple experiments on similar entities search task with different values of model’s parameters. Figure 19 shows the experimental outputs for similar entities search task as the function of each of three model’s parameters while fixing the other two parameters. From the results, we observe the performance of our proposed W-KG2Vec model is gradually improved by increasing the number of walk per node (w). The model’s accuracy performance becomes stable when number of walk per node is going above 800. Similar to that, the increase of walk length (l) parameter also leads to significant improvement on overall model’s accuracy which is stable at above 120. The number of node and walk length parameters is considered as important in the aspect of semantic meaning capturing between entities in KGs. The higher setup number of (w) and (l) values are equal to higher number of contextual entities which are generated for each target

Table 9 Details of different textual representation learning implementations for W-KG2Vec

Implementation	Description	Configurations
W-KG2Vec (original)	This implement is setup with the default BERT-Text2Vec model for textual representation learning	Same configurations which are descried in Table 12
W-KG2Vec-LDA	We used the LDA topic model to learn the topic representations of given text corpus in two given datasets with the initial number of latent topics is 10 ($k = 10$)	Model’s parameters: Number of latent topics: 10 ($k = 10$) Number of words per latent topic: 20
W-KG2Vec-Word2Vec	Separated words in textual descriptions of two given KGs are learnt by applying well-known Word2Vec model. Then, these word embedding vectors will be used to form the full-text descriptions of entities by taking the average vector of all words which are occurred in the given descriptions	Model’s parameters: Window size (w): 5 Negative sampling batch size: 5 Word embedding dimension (d): 128
W-KG2Vec-Doc2Vec	For this implementation, we used the well-known Doc2Vec model to learn the textual representations of entities’ descriptions in two given datasets. For experiments with Doc2Vec model, we used both Distributed Memory (DM) and Distributed Bag-Of-Words (DBOW) implementations on the same dataset and reported the average results of both two approaches as the final experimental output	Model’s parameters: Window size (w): 5 Document embedding dimension (d): 128

Table 10 Evaluations of different textual representation learning implementations for W-KG2Vec on YAGO

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
W-KG2Vec-LDA	0.78921	0.77281	0.74281	0.73821	0.71207	0.68261	0.67261	0.66982	0.65078	0.64281
W-KG2Vec-Word2Vec	0.82108	0.81995	0.79021	0.76281	0.73526	0.70281	0.69787	0.67521	0.66872	0.66271
W-KG2Vec-Doc2Vec	0.81828	0.80821	0.80821	0.75281	0.72617	0.69281	0.68271	0.66852	0.65087	0.64071
W-KG2Vec	0.83215	0.83018	0.81722	0.77261	0.75627	0.73281	0.72891	0.71978	0.70972	0.70572

Table 11 Evaluations of different textual representation learning implementations for W-KG2Vec on Freebase

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
W-KG2Vec-LDA	0.80295	0.79741	0.76552	0.7455	0.74111	0.70325	0.69121	0.67993	0.67511	0.66207
W-KG2Vec-Word2Vec	0.82885	0.82823	0.8077	0.76929	0.74474	0.72354	0.72649	0.69333	0.69026	0.67195
W-KG2Vec-Doc2Vec	0.83889	0.83632	0.83278	0.76772	0.73829	0.70781	0.69429	0.68088	0.67034	0.65394
W-KG2Vec	0.83995	0.84175	0.83532	0.79589	0.76764	0.74709	0.75684	0.7397	0.73864	0.75297

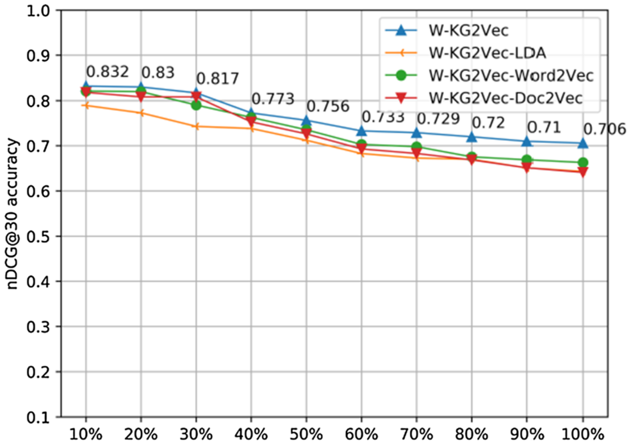


Fig. 17 Comparisons of different textual representation learning implementations for W-KG2Vec on YAGO

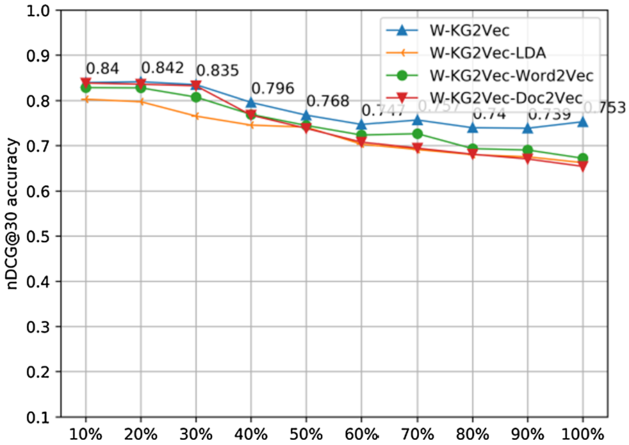


Fig. 18 Comparisons of different textual representation learning implementations for W-KG2Vec on Freebase

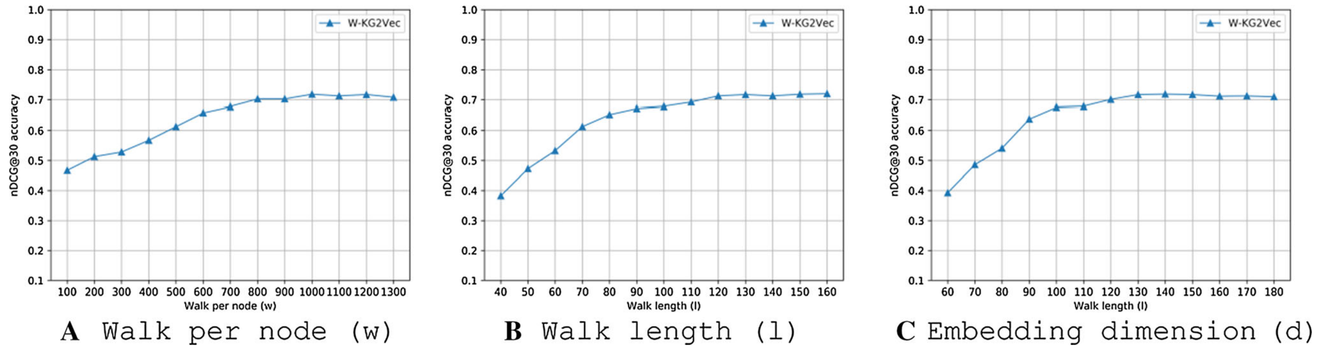
entity. In fact, for the training set generation process of our proposed W-KG2Vec model, each generated set of contextual entities of each target entity is majorly controlled by these two parameters which ensure the size of generated contextual entities is large enough to guarantee the quality of learnt entity embedding vectors.

For the embedding vector dimension parameter (d), we can see that when the embedding dimension is larger than 130, the accuracy performance of our proposed W-KG2Vec model reaches the highest value and become stable. The chosen number of embedding dimension for entities is also important which can influence the performance of overall representation learning process. With too high value of (d), our model will need more time as well as computer's resource to complete the training process. In order words, the configured dimensionality of embedding vector is quite important in network/KG embedding approach which is frequently chosen heuristically by evaluating the size and actual distinctive feature of entities in the given KG.

Different textual embedding approaches for W-KG2Vec model In previous section, we have demonstrated studies related to the use of different textual embedding models beside the BERT-Bi-LSTM, including LDA, Word2Vec and Doc2Vec models. To thoroughly evaluate the changes of model's hyperparameters, including the size of embedding vector dimensionality (Bert-Bi-LSTM, Word2Vec and Doc2Vec) and number of latent topics (LDA). As shown from experimental outputs in Fig. 20, our proposed W-KG2Vec model reaches the highest and stable accuracy performance with the value of embedding vector dimensionality (d) at range > 110 for Bert-Bi-LSTM approach, > 80 for both Word2Vec and Doc2Vec approaches. With the approach of LDA topic modeling, the model also reaches the highest accuracy performance with number of latent topic (k) is over 8.

4.3.5 System performance and scalability evaluations

In the era of big data with tremendous large-scaled KGs, it is important to demonstrate the efficiency and scalability of the proposed KG embedding model. To evaluate the efficiency and scalability of our proposed W-KG2Vec model, we conducted experiments with default configurations (as shown in Table 12) on a single server with Intel® Xeon® E7-8890 v4 CPU—24 cores CPU and 64 Gb memory. We ran the experiments with different number of threads from 1 to 24 and reported the speedup rates with respect to number of threads usage. In this experiment, we used YAGO-{small, large} which is considered as a quite large KG with more than 3.8 M entities and 5.1 M relations as the main dataset. Figures 21 and 22 show the average



A Walk per node (w) **B** Walk length (l) **C** Embedding dimension (d)

Fig. 19 Parameter sensitivity studies on W-KG2Vec model

speedup rate of our proposed W-KG2Vec model over multi-threaded running environment in two versions of YAGO dataset. As shown from the experimental outputs, the W-KG2Vec model achieves an acceptable sub-linear speedup rate which is quite close to the optimal line. Overall, this experiment demonstrates that our proposed W-KG2Vec model is efficient and scalable for handling large-scaled KGs with millions of entities and relations.

Table 12 Experimental setup parameters for W-KG2Vec model

Parameter	Value
Size of negative sampling (n)	5
Number of random walk (RW) start at each entity (γ)	80
The walk length (l)	100
The vector dimension (d) for entity representation	128
Number of walks per entity (w)	800

5 Conclusions and future works

In this paper, we formally present a novel approach of text-enhanced KG representation learning, called W-KG2Vec. In context of heterogeneous network with diverse types of entities and relations, KG embedding is considered as a challenging task for complex similar searching/querying task. A common technique for solving the complex entities searching in KGs is modelling user's queries as meta-path-based patterns. However, most of well-known KG embedding techniques are considered as direct relation/triple-based approach which is incapable to handle complex similar entities searching task. Other recent path-based KG embedding techniques are also lack of thorough evaluations on textual semantic meanings as well as diverse types of relations between KG's entities which leads to the decrease in quality of KG representation. To address these challenges, we proposed a joint textual representation learning with weighted meta-path-based random walk mechanism to leverage the accuracy performance of KG embedding task. The introduction of BERT-Text2Vec is our first contribution in this paper. BERT-Text2Vec is a combination of BERT pre-trained model and LSTM encoder which is aimed to learn the bidirectional sequential representations of textual descriptions of KG's entities. Then, these textual representations are used to compute the text-based similarity weights of pairwise entities in the given KGs. The computed text-based similarity weights between entities play an important role in our proposed weighted meta-path-based random walk strategy. The weighted meta-path-based random walk mechanism

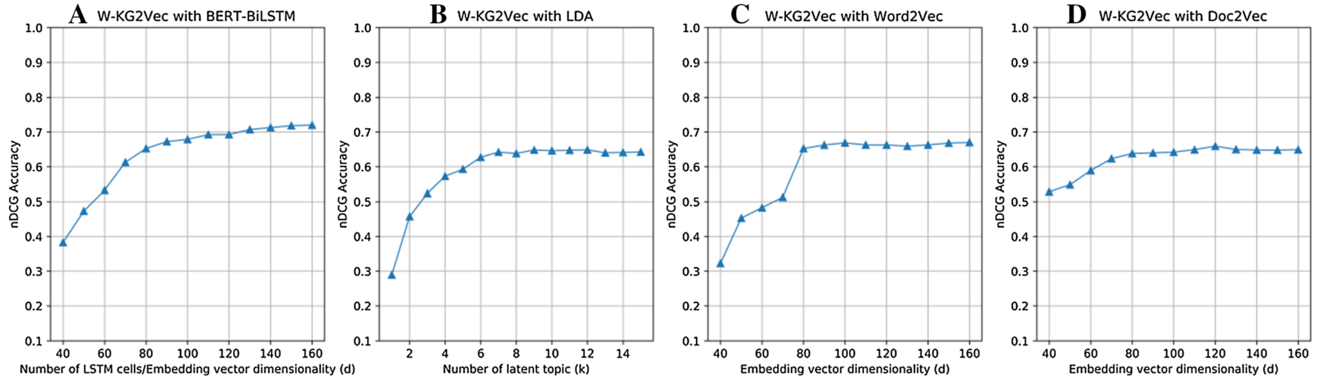


Fig. 20 Parameter sensitivity studies on different textual embedding approaches for W-KG2Vec model

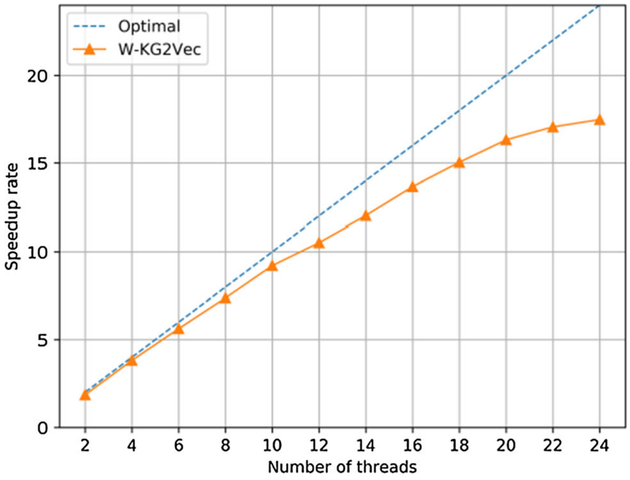


Fig. 21 Average speed-up rate for our proposed W-KG2Vec model in YAGO-small

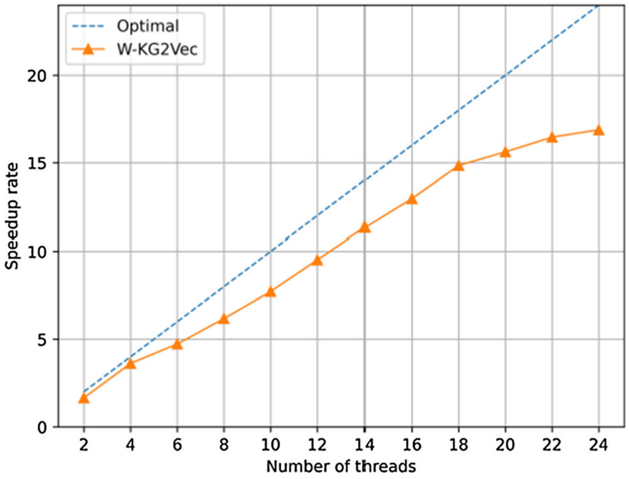


Fig. 22 Average speed-up rate for our proposed W-KG2Vec model in YAGO-large

supports to generate contextual entities for each entity in KGs which are used for training the representation model by applying heterogeneous skip-gram method. Extensive experiments on benchmark datasets demonstrate the capability of W-KG2Vec model on better handling complex entities searching/querying task in comparing with recent state-of-the-art KG embedding baselines. Our future works include various improvements which are mainly related to model's scalability. We tend to extend our proposed W-KG2Vec model to incorporate with the distributed processing platform such as Apache Spark which enable the capability of handling massive KGs with billions of entities and uncountable relations.

Acknowledgements This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2020-26-01.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O (2013) Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst 2:2787–2795
2. Bordes A, Weston J, Usunier N (2014) Open question answering with weakly supervised embedding models. In: Joint European conference on machine learning and knowledge discovery in databases
3. Bordes A, Glorot X, Weston J, Bengio Y (2012) Joint learning of words and meaning representations for open-text semantic parsing. Artif Intell Stat, pp 127–135
4. Bordes A, Chopra S, Weston J (2014) Question answering with subgraph embeddings. arXiv preprint https://arxiv.org/abs/1406.3676.
5. Bordes A, Weston J, Collobert R, Bengio Y (2011) Learning structured embeddings of knowledge bases. In: Twenty-fifth AAAI conference on artificial intelligence
6. Cao X, Shi C, Zheng Y, Ding J, Li X, Wu B (2018) A heterogeneous information network method for entity set expansion in knowledge graph. In Pacific-Asia conference on knowledge discovery and data mining
7. Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A (2017) Supervised learning of universal sentence representations from natural language inference data. arXiv preprint https://arxiv.org/abs/1705.02364
8. Dettmers T, Minervini P, Stenetorp P, Riedel S (2017) Convolutional 2d knowledge graph embeddings. arXiv preprint https://arxiv.org/abs/1707.01476
9. Fang Y, Wang H, Zhao L, Yu F, Wang C (2020) Dynamic knowledge graph based fake-review detection. Appl Intell 50:4281–4295
10. Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining
11. Han B, Chen L, Tian X (2018) Knowledge based collection selection for distributed information retrieval. Inf Process Manage 54(1):116–128
12. Han X, Liu Z, Sun M (2016) Joint representation learning of text and knowledge for knowledge graph completion. arXiv preprint https://arxiv.org/abs/1611.04125
13. He S, Liu K, Ji G, Zhao J (2015) Learning to represent knowledge graphs with Gaussian embedding. In: Proceedings of the 24th ACM international on conference on information and knowledge management
14. Hussein R, Yang D, Cudré-Mauroux P (2018) Are meta-paths necessary? revisiting heterogeneous graph embeddings. In: Proceedings of the 27th ACM international conference on information and knowledge management
15. Li B, Pi D (2020) Network representation learning: a systematic literature review. Neural Comput Appl 32:1–33
16. Lin X, Liang Y, Giunchiglia F, Feng X, Guan R (2019) Relation path embedding in knowledge graphs. Neural Comput Appl 31 (9):5629–5639
17. Lin J, Zhao Y, Huang W, Liu C, Pu H (2020) Domain knowledge graph-based research progress of knowledge representation. Neural Comput Appl 33:1–10

18. Lin Y, Liu Z, Sun M, Liu Y, Zhu X (2015) Learning entity and relation embeddings for knowledge graph completion. In: Twenty-ninth AAAI conference on artificial intelligence
19. Liu W, Zhou P, Zhao Z, Wang Z, Ju Q, Deng H, Wang P (2020) K-BERT: enabling language representation with knowledge graph. In: AAAI
20. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprin*t* https://arxiv.org/abs/1301.3781
21. Nguyen DQ, Sirts K, Qu L, Johnson M (2016) Stranse: a novel embedding model of entities and relationships in knowledge bases. arXiv preprint https://arxiv.org/abs/1606.08140
22. Nickel M, Tresp V, Kriegel HP (2011) A three-way model for collective learning on multi-relational data. In ICML
23. Nie A, Bennett E, Goodman N (2019) DisSent: learning sentence representations from explicit discourse relations. In Proceedings of the 57th annual meeting of the association for computational linguistics
24. Pham PDP (2019) W-MetaPath2Vec: the topic-driven meta-path-based model for large-scaled content-based heterogeneous information network representation learning. Expert Syst Appl 123:328–344
25. Pham P, Do P (2020) W-Metagraph2Vec: a novel approval of enriched schematic topic-driven heterogeneous information network embedding. Int J Mach Learn Cybern 11:1–20
26. Pham P, Do P (2020) W-Com2Vec: A topic-driven meta-path-based intra-community embedding for content-based heterogeneous information network. Intell Data Anal 24(5):1207–1233
27. Pham P, Do P, Ta CD (2018) W-PathSim: novel approach of weighted similarity measure in content-based heterogeneous information networks by applying LDA topic modeling. In: Asian conference on intelligent information and database systems
28. Socher R, Chen D, Manning CD, Ng A (2013) Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems
29. Sun Z, Deng ZH, Nie JY, Tang J (2019) Rotate: knowledge graph embedding by relational rotation in complex space. arXiv preprint https://arxiv.org/abs/1902.10197
30. Sun Y, Han J, Yan X, Yu PS, Wu T (2011) Pathsim: meta path-based top-k similarity search in heterogeneous information networks. In: Proceedings of the VLDB endowment
31. Toutanova K, Chen D, Pantel P, Poon H, Choudhury P, Gamon M (2015) Toutanova, Representing text for joint embedding of text and knowledge bases. In: Proceedings of the 2015 conference on empirical methods in natural language processing
32. Wang H, Jiang S, Yu Z (2020) Modeling of complex internal logic for knowledge base completion. Appl Intell 50:3336–3349
33. Wang Q, Mao Z, Wang B, Guo L (2017) Knowledge graph embedding: A survey of approaches and applications. IEEE Trans Knowl Data Eng 29(12):2724–2743
34. Wang X, He X, Cao Y, Liu M, Chua TS (2019) Kgat: knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining
35. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph embedding by translating on hyperplanes. In: Twenty-Eighth AAAI conference on artificial intelligence
36. Wang Z, Zhang J, Feng J, Chen Z (2014) Knowledge graph and text jointly embedding. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)
37. Wu Y, Pan J, Lu P, Lin K, Yu Z (2017) Knowledge graph embedding translation based on constraints. J Inf Hiding Multimedia Signal Process Ubiquitous Int 8(5):1119–1131
38. Xiao H, Huang M, Zhu X (2016) TransG: A generative model for knowledge graph embedding. In: Proceedings of the 54th annual meeting of the association for computational linguistics (Volume 1: Long Papers)
39. Xu J, Chen K, Qiu X, Huang X (2016) Knowledge graph representation with jointly structural and textual encoding. arXiv preprint https://arxiv.org/abs/1611.08661
40. Yao L, Mao C, Luo Y (2019) KG-BERT: BERT for knowledge graph completion. arXiv preprint https://arxiv.org/abs/1909.03193
41. Zhang D, Yuan B, Wang D, Liu R (2015) Joint semantic relevance learning with text data and graph knowledge. In: Proceedings of the 3rd workshop on continuous vector space models and their compositionality
42. Zhong H, Zhang J, Wang Z, Wan H, Chen Z (2015) Aligning knowledge and text embeddings by entity descriptions. In: Proceedings of the 2015 conference on empirical methods in natural language processing