IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 2, 2020, accepted May 11, 2020, date of publication May 15, 2020, date of current version May 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995074

Utilizing Textual Information in Knowledge Graph Embedding: A Survey of Methods and Applications

FENGYUAN LU, PEIJIN CONG, AND XINLI HUANG, (Member, IEEE)
School of Computer Science and Technology, East China Normal University, Shanghai 200062, China

Corresponding author: Xinli Huang (xlhuang@cs.ecnu.edu.cn)

This work was supported in part by the National Key Research and Development Plan of China under Grant 2019YFB2012803, in part by the Key Project of Shanghai Science and Technology Innovation Action Plan under Grant 19DZ1100400 and Grant 18511103302, and in part by the Key Program of Shanghai Artificial Intelligence Innovation Development Plan under Grant 2018-RGZN-02060.

ABSTRACT Techniques that map the entities and relations of the knowledge graph (KG) into a low-dimensional continuous space are called KG embedding or knowledge representation learning. However, most existing techniques learn the embeddings based on the facts in KG alone, suffering from the issues of imperfection and spareness of KG. Recently, the research on textual information in KG embedding has attracted much attention due to the rich semantic information supplied by the texts. Thus, in this paper, a survey of techniques for textual information based KG embedding is proposed. Firstly, we introduce the techniques for encoding the textual information to represent the entities and relations from perspectives of encoding models and scoring functions, respectively. Secondly, methods for incorporating the textual information in the existing embedding techniques are summarized. Thirdly, we discuss the training procedure of textual information based KG embedding techniques. Finally, applications of KG embedding with textual information in the specific tasks such as KG completion in zero-shot scenario, multilingual entity alignment, relation extraction and recommender system are explored. We hope that this survey will give insights to researchers into textual information based KG embedding.

INDEX TERMS Knowledge graph embedding, textual information, text-based embedding, text-improved embedding, embedding-based applications.

I. INTRODUCTION

Recent years, KG has experienced rapid development. Some typical achievement have been constructed and published, e.g., YAGO [1], Freebase [2] and DBpedia [3]. KG provides a structure form to store the human knowledge. It is a structured representation of relational facts, composed of entities, relations and descriptions. Entities represent concrete objects and abstract concepts, and relations represent the relationships between the entities and descriptions define or describe the entities. The knowledge, called *fact* as well, is particularly stored as a triple (*head entity*, *relation*, *tail entity*) under the scheme resource description framework (RDF). Given a fact that moon is the satellite of earth, it is stored as *(Moon, satellite of, Earth)*. KGs make the knowledge available to the intelligent systems and is applied to many artificial intelligence task, such as recommendation systems [4], question answering [5], [6], semantic parsing [7], named entity disambiguation [8] and information extraction [9].

Knowledge representation learning or KG embedding is one of the advanced researches, projecting the elements in KG into the continuous vector space. Such technique is widely applied to KG-related tasks, such as KG completion [10], relation extraction [11], entity classification [12]. Much work has been proposed, e.g., TransE and its extensions [14]–[19] and semantic matching models [20]–[23]. They directly represent entities and relations in the real-valued point-wise or complex vector space. Representations are made compatible to the ture facts existing in KG, relying heavily on the connectivities in KG. Yet, most KGs are built through semi-automatical methods, which makes them incomplete. The performance of the KG embedding in

The associate editor coordinating the review of this manuscript and approving it for publication was Chang Choi.

the downstream tasks could suffer from the spareness [24]. Considering that entities in KGs usually have precise and concise description. Examples of the entity description are illustrated in FIGURE 1. In addition, rich external context information is also available, e.g., textual mentions and text corpus. The textual information could be used to discover new relationships and offer precise expression. More importantly, it is slightly affected by the spare connectivity in KG. As a result, methods that utilize the textual information in KG embedding start to get attention [25]–[29]. We category these works according to if the representation is built from the textual informaiton: (i) Text-based KG embedding: The textual information are encoded to represent the entities and relations. Then the text-based representation is used to extend the existing KG embedding techniques or do the embedding task alone. The encoding models are learned by maximizing the overall plausibility of the facts in KG. (ii) Text-improved KG embedding: Unlike the former, no encoding models are made to be compatible with the facts and are used to build the text-based representation. The textual information is incorporated in different phases of the existing techniques, i.e., initialization, augmenting the representation and joint embedding, aiming to achieve better performance.

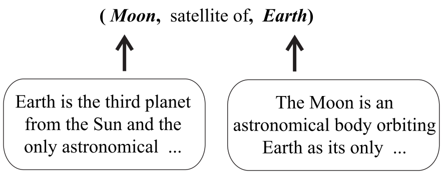
(**Moon**, satellite of, **Earth**)
Earth is the third planet from the Sun and the only astronomical ...
The Moon is an astronomical body orbiting Earth as its only ...

FIGURE 1. Example of the entity description [25].

Previous literature has made a survey on relational learning for KG [30], KG refinement [31], KG embedding [32] and knowledge representation learning [33]. Ji *et al.* [34] make a comprehensive review on KG representation, acquisition and application. Wang *et al.* [32] categorized the KG embedding based on facts alone with the scoring function and discussed the different types of auxiliary information utilized in KG embedding. Lin *et al.* [33] focus on the quantitive analysis of the KG embedding techniques. These work just briefly review few methods which incorporated the textual information. However, in this paper, we propose a comprehensive review on techniques that utilize the textual information in KG embedding, including state-of-the-art and latest trends. They are classified into two catogories based on whether the work builds the representation form the texts. Additionally, we further introduce the details in the training procedure and how the embedding with textual information is applied to the downstream tasks.

The remaining parts are organized as follows. Section II provides the premise, including the notation and definition, KG embedding with facts and various textual information. Then we introduce the text-based KG embedding in Section III and describe the encoding models and scoring function. Section IV reviews the text-improved KG embedding techniques which utilize the texts to achieve better performance. Section V discusses the training procedure of the mentioned methods. Some specific applications of KG embedding with textual information are further explored in section VI. At last, the conclusion and future directions are presented in section VII.

II. PREMISE

A. NOTATION AND DEFINITION

In this section, we first introduce KG's definition and components. Following the definition in [32], KG is defined as $G = \{E, R, F\}$, where E denotes the set of entities, R refers to the set of relations and F represents the set of facts. Inparticular, a fact is defined as a triple (h, r, t), where h, r, and t denote the head, relation, and tail, respectively. Other notations and corresponding descriptions are listed in TABLE 1.

TABLE 1. Notations and descriptions.

Notation	Description
G	A KG
F	A set of facts/triples
$e \in E, r \in R$	Entity set and relation set
(h, r, t)	A triple/fact contains head, relation and tail
$(\mathbf{h}_s, \mathbf{r}_s, \mathbf{t}_s)$	Structure-based embedding of head, relation and tail
$(\mathbf{h}_d, \mathbf{r}_d, \mathbf{t}_d)$	Text-based embedding of head, relation and tail
$f_r(h, t)$	Scoring function
$\mathcal{L}$	Loss function
d_h, d_t	Description of head and entity
$\{w_1, \ldots, w_m\}$	Text corpus
$\mathbb{R}^d$	d dimensional real-valued space
$\mathbb{C}^d$	d dimensional complex-valued space
$\odot$	element-wise multiplication

B. KG EMBEDDING WITH FACTS

Then some representative embedding techniques which only depend on facts are introduced, much previous work has been studied on such techniques. They proved to be state-of-the-art and are widely used as the basis of embedding task with the textual information. These work could catch the structure information of the KG and provide the structure-based embeddings. Entities and relations are directly represented as the real-valued vector, matrix or complex-valued vector. Scoring functions are defined to assess the validity of facts. We divide them into two groups according to the scoring function.

1) TRANSLATION-BASED MODEL

TransE [10] projects the entity and relation into the same vector space. Inspired by [35], TransE follows the translation of the entity embeddings and takes relations as operations. Particularly, given a fact (h, r, t), it is represented as $\mathbf{r}_s \approx \mathbf{h}_s - \mathbf{t}_s$. Intuitionally, to measure the validity to facts, let the score of the fact equal to the distance between $\mathbf{h}_s + \mathbf{r}_s$ and $\mathbf{t}_s$, i.e.,

$$f_r(h, t) = -\|\mathbf{h}_s + \mathbf{r}_s - \mathbf{t}_s\|_{1/2}. \tag{1}$$

If (h, r, t) is observed in KG, $f_r(h, t)$ should obtain a high score. TransE is proved to be concise and effective. However, it assigns a unique embedding to each entity no matter which different triples it constitutes. It is indicated that TransE is poor at dealing with the multi-relation problems. To handle the 1-to-N, N-to-1 and N-to-N relations, TransH [13] introduces a hyperplane to represent different relations between entities. h and t are projected onto the relation-specific hyperplane with the normal vector $\mathbf{w}_r$. The scoring function of TransH follows TransE defined as

$$f_r(h,t) = -\|(\mathbf{h}_s - \mathbf{w}_r^\top \mathbf{h}_s \mathbf{w}_r) + \mathbf{r}_s - (\mathbf{t}_s - \mathbf{w}_r^\top \mathbf{t}_s \mathbf{w}_r)\|_{1/2}. \quad (2)$$

Each structure-based relation in TransR has an associated space. Entity embeddings are first projected in the relation-specific space with the projection matrix $\mathbf{M}_r \in \mathbb{R}^{d\times d}$. Then the translation is performed in the relation space as

$$f_r(h,t) = -\|\mathbf{M}_r \mathbf{h}_s + \mathbf{r}_s - \mathbf{M}_r \mathbf{t}_s\|_2^2. \quad (3)$$

2) SEMANTIC MATCHING MODEL

RESCAL [21] models the interaction between the entity pair through representing the relation as a weighted matrix $\mathbf{M}_r$. The scoring function is based on the bilinear formulation, i.e.,

$$f_r(h,t) = \mathbf{h}_s^\top \mathbf{M}_r \mathbf{t}_s. \quad (4)$$

To simplify the model, $\mathbf{M}_r$ is restricted to be diagonal in DISTMULT [37]. ComplEx [38] introduces complex vector space and represents the entities and relations with the complex-value embeddings, i.e., $\mathbf{h}_s, \mathbf{r}_s, \mathbf{t}_s \in \mathbb{C}^d$. The scoring function is the extension of DISTMULT, i.e.,

$$f_r(h,t) = Re(\mathbf{h}_s^\top diag(\mathbf{M}_r)\overline{\mathbf{t}_s}), \quad (5)$$

where $Re(\cdot)$ keeps the real part in the complex. $\overline{\mathbf{t}_s}$ is the conjugating operation.

Neural Tensor Network (NTN) [36] introduces the neural network architecture. Vectors of pair entity are given as the input. Next in the hidden layer, a bilinear tensor $\mathbf{M}_r^{[1:d]} \in \mathbb{R}^{d\times d\times d}$ is used to combine the entities $h, t \in \mathbb{R}^d$ across different relations. The score is given by the linear output, which reflects the possibility of the relation. The model can be represented as:

$$f_r(h,t) = \mathbf{r}_s^\top tanh(\mathbf{h}_s^\top \mathbf{M}_r^{[1:d]} \mathbf{t}_s + \mathbf{W}_r(\mathbf{h}_s \oplus \mathbf{t}_s) + \mathbf{b}_r), \quad (6)$$

where $\mathbf{b}_r \in \mathbb{R}^d$ is the bia and $\mathbf{W}_r \in \mathbb{R}^{d\times d}$ denotes relation-specific weight matrices.

C. TEXTUAL INFORMATION

Rich textual information is available for KG embedding. In this section, different kinds of textual information are introduced. Especially the entity description and textual mentions are relatively precise and labeled texts. However, in most cases, the textual information is noisy and has weak relevance with the KG. It is hard to build the representation directly from the textual information. Some preprocessing methods are described as well based on the type of the information.

1) RAW TEXTS

Raw texts or unlabeled texts could contain much semantic information and they are easy to collect, e.g., a news corpus [39], Wikipedia articles. And they are expected to have a high coverage of entities. However, these texts don't have direct and close associations with the KG. Entities appear at random positions in document with an unknown title. Relations are implicit and entities are always buried in the noise. The wanted information are hard to derived from the raw texts, which makes it tough to represent the entities and relations accurately. Wang *et al.* [26] traverse the corpus and assume each distinct pair words in the window context and the implicit relation between them is a candidate fact. Entity name and description are used to align the candidate fact with the facts in KG. Some KG-related tools are also available for preprocessing these texts, i.e., entity linking. An *et al.* [40] link the Wikipedia inner-link and a corresponding entity in Freebase if they share the titles and link a word in corpus as a WordNet entity if the word belongs to its synsets. Wang *et al.* [24] also annotate the corpus by labeling the entities in KG as the FIGURE 2 illustrated. Wu *et al.* [42] extract all the sentence containing the entity name in the corpus as the reference sentences of the corresponding entity. General tools [41] are used but they are not able to completely remove the noises in the raw texts.

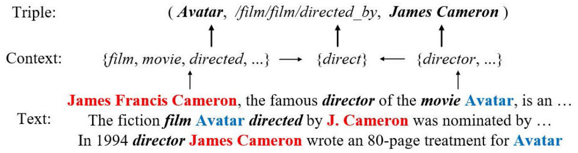
Triple:
(*Avatar*, */film/film/directed_by*, *James Cameron*)
Context:
{*film*, *movie*, *directed*, ...} → {*direct*} ← {*director*, ...}
Text:
James Francis Cameron, the famous *director* of the *movie* Avatar, is an …
The fiction *film* Avatar *directed* by J. Cameron was nominated by …
In 1994 *director* James Cameron wrote an 80-page treatment for Avatar

FIGURE 2. Label the entity in text corpus [24].

2) TEXTUAL MENTIONS

Textual mentions refer to the individual sentence containing the entity pairs which are derived from ClueWeb. The sentence is processed with the dependency parser and represented as lexicalized dependency paths in [43] and [44]. Then the path, which contains words and dependency arcs, is defined as the textual relation between the entity pair. An instance of textual relation is illustrated in FIGURE 3.

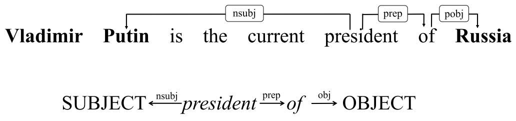
nsubj
prep
pobj
Vladimir Putin is the current president of Russia
SUBJECT ←nsubj— president —prep→ of —obj→ OBJECT

FIGURE 3. Textual relation connecting the entity pair [44].

However, the textual mentions are built to express relationship between entity pairs and inevitably introduces the noisy information. For example, the sentence "Miami Dolphins in 1966 and the Cincinnati Bengals in 1968" does not express any relationship. An *et al.* [40] aims to provided precise textual mentions for the following relation representation learning. An extractor is used to collect precise textual mentions for each fact (h, r, t). All the sentences possessing h and t are collected as the candidate textual mentions at the begining. The sentence is kept as the accurate textual mentions only if it meets one of the conditions: (i) containing the hyponym/synonyms word of r, (ii) containing the similar words with relation names.

3) ENTITY DESCRIPTION

In most KGs, e.g., Freebase and Wikidata, there are precise descriptions or definitions belong to entities. Entity description is considered to have strong association with the KG. As a result, entity descriptions are promising texts for entity representation and are popular in the most of the text-based representing models [25], [27], [40]. Classical textual features like TF-IDF could be used to extract keywords from the entity description. The flaw of entity description is that not every entity has the associated description in KG. Other textual information like entity name is also considered though they have little semantic information.

III. TEXT-BASED KG EMBEDDING

Most text-based KG embedding techniques utilize the textual information to extend the existing techniques with facts alone and represent the entity or relation with text-based and structure-based embedding. In recent years, reseaches that represent the entity and relation with textual information alone [46], [47] start to appear owing to the expressive encoders. We make a review on these text-based methods and find out that they typically cover the following three key elements: (i) building the text-based representations of entites and relations, (ii) defining a scoring function containing the text-based represenations, and (iii) training the encoding models and making it compatible to the facts. In this section we introduce the methods from two views: encoding models and scoring functions. The details of training procedure are presented in section V.

A. ENCODING MODELS

Particularly, the representation is constructed based on the word embeddings via the encoding models. Different encoders and mechanism are proposed to learn the expressive representation from the textual information. We further category them in linear models, convolutional neural networks, recurrent neural networks, topic models and transformer.

1) LINEAR MODELS

Simple linear models can be used to generate the text-based representation. Description-embodied knowledge representation learning (DKRL) [25] and Joint(BOW) [48] proposed continuous bag-of-words (CBOW) to encode the keywords extracted for the entity description. The vectors of keywords are summed up as the text-based entity representation. The limitation of CBOW is that it treats all the keywords equally and neglects the word order in description.

Veira *et al.* [49] make use of the Wikipedia entry for the entity name as the entity description and generate the relation-specific weighted word vectors (WWV) for the entity. WWV endows different importance to the words in description. The importance is decided depending on the frequency and the relationship with the relation of the word. Matrix A is used to record the number of occurrences in the description for each word and matrix B for the relevance. Given an entity e_i with relation r_j and entity description, the text-based representation is defined as

$$\mathbf{e}_i^{(rj)} = \frac{(\mathbf{A}_i \odot \mathbf{B}_j)W}{\|\mathbf{A}_i \odot \mathbf{B}_j\|}, \tag{7}$$

where W is a $n_w \times d$ word vectors matrix. Moreover, WWV introduces an auxiliary matrix P to decrease the number of parameters to be estimated, refers to parameter-efficient weighted word vectors (PE-WWV). P is composed of the word feature of the relations and the relevance between r_i and word w_k is defined as $P_i W_k^\top$. So e_i with r_j can be expressed as

$$\mathbf{e}_i^{(rj)} = \frac{\sum\limits_{w_k \in text(e_i)} A_{ij} exp(\mathbf{P}_i \mathbf{W}_k^\top)\mathbf{W}_k}{\sum\limits_{w_k \in text(e_i)} A_{ij} exp(\mathbf{P}_i \mathbf{W}_k^\top)}. \tag{8}$$

WWV and PE-WWV are much more expressive than the CBOW, but they still failed to exploit the word order.

2) CONVOLUTIONAL NEURAL NETWORKS

The deep neural networks could have impressive performance on encoding the corresponding entity description and textual mentions. Convolutional neural network (CNN) is an efficient and effective model for the researches in computer vision and natural language processing. It can be used to learn deep expressive features in the textual information.

DKRL(CNN) [25] assumed that the word orders in entity description may involve the implicit relations between entities that the KG omits and can be learned through the neural network models [50]. It constructs a five-layer CNN to fully discover the implicit relation in the word order. Except stop words, the vector embedding of the words in the description are regarded as input. Max-pooling operation is used for reducing the scale of the parameter space of CNN and filtering noises. After that, mean-pooling operation preserves the textual information for the entity embedding. The non-linear output layer builds the text-based entity representation. The architecture of DKRL(CNN) is illustrated in FIGURE 4. He *et al.* [51] assume that the descriptions of relations could also supply much semantic information and proposed Relation Text-embodied Knowledge Representation Learning (RTKRL). It adapts the similar CNN architecture and takes the position feature into consideration to represent the relation by encoding the fine-grained descriptions. Lexicalized dependency paths represent the textual relation between the entities. Synonymous textual relations share the similar paths consisting of similar patterns, words and dependency arcs. To use the statistical strength and learn the Compositional Representations of Textual Relations (CONV),

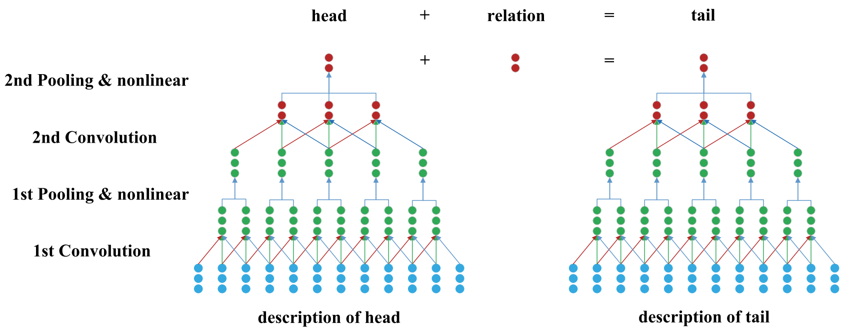

FIGURE 4. Architecture of the CNN in DKRL [25].

Toutanova *et al.* [44] construct a one-hidden-layer CNN to encode their internal structure. Words and dependency arcs in the paths are projected to the input layer as word embedding $v \in \mathbb{R}^d$ with an embedding matrix V. The text-based relation representation is as follows

$$\mathbf{r}_d = max\{\mathbf{h}|\mathbf{h}_i = tanh(\mathbf{W}_i^0\mathbf{v}_i + \mathbf{W}_i^1\mathbf{v}_{i+1} + \mathbf{W}_i^2\mathbf{v}_{i+2} + \mathbf{b})\}, \quad (9)$$

where $\mathbf{h}_i$ denotes the i-th vector of hidden layer, $\mathbf{W}_i^0, \mathbf{W}_i^1, \mathbf{W}_i^2 \in \mathbb{R}^{d\times d}$ are the position-specific maps and $\mathbf{b}$ bias the vector.

However, the textual relation probably exists in multiple textual mentions containing the same entity pair and it is essential to figure out which better expresses the relation. Tang *et al.* [43] propose Multi-source Knowledge Representation Learning (MKRL), introducing the position embedding and attention mechanism [52] in CNN to encode the lexicalized dependency paths extracted from the textual mentions. For the i^{th} word in the path, position embedding x_{ih} and x_{it} is defined as the relative distance to the head entity and the tail entity from the word. Then the position embedding and the word embedding constitute the concatenated embedding (x_i, x_{ih}, x_{it}) as the input of the encoder. To build the final representation from the related textual mentions $\{s_1, \ldots, s_n\}$, sentence-level attention mechanism is used to filter noise and preserve information. For each output representation of single sentence $\mathbf{s}_i$ after the max-pooling operation, the correpsonding structure-based relation embedding $\mathbf{r}$ is defined as the attention

$$att_i = \frac{exp(\mathbf{s}_i\mathbf{Mr})}{\sum_n exp(\mathbf{s}_n\mathbf{Mr})}. \quad (10)$$

Then the text-based relation representation is as follow

$$\mathbf{r}_d = \sum_{i=1}^{n} \frac{att_i \cdot \mathbf{s}_i}{\sum_j^n att_j}. \quad (11)$$

3) RECURRENT NEURAL NETWORKS

Recurrent neural networks are utilized to capture long-term relational dependency. Wu *et al.* [42] propose the Sequential Text-embodied Knowledge Representation Learning (STKRL). STKRL first extracts the reference sentences for each entity from corpus and regards the entity representation as the multi-instance learning problem. Given an entity e, position-based RNN/LSTM encoder is used to get the set of sentence-level representations $\{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$. Position features of words constitute the entity name are marked as 0. The words around the name are marked based on the relative distance and the left side has negative values and the right side has the positive. The extracted sentence expresses the entity differently. So the corresponding structure-based embedding $\mathbf{e}_s$ is used to assign different importance for $\mathbf{s}_i$ by calculating the cosine similarity $att(\mathbf{s}_i, \mathbf{e}_s)$ between them. The text-based representation of e is as follow:

$$\mathbf{e}_d = \sum_{i=1}^{m} \frac{att(\mathbf{s}_i, \mathbf{e}_s) \cdot \mathbf{s}_i}{\sum_{i=1}^{m} att(\mathbf{s}_i, \mathbf{e}_s)}. \quad (12)$$

Wang *et al.* [29] propose Entity Descriptions-Guided Embedding (EDGE) to encode the entity description. BiLSTM is introduced in EDGE to well handle the context and word sequence in the entity description. The pre-trained embeddings of words in description released by word2vec are given as the input of BiLSTM. However, the word embeddings are not directly input into the encoder. Considering entity description could enhance the structure-based embedding, vice versa. EDGE aims to refine the word embedding with structure-based embedding iteratively. For the word/phrase w_i in description, if exists an entity e_i with the same name. w_i is updated as

$$\mathbf{w}_i = \frac{\sum_{e_j\in r(e_i)} \beta_{ij}\mathbf{e}_j + \alpha_i\mathbf{e}_i}{\sum_{e_j\in r(e_i)} \beta_{ij} + \alpha_i}, \quad (13)$$

where α_i, β_{ij} are adjustable values, $e_j \in r(e_i)$ denotes the neighbors of e_i in KG with different relation edge.

Although different measures are taken with the RNN/LSTM to enhance the text-based representation, they fails to handle the multi-relation problem and represent the same entities or relations in different triples with a unique text-based representation. Xu *et al.* [48] take the both directions of word sequence into consideration to fully discover the semantic information in the word orders. They utilize BiLSTM to represent the entity with its description, referred to as Joint(BiLSTM). The output vectors are concated at each step. Particularly, given the output at the i^{th} position $\mathbf{z}_i \in \mathbb{R}^d$

of BiLSTM. The text-based entity representation is defined as

$$\mathbf{e} = \sum_{i=1}^{n} \mathbf{z}_i. \tag{14}$$

Considering that words in the description make different contribution to the entity representation given different relations. Some words are the keywords for one relation, but not for another. For example, the "*parentOf*″ relation will emphasize on social relations and gender attributes of the person. Based on Joint(LSTM), word-level attention mechanism is applied in BiLSTM to do the relation-specific encoding of the entity description in different contexts, referred to as Joint (A-LSTM). Given a structure-based representation of relation $\mathbf{r}_s$ and entity description D with size n, the attention of r is defined as

$$\begin{aligned} e_i(r) &= \mathbf{v}_a^\top \tanh(\mathbf{W}_a\mathbf{z}_i + \mathbf{U}_a\mathbf{r}), \\ \alpha_i(r) &= \frac{exp(e_i(r))}{\sum_{j=1}^{n} exp(e_j(r))}, \end{aligned} \tag{15}$$

where $\mathbf{W_s} \in \mathbb{R}^{d\times d}$, $\mathbf{v}_a^\top \in \mathbb{R}^d$ are parameters. Such that, for each entity e related to r, the text-based representation is defined as

$$\mathbf{e}_r = \sum_{i=1}^{n} \alpha_i(r) * \mathbf{z}_i. \tag{16}$$

An *et al.* [40] employ BiLSTM to encode the entity descriptions and triple-specific relation mentions to learn accurate text-based representations, referred to as ATE, and introduce mutual attention between the entities and relations, referred to as AATE. Particularly, the mutual attention has two phrases. Given the relation mention $\{w_1, w_2, \ldots, w_n\}$, at the first stage, the representation of the textual relation $\mathbf{r}' \in \mathbb{R}^d$ generated by averaging the hidden vectors of BiLSTM is used as attention to infer the entity representation $\mathbf{e}_d$.

$$\begin{aligned} \alpha_i(\mathbf{r}') &= \frac{\exp(e_i(\mathbf{r}'))}{\sum_j^n \exp(e_j(\mathbf{r}'))}, \\ e_i(\mathbf{r}') &= \mathbf{h}_i^\top \mathbf{W}_e \mathbf{r}', \end{aligned} \tag{17}$$

where $\mathbf{h}_i \in \mathbb{R}^h$ is the hidden representation of w_i. Then the text-based entity representation is as follow:

$$\mathbf{e}_d = \tanh(\sum_i^n \alpha_i(\mathbf{r}')\mathbf{h}_i). \tag{18}$$

At the second stage, $\mathbf{h}_d + \mathbf{t}_d$ is used to infer the attention representation $\mathbf{r}_d$, following the same formulation. It is proved that, with the mutual attention, AATE could better handle the multi-relation problem than ATE in the experiment.

4) TOPIC MODELS

The aforementioned methods focus on catching the word-level or sentence-level information in the texts but neglect the latent topics of entities. Ouyang *et al.* [53] propose entity topic based representation learning (ETRL) and Xiao *et al.* [54] propose the Semantic Space Project (SSP). They utilize the latent topics in entity description to represent the entities and relations. Particularly, entity descriptions are treated as documents and topic model NMF is used to learn the text-based representations

$$\mathcal{L}_{nmf} = \sum_{i=1}^{j=n} \sum_{j=1}^{m} \|M_{i,j} - \mathbf{v}_{e_i}\mathbf{s}_{w_j}^\top\|_2^2, \tag{19}$$

where $M_{i,j}$ denotes the frequency of word w_j existing in the description of entity e_i, $\mathbf{v}_{e_i}$ and $\mathbf{s}_{w_j}$ are their topic representation respectively. The topic relation representation is defined as the average of the entities in ETRL.

5) TRANSFORMER

Bidirectional Encoder Representations from Transformer (BERT) [45] is a state-of-the-art pre-trained contextual language representation model. With such encoder, Yao *et al.* propose KG-BERT [46] to encode the entity description and relation name. Entities, relations and triples are represented as the sequence of words constitute their names or descriptions. The sequences are treated as input tokens Tok. [CLS] is a classification token and the first token of the sentence. Token [SEP] divides the sentences of entities and relations into different segments. Then token, segment and position embedding constitute the input representation E. The final hidden state C is used as the aggregate sequence representation for the triple. The architecture of KG-BERT is illustrated in FIGURE 5. KG-BERT discards the structure-based representation and the scoring function is as follow, $W \in \mathbb{R}^{2\times h}$ is the classification layer weights.

$$f_r(h, t) = sigmoid(CW'). \tag{20}$$

All the mentioned encoding models and related textual information are listed in TABLE 2.

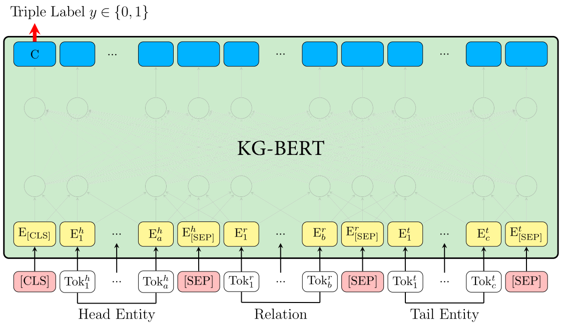

FIGURE 5. Illustration of KG-BERT for measuring the fitness of the fact [46].

B. CONSTRUCT THE SCORING FUNCTION

Apart from KG-BERT [46], other methods extend the existing embedding techniques. To measure the plausibility of the facts from two different prospects and learn the representations simutaneously, the scoring function is made up by the structure-based and text-based representation.

TABLE 2. Encoding models and textual information of text-based KG embedding.

Category	Method	Encoding models	Textual Information
Linear models	CBOW [25] [48]	sum of the keywords vectors	entity description
	WWV [49]	relation-specific weighted of word vectors	entity description
CNNs	DKRL(CNN) [25]	CNN+max pooling+mean pooling	entity description
	CONV [44]	single-layer CNN+max pooling	textual mention
	AATE [40]	position-based CNN+word-level attention	entity description &textual mention
	RTKRL [51]	position-based CNN+max pooling+mean pooling	textual mention
RNNs	STKRL [42]	LSTM+weighted sentence	raw texts
	Joint(LSTM) [48]	BiLSTM	entity description
	Joint(A-LSTM) [48]	BiLSTM+word-level attention	entity description
	EDGE [29]	BiLSTM+max pooling+knowledge constraint	entity description
	MKRL [43]	BiLSTM+mutual word-level attention	textual mention
Topic model	ETRL [53], SSP [54]	NMF	entity description
Transformer	KG-BERT [46]	BERT	name or description

Considering text-based representation in single method can be applied to multiple models, we concentrate on the combination of the both representations in scoring function, rather than the formulation.

1) REPLACE THE ASSOCIATED REPRESENTATIONS

Replacement is a simple and effective way to construct the extended scoring function. Reference [49] replaced the structure-based entity representation in TransE with the corresponding text-based one, the scoring function is as follow

$$f_r(h, t) = -\|\mathbf{h}_d^{(r_s)} - \mathbf{r}_s + \mathbf{t}_d^{(r_s)}\|. \tag{21}$$

Not limited to this, the text-based representation is applied to other techniques, such as TransR, RESCAL as well. Contrary to directly subsituting the representation in single formulation, [25] and [42] built multiple formulas for different replacements. Given the fact (h, r, t), the scoring function is the amount of the formulas:

$$\begin{aligned} f_r(h, t) &= f_S + f_D, \\ f_D &= f_{DD} + f_{DS} + f_{SD}. \end{aligned} \tag{22}$$

DKRL extends TransE, so that $f_S = \|\mathbf{h}_s + \mathbf{r} - \mathbf{t}_s\|$, $f_{DD} = \|\mathbf{h}_d + \mathbf{r} - \mathbf{t}_d\|$, $f_{DS} = \|\mathbf{h}_d + \mathbf{r} - \mathbf{t}_s\|$ and $f_{SD}(h, t) = \|\mathbf{h}_d + \mathbf{r} - \mathbf{t}_d\|$. Structure-based and text-based representation share the same relation embedding, resulting the mutual promotion between the two types of representation. Instead of TransE, the text-based embedding is mapped on the relation-specific hyperplane and relation-specific space before being injected to scoring function of TransH and TransR in RDRL [55].

2) UNIFY THE REPRESENTATION

Some combination mechanisms are proposed to integrate the both representations into a single one. Given an entity e, gate mechanism is used to combine the e_s and e_d. Joint(BiLSTM) applies a linear interposition between e_s, e_d and use a real-value vector g to control how much the joint representation depends on structural or textual information. Particularly,

$$\mathbf{e} = \mathbf{g}_e \odot \mathbf{e}_s + (1 - \mathbf{g}_e) \odot \mathbf{e}_d, \tag{23}$$

where $\mathbf{g}_e$ is the gate to balance e_s and e_d and its elements are in [0, 1], $\odot$ is an entry-wise multiplication. Unified represenaions are applied for TransE as

$$\begin{aligned} f_r(h, t) = \|&(\mathbf{g}_h \odot \mathbf{h}_s + (1 - \mathbf{g}_h) \odot \mathbf{h}_d) \\ &+\mathbf{r} - (\mathbf{g}_t \odot \mathbf{h}_s + (1 - \mathbf{g}_t) \odot \mathbf{t}_d)\|_2^2. \end{aligned} \tag{24}$$

The construction of Joint(BiLSTM) is illustrated in FIGURE 6. The gate mechanism only considers the entity representation projected in the real-valued vector space. An *et al.* [40] make a more comprehensive combination mechanism, a weight factor $\alpha \in [0, 1]$ is used to integrate the embeddings.

$$\begin{aligned} Re(\mathbf{h}) &= \alpha \cdot Re(\mathbf{h}_s) + (1 - \alpha) \cdot \mathbf{h}_d, \\ Re(\mathbf{r}) &= \alpha \cdot Re(\mathbf{r}_s) + (1 - \alpha) \cdot \mathbf{r}_d, \\ Re(\mathbf{t}) &= \alpha \cdot Re(\mathbf{t}_s) + (1 - \alpha) \cdot \mathbf{t}_d, \end{aligned} \tag{25}$$

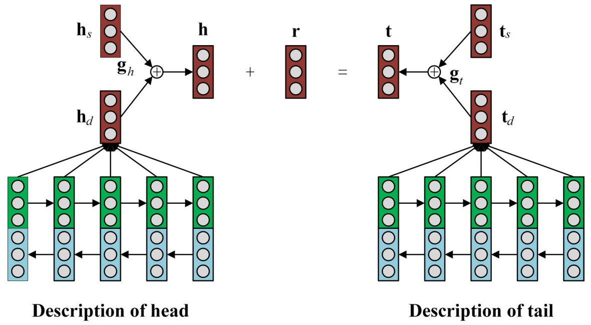

FIGURE 6. Unify the text-based and structure-based representation with gate mechanism and construct the scoring function in Joint(BiLSTM).

TABLE 3. Summary of the embeddings, extended models and combination mechanism in text-based KG embedding.

Models	Structure-based embedding	Text-based embedding	Extended model	Combination mechanism
CONV [44]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{r}_d \in \mathbb{R}^d$	DISTMULT	$f_r(h,t) = \mathbf{r}_s^\top(\mathbf{h}_s \circ \mathbf{t}_s) + \mathbf{r}_d^\top(\mathbf{h}_s \circ \mathbf{t}_s)$
WWV [49]	$\mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{h}_d^{(r_s)}, \mathbf{t}_d^{(r_s)} \in \mathbb{R}^d$	SE [58]/TransE/ TransR/RESCAL/ DISTMULT/HOLE	$f_r(h,t) = -\Vert\mathbf{h}_d^{(r_s)} - \mathbf{r}_s + \mathbf{t}_d^{(r_s)}\Vert$
DKRL [25]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{h}_d, \mathbf{t}_d \in \mathbb{R}^d$	TransE	$f_r(h,t) = f_S + f_D$
RTKRL [51]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{r}_d \in \mathbb{R}^d$	TransE	
RDRL [54]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{h}_d, \mathbf{t}_d \in \mathbb{R}^d$	TransR	
STKRL [42]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{h}_d, \mathbf{t}_d \in \mathbb{R}^d$	TransE	
Joint [48]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{h}_d, \mathbf{t}_d \in \mathbb{R}^d$	TransE	$\mathbf{h} = \mathbf{g}_h \odot \mathbf{h}_s + (1 - \mathbf{g}_h) \odot \mathbf{h}_d$ $\mathbf{t} = \mathbf{g}_t \odot \mathbf{t}_s + (1 - \mathbf{g}_t) \odot \mathbf{t}_d$ $\mathbf{r} = \mathbf{r}_s$
EDGE [29]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{h}_d, \mathbf{t}_d \in \mathbb{R}^d$	TransE	
AATE [40]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{h}_d, \mathbf{t}_d \in \mathbb{R}^d, \mathbf{r}_d \in \mathbb{R}^d$	TransE/TransH/ TransR/ComplEx	$Re(\mathbf{h}) = \alpha \cdot Re(\mathbf{h}_s) + (1-\alpha) \cdot \mathbf{h}_d$ $Re(\mathbf{r}) = \alpha \cdot Re(\mathbf{r}_s) + (1-\alpha) \cdot \mathbf{r}_d$ $Re(\mathbf{t}) = \alpha \cdot Re(\mathbf{t}_s) + (1-\alpha) \cdot \mathbf{t}_d$
SSP [54]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{s} \in \mathbb{R}^d$	TransE+NMF	$-\lambda\Vert(\mathbf{d} - \mathbf{s}^\top\mathbf{d}\mathbf{s})\Vert_2^2 + \Vert\mathbf{d}\vert_2^2$ $\mathbf{d} = \mathbf{h}_s + \mathbf{r}_s - \mathbf{t}_s$
ETRL [53]	$\mathbf{h}_s, \mathbf{t}_s, \mathbf{r}_s \in \mathbb{R}^d$	$\mathbf{h}_d, \mathbf{t}_d \in \mathbb{R}^d, \mathbf{r}_d \in \mathbb{R}^d$	TransE+NMF	$\mathbf{h} = \mathbf{M}_e\mathbf{h}_d + \mathbf{h}_s$ $\mathbf{t} = \mathbf{M}_e\mathbf{t}_d + \mathbf{t}_s$ $\mathbf{r} = \mathbf{M}_r\mathbf{r}_d + \mathbf{r}_s$

where $Re(\cdot)$ denotes the real part vector of the representation. If the structure-based relation representation is matrix, it is treated as a vector with each element the same as the element in diagonal matrix. Simple linear transformations can also be used to unify the embedding. In this way, [53] defines the joint representation as

$$\begin{aligned} \mathbf{h} &= \mathbf{M}_e\mathbf{h}_d + \mathbf{h}_s, \\ \mathbf{t} &= \mathbf{M}_e\mathbf{t}_d + \mathbf{t}_s, \\ \mathbf{r} &= \mathbf{M}_r\mathbf{r}_d + \mathbf{r}_s. \end{aligned} \tag{26}$$

3) OTHERS

SSP provides an idea that executes the embedding progress in a semantic subspace by modeling the association between the KG facts and textual information. The architecture is similar to TransH. But SSP measures the plausibility of the facts by projecting $\mathbf{h} + \mathbf{r} - \mathbf{t}$ onto the semantic-specific hyperplane, rather than a relation-specific hyperplane. In this way, the facts and the corresponding texts are interacted, so semantic relationships and textual contexts are able to be used to contribute the knowledge graph embedding. Let $\mathbf{d} \approx \mathbf{h} + \mathbf{r} - \mathbf{t}$ and $\mathbf{s}$ denotes the loss vector and normal vector respectively, such that the scoring function in SSP is defined as

$$f_r(h, t) = -\lambda\Vert(\mathbf{d} - \mathbf{s}^\top\mathbf{d}\mathbf{s})\Vert + \Vert\mathbf{d}\Vert_2^2. \tag{27}$$

The embeddings, extended models are listed in TABLE 3, and they are categorized by the combination mechanism when constructing the scoring function.

IV. TEXT-IMPROVED KG EMBEDDING

The difference between text-improved and the text-based KG embedding techniques is that the former does not cover the three key elements and focus on enhancing the structure-based KG embedding with the textual information. We roughly category the methods based on the usage of the texts.

A. INITIALIZE THE ENTITY EMBEDDING

The entity embeddings are usually initialized randomly in the existing embedding approaches. Some methods improve KG embedding by initializing the embeddings with textual information. NTN [36] and DISTMULT [37] learn the distribute embeddings of the words which constitute the entity name. Then the vectors are averaged and used to intialize the entity embedding. However, the entity name contains little semantic information. Long *et al.* [58] follow NTN but replace the entity name with the entity description. Particularly, for each entity e, given an associated fragment of textual information D, e_d is defined as

$$\mathbf{e}_d = \frac{1}{|D|}\sum_{i=1}^{|D|}\mathbf{w}_i, \tag{28}$$

where $\mathbf{w}_i$ denotes the embedding of the word in texts. NTN learns the word embedding in an unlabeled news corpus, DISTMULT makes use of the pre-trained embedding released by Word2Vec. Word2vec and GloVe in [58] are trained on a large corpus which contains many entities from FreeBase. DISTMULT introduced another method which treats the entity as the word/phrase in corpus and learns the distributed embedding of the entity for initializing directly.

The experiments of all these methods prove that initialization with the textual embedding is useful in the knowledge graph embedding. However, not all the initialization with the word embedding leads to improvement on performance in the downstream application. In DISTMULT, the initialization with the average word vectors in entity name is observed performance drops on the link prediction task. For more than 73% entities in datasets are non-compositional phrases like person names, locations and films which are not suitable to represent the entity.

B. AUGMENT THE STRUCTURE-BASED KG EMBEDDING

To incorporate the textual information, some work augment the entity and relation embedding in the existing models, specifically, they represent the entity or relation as the linear transformation of the embedding of textual information. To capture the implicit relationship between entities and attributes, FeatureSum [49] utilizes the unstructured corpus and uses word2vec model to embed entity name that in the same context with close word vectors, generating the associated word vectors $\mathbf{e}_d$. A transformation is designed in FeatureSum

$$\hat{\mathbf{e}} = \mathbf{e} + \mathbf{e}_d\mathbf{M}, \tag{29}$$

where $\mathbf{M}$ is used to map to the vector space of $\mathbf{e}$, which is regardless of the relation type. Sun *et al.* [56] concatenate the pre-trained vectors of entity description released by the Doc2Vec models [59], i.e., DM and DBOW, as the description embedding. The entity representation is defined as

$$\hat{\mathbf{e}} = \sigma(\mathbf{e} + \mathbf{e}_d\mathbf{M}_r), \tag{30}$$

where $\mathbf{M}_r$ is a relation-specific weight matrix for description embeddings.

The producted word and description embedding are unique for the entity in various facts made up of different relations, which is a negative impacting on handling the mutli-relation problem. Wang *et al.* [24] presented a text-enhanced KG embedding method (TEKE). For better coping with the 1-to-N, N-to-1 and N-to-N problems, entity representations are augmented with the corresponding textual context embedding. Additionally, TEKE defines the joint contexts of the pair entity as the textual context embedding of each relation between them. Such that, relations between different entities or the same entity with different contexts will have distant representations. TEKE transforms the Wikipedia text corpus to a co-occurrence network. The network is constructed by the words and labeled entites. Each entity is treated as a node and the words in the textual context are defined as the neighbor nodes $n(e)$ of it. Co-occurrence frequency y is the connection edge between them in the context and threshold is used to remove the noise. The textual context embedding of the entity is defined as follow

$$\mathbf{n}(e) = \frac{1}{\sum_{x_e^i \in n(e)} y_e^i} \sum_{x_e^i \in n(e)} y_e^i \mathbf{x}_e^i. \tag{31}$$

Given two nodes h, t, the textual context of relation is defined as the intersection of $n(h)$ and $n(t)$. Associated embedding is defined as the weighted average as well. The augmented representations of the existing embedding techniques, such as TransE, are as follow

$$\begin{aligned} \hat{\mathbf{h}} &= \mathbf{W}_e\mathbf{h}_d + \mathbf{b}_h, \\ \hat{\mathbf{t}} &= \mathbf{W}_e\mathbf{t}_d + \mathbf{b}_t, \\ \hat{\mathbf{r}} &= \mathbf{W}_r\mathbf{n}(h,t) + \mathbf{b}_r, \end{aligned} \tag{32}$$

where $\mathbf{W}_e$, $\mathbf{W}_r$ are weight matrices, and $\mathbf{b}_h$, $\mathbf{b}_t$, $\mathbf{b}_r$ bia vectors. The extension on entity representation is available for the existing models except ComplEx. The extension on relation can apply for TransH and TransR.

C. JOINT EMBEDDING OF THE TEXTS AND FACTS

Joint embedding aims to project the textual information and structural knowledge into the same continous vector space for improving the structure-based embedding. Specifically, these methods represent and score the facts with the existing models. Moreover, they model the textual information and make it interactive with the entities and relations.

Wang *et al.* [26] first present the method of jointly learning (Jointly) the KG embedding and word embedding by aligning the facts and the words in raw texts. Knowledge model, text model and alignment model are the components of Jointly. The knowledge model follows TransE to score the facts and designs a conditional likelihood loss $\mathcal{L}_K$ to learn the general KG embedding. The text model defines the scoring function to measure the plausibility of the two words w and v co-occurring in the context.

$$f(w, v) = \mathbf{b} - \frac{1}{2}\|\mathbf{w} - \mathbf{v}\|^2. \tag{33}$$

Based on the function, $\mathcal{L}_A$ is designed to measuring the overall fitness of the word pairs. Alignment models are designed to guarantee the entity embedding and the word embedding projected in the same vector space. Various alignment texts are used: entity names, Wikipedia anchors and entity description. Particularly, given the fact (h, r, t) and word pair (w, v), alignment model generates new triples and pairs like (h, r, w), (h, v) depending on whether the words are in the alignment texts. The total plausibility is measured by the loss $\mathcal{L}_A$. Finally, [26], [27] learn the embeddings by maximizing

$$\mathcal{L} = \mathcal{L}_K + \mathcal{L}_T + \mathcal{L}_A. \tag{34}$$

The alignment by entity description has been proved better than the other two alignment mechanisms in the experiment.

RLKB [28] proposes a method of jointly embedding the entities, relations and words in entity descriptions in the same vector space. Following Jointly, RLKB designs the $\mathcal{L}_K$ based on TransE to measure the fitness of facts. Then entity description is made interactive with the entities. Given the set of keywords $\{w_1, w_2, \ldots, w_n\}$ extracted from the descritpion of entity e, RKLB forces the entity embedding close to the

TABLE 4. Mechanism and textual information of text-improved KG embedding.

Category	Method	Mechanism	Textual Information
Initialization	NTN [36]	mean of the word embedding	entity name
	DISTMULT [37]	treating entities as word/phrase in corpus	raw texts
	Teng et al. [58]	mean of the word embedding	entity description
Augmentation	TEKE [24]	textual context embedding	raw texts
	FeatureSum [48]	treating entities as word/phrase in corpus	raw texts
	DEKE [56]	concatenation of DM and DBOW vectors	entity description
Joint embedding	Jointly [26] [27]	alignment texts and facts	raw texts/anchor/name
	RKLB [28]	close to word embedding	entity description
	JointE+SATT [39]	CNN+max pooling+mutual attention	textual mention

embeddings of the keywords on Euclidean Distance.

$$\Theta(d_e\|e) = \beta - \sum_{j=1}^{n} \frac{1}{2}\|\mathbf{e} - \mathbf{w}_j\|_2^2, \tag{35}$$

where $\mathcal{L}_D$ is defined to measure the total distance. Embedding is learned by maximizing the loss

$$\mathcal{L} = \mathcal{L}_K + \mathcal{L}_D. \tag{36}$$

Jointly and RLKB perform the linguistic analysis on textual information, which is not powerful enough to catch the important features. The words, entities and relations are directly embeded into the continous vectors, failing to make use of the excellent feature in texts and facts. Han *et al.* [39] introduce mutual attention mechanism between the knowledge model and text model to filter the noise in sentences and obtain more discriminative KG embeddings, referred to as JointE+SATT. Given the set of sentences $\pi_{r_s} = \{s_1, s_2, \ldots, s_m\}$ containing the associated entites (h, t) and textual relation r_s, a position-based CNN is used to encode each sentence in text model. To represent r_s, latent relation $\mathbf{r}_{ht} = \mathbf{h} - \mathbf{t}$ is defined as the attention over the output embedding. Scoring function $\mathbf{o} = \mathbf{M}\mathbf{r}_s$ is defined to measure the fitness. Given the entity pairs $\phi_r = \{(h_1, t_1), \ldots, (h_n, t_n)\}$ with the common relation r, the knowledge model utilizes $\mathbf{M}_{r_s}$ as the attention over the latent relations. The framework of the mutual attention is illustrated in FIGURE 7. Related facts are scored with the global relation r_k in the formulation of TransE or TransD. Embeddings are learned via the amounted plausibility of the relations, facts and parameters θ.

$$\mathcal{L} = \mathcal{L}_K + \mathcal{L}_A + \lambda\|\theta\|_2, \tag{37}$$

where λ is the harmonic factor. Different mechanisms and textual information are listed in TABLE 4.

V. MODEL TRAINING AND COMPARISION

Two assumptions are widely used in the training procedure of KG embedding: open world assumption (OWA) and closed world assumption (CWA). Under CWA, unobserved facts in the KG are defined false. Several KG embedding works use the CWA. For example, RESCAL [21] is trained under CWA.

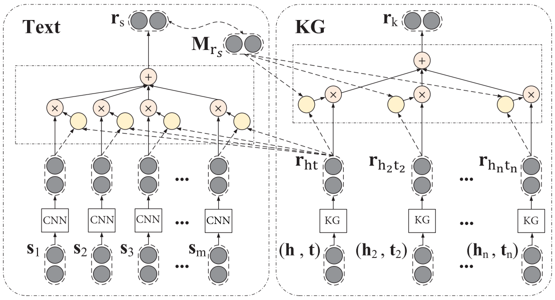

FIGURE 7. The framework of joint embedding with mutual attention [39].

All the aforementioned techniques are trained under the open world assumption. A fact triple that doesn't exist in the KG is defined as unknown whether it is true or false [60]. OWA is much more fit the status of knowledge graph than CWA, for the existing KGs are far from complete and huge amounts of true facts are still missing.

A. LOSS FUNCTION

Some favored loss functions are introduced for the model optimization. To make true facts have higher scores, most methods select the margin-based pairwise ranking loss to learn the text-based and structure-based KG embedding.

$$\mathcal{L} = \sum_{(h,r,t)\in F^+} \sum_{(h',r',t')\in F^-} \max(0, \lambda + f_r(h,t) - f_r'(h',t')), \tag{38}$$

where F^+ denotes the set of positive samples and F^- the negatives. (h, r, t) is the positive sample in F^+ and (h', r', t') is the negative one in F^-. Higher score is assigned for the true fact than the false one, and through maxizing the margin between (h, r, t) and (h', r', t'), embeddings are learned.

The joint embedding techniques favor to use the likelihood-based loss function, aiming to measure the plausibility of every element in the triple.

$$\mathcal{L} = \sum_{(h,r,t)\in G} \log P(h, r, t),$$
$$\log P(h, r, t) = \log P(h|r, t) + \log P(r|h, t) + \log P(t|r, h), \tag{39}$$

where $P(h|r,t)$ is the conditional probability of head h when given r and t.

$$P(h|r,t) = \frac{exp\{f_r(h,t)\}}{\sum_{h'\in E} exp\{f_r(h',t)\}}. \tag{40}$$

The same formulation is defined for $P(r|h,t)$ and $P(t|h,r)$.

B. NEGATIVE SAMPLING

Negative samples can be generated through corrupting the true facts. A simple method is to replace the head entity or tail entity of a positive fact $(h,r,t) \in F^+$ with an entity sampled randomly from the entity set E [10]. Samples follow the uniform distribution. However, this may lead to too many positive samples in the training. More effective sampling strategies are proposed in the following researches like [13], [61]. Bernoulli sampling performs the replacement with the possbility $\frac{tph}{tph+htp}$ or $\frac{hpt}{tph+htp}$, where tph and hpt denote the average number of tail entities per head entity and the average number of head entities per tail entity respectively. For the textual information learning, entity descriptions are replaced accordingly during the training. Some of the introduced models treat the entities and relations as the word/phrase in the textual information to learn the contexts and expressions of them. The training procedure needs to be modified according to their own definitions or assumptions. In Literature [26], when word w and context word v are considered a positive fact (w,v) with implict relation. Distinct word $\tilde{w}$ is sampled uniformly from the textual information to form negative samples $(\tilde{w},v)$ which are never concurrent in the textual information.

Jointly [26] and RKLB [28] use negative sampling to simplify the loss function. The likelihood-based loss is tough to do the normalization for the millions of normalizers and the large number of candidate-entities influence the efficiency of training phrase. So negative sampling strategy [35] is used to transform the loss objective and reduce the number of candidate entities. A sampled subset of negative candidates from the corresponding negative distributions P_{neg} and the probability of a positive triple is defined as $P(1|h,r,t)$ and false as $P(0|h',r,t)$. $\log P(h|r,t)$ in (41) is guaranteed to approximate the following equation and replaced with it.

$$\log P(1|h,r,t) + \sum_{i=1}^{n} \mathbb{E}_{\tau^- \sim P_{neg}}[P(0|h',r,t)], \tag{41}$$

where

$$P(1|h,r,t) = \frac{1}{1+exp\{-f_r(h,t)\}},$$
$$P(0|h',r,t) = \frac{1}{1+exp\{f_r(h',t)\}}. \tag{42}$$

The same transformation is applied to $\log P(r|h,t)$ and $\log P(t|h,r)$.

C. DATASET

FB15K [10] and WN18 [10] are the most widely used datasets for the knowledge graph model training. But for models required entity descriptions, they still need to be refined. Entities which have too few descriptive words or even no words and the triples those entities comprise are removed from FB15K. The same manipulation is conducted in WN18. New Dataset is also created for the model training and testing, i.e., DBpedia500k [47]. Some downstream tasks usually need task-specific datasets. FB20K [25] is generated for the open-world KG completion in which one of the entities in test triples are unseen in the KG. FB15k-237-OWE is generated based on the FB15k-237 [62] in which redundant inverse relations have been removed and the entity descriptions are restricted in 5 words on average.

D. MODEL COMPARISON

TABLE 5 compares the parameters and prerequisites of aforementioned techniques, except KG-BERT [46] and MKRL [43]. The former does not share the details of model and the latter introduces entity type as auxiliary information besides texts. n_e, n_r, and n_w denote the size of entity set, relation set and the vocabulary respectively. p is the dimensionality of the position embedding and k is the size of node in hidden layer. All the models are trained under the open world assumption. Prerequisites are the techniques used to do the preprocessing or pre-training, in which we do not consider the parameters. Parameters of different models are calculated under the conditions that the structure-based embedding is provided by TransE, except Literature [36], [37] and [44]. We could make several conclusions on the parameters of the different models. First, initialization is apparently the most parameter efficient way to enhancement the KG embedding and the improvement is based on the suitable textual information. Second, most text-improved techniques have less parameters than the text-based. For they rarely apply complicated mechanism to cope with the textual information and incoporate it with the existing models. Comparing with TABLE 4, more expressive encoding models have more parameters in text-based embedding. On the other hand, models based on raw texts would have more parameters than those based on labeled texts. DNN models can't be applied to cope with the raw texts directly. Preprocessing methods, i.e, entity linking and distant supervision are demanded to generate textual mentions or annotate the raw texts. However, these work may label noisy data and force the DNNs to introduce additional mechanisms, e.g. attention mechanism which also results in more parameters. The models work on the entity description usually has the least parameters, for the entity descriptions are precise and concise in KGs.

We next discuss the performance of these methods and restrict the scene in link prediction task with two datasets, i.e., WordNet and Freebase data. Intuitionally, models with more mechanisms have more parameters and lead to better performance. However, based on the same textual information, the models with more parameters do not necessarily perform better. For example, Joint(A-LSTM) is slightly worse than the Joint(LSTM) in WordNet. The reason is that the number of relations is too small for Joint(A-LSTM) to take

TABLE 5. Parameters and prerequisites of different models.

Method	Parameters	Prerequisites
CBOW [25] [48]	$n_e d + n_r d + n_w d$	TF-IDF
WWV [49]	$n_w d + n_r n_w$	–
PE-WWV [49]	$n_w d + n_r d$	–
DKRL(CNN) [25]	$n_e d + n_r d + n_w d + 2kd$	–
CONV [44]	$n_e d + n_r d + n_w d + kd$	distant supvervision
AATE [40]	$n_r d + n_w d + n_e d + 8(kd + k^2 + d) + 2kd$	entity linking, accurate mention
RTKRL [51]	$n_e d + n_r d + n_w d + 2kd$	–
STKRL [42]	$n_e d + n_r d + n_w d + 4(kd + k^2 + d)$	extract reference sentences
Joint(LSTM) [48]	$n_e d + n_r d + n_w d + 4(kd + k^2 + d)$	–
Joint(A-LSTM) [48]	$n_e d + n_r d + n_w d + 4(kd + k^2 + d) + 2d^2 + 2d$	–
EDGE [29]	$n_e d + n_r d + n_w d + 4(kd + k^2 + d)$	–
ETRL [53], SSP [54]	$2n_e d + n_r d + n_w d$	NMF
NTN [36]	$n_e d + n_r(d^3 + 2d^2 + 2d)$	word distribution
DISTMULT [37]	$n_e d + n_r d$	word2vec
Teng et al. [58]	$n_e d + n_r d$	GloVe, word2vec
TEKE [24]	$n_e d + n_w d + 2d^2$	word2vec, Wikify
FeatureSum [48]	$n_e d + n_r d + d^2$	word2vec
DEKE [56]	$n_e d + n_r(d + d^2)$	DM, DBOW
Jointly [26]	$n_e d + n_r d + 2n_w d$	–
RKLB [28]	$n_e d + n_r d$	word2vec
JointE+SATT [39]	$n_e d + n_r d + (n_w + k)(d + 2p) + d^2$	entity linking

relation as attention. In [53], the CNN architecture in DKRL is reimplemented and the extended model is replaced with TransR. Better performance is observed in the experiment, which indicates more expressive composition can be helpful to performance. In most cases, the combination between structure-based and text-embedding embedding should have better performance than only using one of them. However, the Hits@10 of PE-WWV has different results in WordNet and Freebase. In WordNet, structure-based embedding provided by TransE and TransR performs better than PE-WWV while in Freebase the PE-WWV performs better [49]. The reason is that there is no common words between the definition of the pair of entities. Similar case happens to TEKE as well for the models have more parameters and needs more rounds to converge.

TABLE 3 compares structure-based and text-based embedding and extended models of text-based KG embedding. The ''Extended Models'' column lists the existing embedding techniques used in the experiment section. Apparently, TransE is the most popular model used to catch the structural information. Most models choose to combine the representations into a joint one or consititue a hybrid scoring function. It is a conundrum to determine which way is best to construct the scoring function, for there are no related comparisons are found to our best knowledge.

VI. APPLICATIONS IN KG-RELATED TASKS

After reviewing the current approaches that utilize textual information in KG embedding. We aim to show some applications that the learned entitiy and relation embeddings via such techniques can be applied to. Moreover, we simply review some embedding techniques with texts aiming at these specific applications. The applications are restricted to be relevant to the knowledge graphs, including KG completion task, i.e., link prediction, triple classification and entity classification as well as out-of-KG tasks, i.e., entity alignment, relation extraction and recommender system.

A. LINK PREDICTION

Link prediction is to search an entity which probably constructs a new fact with another given entity and specific relation. For KGs are always imperfect, link prediction aims to discover and add missing knowledge into it. With the existing relation and entity, candidate entities are selected to form a new fact. The task has been experimented in much previous studies [10], [13]. Given $(?, r, t)$ is to predict h or given $(h, r, ?)$ to predict t is a common form where $h \in E$, $r \in R$, $t \in E$, and $(h, r, t) \notin G$. For instance, *(?, satellite of, Earth)* is to predict what goes around the Earth and *(Moon, satellite of, ?)* to predict what object the Moon moves around. It's easy for the learned entity and relation embedding to do such prediction, so long as treating it as a ranking procedure. Through putting the embedding candidate entities into the scoring function $f_r(h, t)$, the answer of the prediction could be obtained by ranking the score of the candidate facts. As a result, it can be handled by all the KG embedding method. The evaluation criterions, such as mean rank, Hits@n (rank proportion smaller than n + 1), and AUC-PR, can be designed based on the ranks.

1) ZERO-SHOT SETTING

In zero-shot scenario, restrictions on entity are relaxed. The setting introduces the set of entities E^i that are out of KG. So the task is formulated as given $(?, r, t)$, $r \in R$, $t \in E \cup E^i$ to predict $h \in E \cup E^i$ or given $(h, r, ?)$, $r \in R$, $h \in E \cup E^i$ to predict $t \in E \cup E^i$ and $(h, r, t) \notin G$. The out-of-KG entities do not carry any instructure information, so KG embedding on basis of facts alone fails to cope with the task, for they rely heavily on the structure information and connectivity in KG. However, the text-based embedding is still available if they have asscociated textual information [25], [26]. The link prediction in zero-shot scenario is also called open-world KG completion in [47].

Recently, [64] introduces an Open-World Extension (OWE) to make traditional KG embedding models to perform link prediction in zero-shot scenario. OWE utilizes entity description to supply the missing structure information. Particularly, the text-based entity embedding is defined as the mean of the vectors and mapped to the vector space of the structure-based embedding. A linear function, an affine function and a 4-layer Multi-Layer Perceptron are introduced. The transformations are trained through the loss function:

$$L(\Theta) = \sum_{k=1}^{m} \|\Psi_{\Theta}^{map}(\mathbf{e}_d) - \mathbf{e}_s\|_2, \qquad (43)$$

where $\mathbf{e}_s$ is released by the pre-trained KG embedding model, Θ denotes the parameters in the transformation models. Suppose that the extended model is TransE and head entity h is out-of KG, the task is performed based on:

$$f_r(h, t) = -\|\Psi^{map}(\mathbf{h_d}), \mathbf{r}, \mathbf{t}\|_2. \qquad (44)$$

ConMask [47] performs the embedding-based entity prediction task solely based on the entity description, name and relation name. Given a fact (h, r, t), ConMask built its text-based represention with relationship-dependent content masking, semantic averaging and target fusion. The masking highlights the relevant words by computing the similarity between each word in description and words constitute the relationship name. The weight of the i^{th} word in the entity description w_i is the largest score among the $(i - k : i)th$ word embeddings. Semantic averaging gets the mean vector of names. Target fusion then extracts the entity embedding with a convolutional neural network (FCN). Normalized text-based embedding is used to measure the fitness of the fact.

$$f_r(h, t) = ConMask(h, r, t). \qquad (45)$$

The overall framework of ConMask is illustrated in FIGURE 8.

Additionally, the link prediction in zero-shot scenario could handle the poor scalability of KG [63], because the KG can be extended with the out-of-KG entities.

B. ENTITY CLASSIFICATION

Entity classification is a multi-label classification task and targets to predict the types of the entity. For each entity e, it has multiple types in KGs. The embeddings of entity are learned by the models and input into the classifier, i.e., Logistic Regression as features. In the zero-shot scenario, the unseen entity is represented by its text-based embedding [25]. For evaluation, mean averaged precision(MAE) is used.

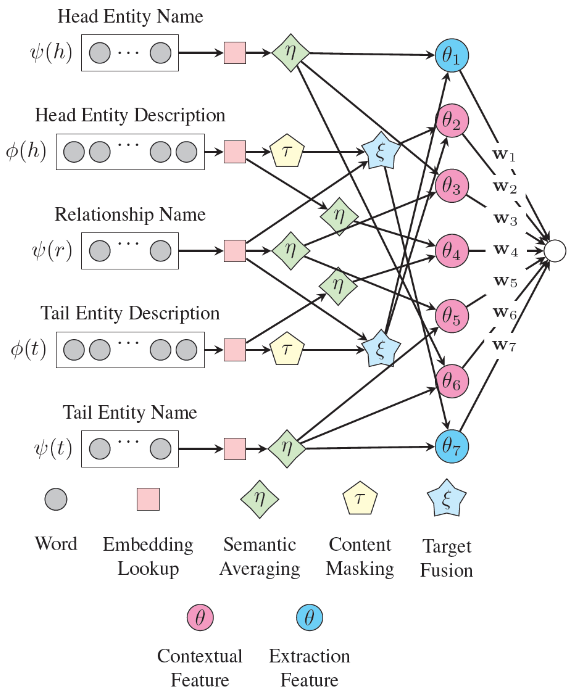

FIGURE 8. Illustration of the ConMask model [47].

C. TRIPLE CLASSIFICATION

Triple classification is to verify whether unseen facts are true or not in testing data, which is typically regarded as a binary classification problem. Scoring function can be used to assign a score for each triple and the decision rule is a specific threshold. Aforementioned embedding methods could be applied for triple classification. Ranking metrics and the micro- and macro-averaged accuracy can be used to evaluate the task.

D. ENTITY ALIGNMENT

Entity alignment (EA) aims to identify pair entities with an equivalent relation in heterogeneous KGs. Given two different KGs G_1 and G_2, E_1 and E_2 respectively denote corresponding set of entities. EA aligns the entities, such that $e_1 \in E_1$ and $e_2 \in E_2$ into an entity pair (e_1, e_2) where e_1 is equivalent to e_2. In practice, a small set of alignment entities (i.e., Equal entities in different knowledge graphs) is given to start the alignment process. Embedding-based alignment calculates the similarity between embeddings of a pair of entities [65]. Recently, the embedding models are leveraged to alignment the entities in multilingual KGs [66]. The application is novel but challenging. For the crosslingual is far from complete and is lack of the supervision, which will influence the performance of cross-lingual inferences.

KDCoE [67] first conducts co-training on structure-based KG embedding and text-based embedding to perform the alignment between entities of multilingual KGs. The method is composed of multilingual KG embedding Model (KGEM) and multilingual entity Description Embedding Model (DEM). KGEM is modified from the previous embedding model MTransE-LT [66]. TransE and a linear transformation alignment model are introduced to jointly learn the cross-lingual inferences. In DEM, a gated recurrent unit (GRU) encoder incorporated with the self-attention mechanism in the layer is applied to learn the semantic feature in multilingual entity descriptions. Pre-trained word embeddings released by cross-lingual Bilbowa [68] is given as the input of the encoder. The text-based embedding is obtained through mapping the averaged output vectors

$$\mathbf{e}_d = \tanh(\mathbf{M}(\frac{1}{|d|}\sum_{i=1}^{|d|}\mathbf{v}_i) + b), \tag{46}$$

where M denotes the affine matrix, d is the description sequence and $\mathbf{v}_i$ is the i^{th} output vector of encoder. The scoring function of the description embedding is as follow

$$f(e, e') \log \sigma(\mathbf{e}^\top \mathbf{e}'), \tag{47}$$

where (e, e') is the entity pair in unordered language pair $I(L, L')$. Finally, KGEM and DEM are iteratively co-trained to propose new ILLs in turn iteratively.

E. RELATION EXTRACTION

Relation extraction [69] aims to extract the latent relationship that connects the entities in the raw texts to help build the KGs automatically. Although relation extraction is extensively studied in NLP tasks, due to the lack of labeled relational data, much researches utilize KG in a distant supervision way [70], [71] [72], also referred as weak supervision or self supervision, use heuristic matching to create training data by assuming that sentences containing the same entity mentions may express the same relation under the supervision of a relational database.

Riedel [11] jointly embed the textual information and KG into the same matrix to do the relation extraction. Entity pairs are represented as the row of the matrix, and the textual mentions or relations in facts are represented as the column. The value of entry depends on the condition where entity pairs appear in the textual mention or form a fact with a observed relation in KG. If meeting the condition, the value is set to 1, otherwise it is set to 0. In the training procedure, entity pairs are along with textual mention and KG relationship at the same time, the latter is distant supervision. Relation extraction is performed through predicting the missing relations given the textual mentions and entity pair. Fan *et al.* [73] use the text features to replace the textual mentions at the first columns in a similar matrix.

F. RECOMMENDER SYSTEM

Recommender systems provide users with suggestions about what they may want to buy or inspect. Collaborative filtering technology is widely used in various recommendation systems and achieved great success. It models the interaction between users and projects (as the product of potential representation). However, because the interaction between users projects may be very sparse, such technologies do not always work well.

Zhang *et al.* [4] improve the performance of collaborative filtering by introducing a hybrid recommendation framework which utilizes the facts, textual information and visual information in KG. Given a item, TransR is used to obtain the structure-based entity representation, stacked de-noising auto-encoder is used to generate the text-based entity representation and stacked convolutional auto-encoders are applied to catch visual-based representation. For each item j, η_j is defined as the original latent vector for j. After incorporating the representations from KG, the latent vector is redefined as

$$\mathbf{e}_j = \mathbf{s}_j + \mathbf{t}_j + \mathbf{v}_j + \eta_j, \tag{48}$$

where $\mathbf{s}_j, \mathbf{t}_j$ and $\mathbf{v}_j$ are the associated structure-based representation, text-based representation and visual-based representation respectively. Given the latent vector u_i of user i, the preference between them is represented as $\mathbf{u}_i^\top \mathbf{e}_j$. The representations are learned and proved to be effective in the recommender system.

VII. CONCLUSION AND FUTURE DIRECTIONS

Much work has been done to handle the sparseness of KG and enhance the performance of embedding with the textual information. The text-based KG embedding methods introduce entity and relation representations built from the entity description and textual mention. Low frequency or poor connectivity of entities and relations slightly influence text-based representation. Various powerful encoding models, i.e., linear models, deep neural networks and topic models are proposed for the representation. Attention mechanism is used to deal with the noise in text and multi-relation problem. Scoring functions are constructed by the structure-based and text-based representations or the unified representations. The text-improved KG embedding help fact-based embedding techniques with different mechanisms. Pre-processed word vectors are used to initialize or augment the KG embedding. Embedding models and text models are made interactive and trained simultaneously, so that the KG embedding can be improved by the texts. Both the text-based and text-improved methods are proved effective. In addition, utilizing textual information in KG embedding can handle zero-shot scenario KG completion and many other tasks.

Future directions of utilizing textual information in KG embedding. **(i) Incorporation with the latest embedding models:** The reviewed techniques mainly incorporate the textual information in TransE, TransR, and TransH. Recently, a lot of achievements have emerged in the field of KG embedding with facts, e.g., ConvKB [74], RotatE [75], and CrossE [76]. Latest encoders, i.e., capsule networks and Graph Neural Networks and new characteristics, i.e., relation patterns and interaction matrix are introduced to enhance the performance of KG embedding. Nevertheless, such techniques still suffer from the imperfection of the KG. To extend the newly proposed models with the textual information is an interesting research direction. **(ii) Scalability of KG:** Scalability is an important attribute of large-scale KGs containing millions of entities and relations. To deal with the noise in the texts and multi-relation problem, existing methods simply improve the expressiveness of the model but ignore the computational efficiency, which could have a negative impact on scalability. So that a balance between the model expressiveness and computation cost should be studied. **(iii) Open-world KG Completion:** Novel entities and relations are increasing and added into the KGs every day. Embedding techniques based on the observed facts in KG alone failed to learn the representations of the newcomers. As mentioned before, the problem can be tackled by introducing the text-based representation. However, except DKRL, ConMask, and OWE, few works try to solve the task. For the structural information is no longer available for the new entities, expressive encoding models used to generate the precise embedding from texts are necessary. In addition, the above three achievements only consider entity prediction but fail to predict the out-of-KG relations.

Challenges on utilizing the textual information. **(i) Coverage on the entities:** Currently, text-based KG embedding rely heavily on entity description which do not have a high coverage on entities like raw texts. The experiments are usually conducted with processed datasets in which each entity owns a description. In reality, the condition is not perfect and the encoding models might be useless. Even the text corpus used in text-improved methods can not be guaranteed that it contains all the entities to be embedded. **(ii) High-frequency entities:** Contrary to the sparseness, high-frequency entities own relatively rich structural information. On the basis of the connectivity, the embeddings can be well-trained by the techniques with facts alone. The text-related mechanisms that are helpful to low-frequency entities might damage the well-learned embeddings. Consequently, the overall performance of the model is influenced. **(iii) Latent relations in texts:** Relations are explicit in facts but implicit in texts. Even in the textual mentions, to definitely represent the relation between the entity pair is formidable. Most text-based and text-improved methods choose to only target the entities. The rest either treat the representations of the pair entity as replacement or introduce complicated process procedure. The former is not precise enough and the latter is inefficient.

As a result, there are still unsolved problems and valuable researches worth exploring. We expect the survey can facilitate future work on the KG embedding with textual information.

REFERENCES

[1] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 697–706.

[2] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 1247–1250.

[3] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia—A large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web*, vol. 6, no. 2, pp. 167–195, 2015.

[4] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 353–362.

[5] A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," 2014, *arXiv:1406.3676*. [Online]. Available: http://arxiv.org/abs/1406.3676

[6] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W. Y. Ma, "Open question answering with weakly supervised embedding models," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2014, pp. 165–180.

[7] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on Freebase from question-answer pairs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1533–1544.

[8] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "Joint learning of words and meaning representations for open-text semantic parsing," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, 2012, pp. 127–135.

[9] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes, "Improving efficiency and accuracy in multilingual entity extraction," in *Proc. 9th Int. Conf. Semantic Syst. (I-SEMANTICS)*, 2013, pp. 121–124.

[10] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2787–2795.

[11] S. Riedel, L. Yao, A. Mccallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 74–84.

[12] M. Nickel, V. Tresp, and H.-P. Kriegel, "Factorizing YAGO: Scalable machine learning for linked data," in *Proc. 21st Int. Conf. World Wide Web (WWW)*, 2012, pp. 271–280.

[13] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1112–1119.

[14] M. Fan, Q. Zhou, E. Chang, and T. F. Zheng, "Transition-based knowledge graph embedding with relational mapping properties," in *Proc. 28th Pacific Asia Conf. Lang. Inf. Comput.*, 2014, pp. 328–337.

[15] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2181–2187.

[16] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 687–696.

[17] G. Ji, K. Liu, S. He, and J. Zhao, "Knowledge graph completion with adaptive sparse transfer matrix," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 985–991.

[18] H. Xiao, M. Huang, and X. Zhu, "From one point to a manifold: Knowledge graph embedding for precise link prediction," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1315–1321.

[19] J. Feng, M. Huang, M. Wang, M. Zhou, Y. Hao, and X. Zhu, "Knowledge graph embedding by flexible translation," in *Proc. 15th Int. Conf. Princ. Knowl. Represent. Reasoning*, 2015, pp. 557–560.

[20] A. Bordes, X. Glorot, J. Weston, and Y. Bengio, "A semantic matching energy function for learning with multi-relational data," *Mach. Learn.*, vol. 94, no. 2, pp. 233–259, Feb. 2014.

[21] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 809–816.

[22] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1955–1961.

[23] H. Liu, Y. Wu, and Y. Yang, "Analogical inference for multi-relational embeddings," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2168–2178.

[24] Z. Wang, J. Li, Z. Liu, and J. Tang "Text-enhanced representation learning for knowledge graph," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1293–1299.

[25] R. Xie, Z. Liu, and M. Sun, "Representation learning of knowledge graphs with hierarchical types," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2965–2971.

[26] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph and text jointly embedding," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1591–1601.

[27] H. Zhong, J. Zhang, Z. Wang, H. Wan, and Z. Chen, "Aligning knowledge and text embeddings by entity descriptions," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 267–272.

[28] M. Fan, Q. Zhou, T. F. Zheng, and R. Grishman, "Distributed representation learning for knowledge graphs with entity descriptions," *Pattern Recognit. Lett.*, vol. 93, pp. 31–37, Jul. 2017.

[29] W. Zhou, S. Wang, and C. Jiang, "Knowledge graph embedding with interactive guidance from entity descriptions," *IEEE Access*, vol. 7, pp. 156686–156693, 2019.

[30] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proc. IEEE*, vol. 104, no. 1, pp. 11–33, Dec. 2016.

[31] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, vol. 8, no. 3, pp. 489–508, Dec. 2016.

[32] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.

[33] Y. Lin, X. Han, R. Xie, Z. Liu, and M. Sun, "Knowledge representation learning: A quantitative review," 2018, *arXiv:1812.10901*. [Online]. Available: http://arxiv.org/abs/1812.10901

[34] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition and applications," 2020, *arXiv:2002.00388*. [Online]. Available: http://arxiv.org/abs/2002.00388

[35] T. Mikolov, W.-T. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 746–751.

[36] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 926–934.

[37] B. Yang, W.-T. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–12.

[38] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 2071–2080.

[39] X. Han, Z. Liu, and M. Sun, "Neural knowledge acquisition via mutual attention between knowledge graph and text," in *Proc. 32th AAAI Conf. Artif. Intell.*, 2018, pp. 4832–4839.

[40] B. An, B. Chen, X. Han, and L. Sun, "Accurate text-enhanced knowledge graph representation learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 745–755.

[41] R. Mihalcea and A. Csomai, "Wikify!: Linking documents to encyclopedic knowledge," in *Proc. 16th ACM Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2007, pp. 233–242.

[42] J. Wu, R. Xie, Z. Liu, and M. Sun, "Knowledge representation via joint learning of sequential text and knowledge graphs," 2016, *arXiv:1609.07075*. [Online]. Available: http://arxiv.org/abs/1609.07075

[43] X. Tang, L. Chen, J. Cui, and B. Wei, "Knowledge representation learning with entity descriptions, hierarchical types, and textual relations," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 809–822, May 2019.

[44] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1499–1509.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[46] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for knowledge graph completion," 2019, *arXiv:1909.03193*. [Online]. Available: http://arxiv.org/abs/1909.03193

[47] B. Shi and T. Weninger, "Open-world knowledge graph completion," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2019, pp. 1957–1964.

[48] J. Xu, X. Qiu, K. Chen, and X. Huang, "Knowledge graph representation with jointly structural and textual encoding," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1318–1324.

[49] N. Veira, B. Keng, K. Padmanabhan, and A. Veneris, "Unsupervised embedding enhancements of knowledge graphs using textual associations," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 5218–5225.

[50] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. 25th Int. Joint Conf. Comput. Linguisitcs*, 2014, pp. 2335–2344.

[51] M. He, X. Du, and B. Wang, "Representation learning of knowledge graphs via fine-grained relation description combinations," *IEEE Access*, vol. 7, pp. 26466–26473, 2019.

[52] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[53] X. Ouyang, Y. Yang, L. He, Q. Chen, and J. Zhang, "Representation learning with entity topics for knowledge graphs," in *Proc. Int. Conf. Knowl. Sci., Eng. Manage.*, 2014, pp. 534–542.

[54] S. Qin, N. Wang, H. Wang, L. Zhou, and H. Zhan, "EHP: Entity hyperplane projection for knowledge graph embedding with entity descriptions," in *Proc. Comput., Commun. IoT Appl. (ComComAp)*, Oct. 2019, pp. 3104–3110.

[55] S. Dai, Y. Liang, S. Liu, Y. Wang, W. Shao, X. Lin, and X. Feng, "Learning entity and relation embeddings with entity description for knowledge graph completion," in *Proc. 2nd Int. Conf. Artif. Intell., Technol. Appl. (ICAITA)*, 2018, pp. 202–205.

[56] X. Sun, Y. Man, Y. Zhao, J. He, and N. Liu, "Incorporating description embeddings into medical knowledge graphs representation learning," in *Proc. Int. Conf. Hum. Centered Comput.*, 2018, pp. 188–194.

[57] A. Bordes, J. Weston, R. Collobert, and Y. Bengio, "Learning structured embeddings of knowledge bases," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 301–306.

[58] T. Long, R. Lowe, J. C. K. Cheung, and D. Precup, "Leveraging lexical resources for learning entity embeddings in multi-relational data," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 112–117.

[59] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

[60] L. Drumond, S. Rendle, and L. Schmidt-Thieme, "Predicting RDF triples in incomplete knowledge bases with tensor factorization," in *Proc. 27th Annu. ACM Symp. Appl. Comput. (SAC)*, 2012, pp. 326–331.

[61] D. Krompaß, S. Baier, and V. Tresp, "Type-constrained representation learning in knowledge graphs," in *Proc. 14th Int. Semantic Web Conf.*, 2015, pp. 57–66.

[62] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proc. 3rd Workshop Continuous Vector Space Models Compositionality*, 2015, pp. 640–655.

[63] J. Ding, S. Ma, W. Jia, and M. Guo, "Jointly modeling structural and textual representation for knowledge graph completion in zero-shot scenario," in *Proc. APWeb WAIM Joint Int. Conf. Web Big Data*, 2018, pp. 369–384.

[64] H. Shah, J. Villmow, A. Ulges, U. Schwanecke, and F. Shafait, "An open-world extension to knowledge graph completion models," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2019, pp. 3044–3051.

[65] H. Zhu, R. Xie, Z. Liu, and M. Sun, "Iterative entity alignment via joint knowledge embeddings," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 4258–4264.

[66] M. Chen, Y. Tian, M. Yang, and C. Zaniolo, "Multilingual knowledge graph embeddings for cross-lingual knowledge alignment," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1511–1517.

[67] M. Chen, Y. Tian, K.-W. Chang, S. Skiena, and C. Zaniolo, "Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment," 2018, *arXiv:1806.06478*. [Online]. Available: http://arxiv.org/abs/1806.06478

[68] S. Gouws, Y. Bengio, and G. Corrado, "Bilbowa: Fast bilingual distributed representations without word alignments," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 748–756.

[69] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics*, 2011, pp. 541–550.

[70] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. Joint Conf. 47th Annu. Meeting ACL, 4th Int. Joint Conf. Natural Lang. Process. (AFNLP)*, 2009, pp. 1003–1011.

[71] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2010, pp. 148–163.

[72] X. Jiang, Q. Wang, P. Li, and B. Wang, "Relation extraction with multi instance multi-label convolutional neural networks," in *Proc. Int. Conf. Computional Linguistics, Tech. Papers*, 2016, pp. 1471–1480.

[73] M. Fan, D. Zhao, Q. Zhou, Z. Liu, T. F. Zheng, and E. Y. Chang, "Distant supervision for relation extraction with matrix completion," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 839–849.

[74] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 327–333.

[75] Z. Sun, Z. H. Deng, J. Y. Nie, and J. Tang, "RotatE: Knowledge graph embedding by relational rotation in complex space," in *Proc. Int. Conf. Learn. Representations.*, 2018, pp. 1–18.

[76] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, and H. Chen, "Interaction embeddings for prediction and explanation in knowledge graphs," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, Jan. 2019, pp. 96–104.

FENGYUAN LU received the B.S. degree from the Department of Computer Science and Technology, Southeast University, Nanjing, China, in 2018. He is currently pursuing the master's degree with the School of Computer Science and Technology, East China Normal University, Shanghai, China. His current research interests are in the areas of the Internet of Things and knowledge graph embedding.

PEIJIN CONG received the B.S. degree from the Department of Computer Science and Technology, East China Normal University, Shanghai, China, in 2016, where she is currently pursuing the Ph.D. degree with the School of Computer Science and Technology. Her current research interests are in the areas of mobile device power management, cloud computing, mobile cloud computing, and mobile edge computing.

XINLI HUANG (Member, IEEE) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, in 2007. He is currently an Associate Professor with the Department of Computer Science and Technology, East China Normal University. His research interests include the areas of Internetworking, software defined networking, cloud computing, future networks, and network security.