applied sciences

MDPI

Article

A Co-Embedding Model with Variational Auto-Encoder for Knowledge Graphs

Luodi Xie [1], Huimin Huang [2,*] and Qing Du [3]

[1] School of Computer Science, Sun Yat-Sen University, Guangzhou 510000, China; xield@mail2.sysu.edu.cn
[2] School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou 325000, China
[3] School of Software, South China University of Technology, Guangzhou 510000, China; duqing@scut.edu.cn
* Correspondence: huanghm45@gmail.com

Abstract: Knowledge graph (KG) embedding has been widely studied to obtain low-dimensional representations for entities and relations. It serves as the basis for downstream tasks, such as KG completion and relation extraction. Traditional KG embedding techniques usually represent entities/relations as vectors or tensors, mapping them in different semantic spaces and ignoring the uncertainties. The affinities between entities and relations are ambiguous when they are not embedded in the same latent spaces. In this paper, we incorporate a co-embedding model for KG embedding, which learns low-dimensional representations of both entities and relations in the same semantic space. To address the issue of neglecting uncertainty for KG components, we propose a variational auto-encoder that represents KG components as Gaussian distributions. In addition, compared with previous methods, our method has the advantages of high quality and interpretability. Our experimental results on several benchmark datasets demonstrate our model's superiority over the state-of-the-art baselines.

Keywords: knowledge graph; embedding; variational auto-encoder

check for updates

Citation: Xie, L.; Huang, H.; Du, Q. A Co-Embedding Model with Variational Auto-Encoder for Knowledge Graphs. *Appl. Sci.* **2022**, *12*, 715. https://doi.org/10.3390/app12020715

Academic Editors: Nikos D. Lagaros and Vagelis Plevris

Received: 29 November 2021
Accepted: 6 January 2022
Published: 12 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Knowledge graph (KG) embeddings are low-dimensional representations for entites and relations. This approach can benefit a range of downstream tasks, such as semantic parsing [1,2], knowledge reasoning [3], and question answering [4,5]. Embeddings are supposed to contain semantic information and should be able to deal with multiple linguistic relations.

At present, research on knowledge graph embedding occurs mainly along three main lines. One of these lines of research includes studies based on translation. TransE [6] was the first model to introduce translation-based embedding, which represents entities and relationships in the same space, and regards the relationship vector r as the translation between the head entity vector h and the tail entity vector t, that is, $h + r \approx t$. Since transE cannot handle one-to-many, many-to-one, and many-to-many relationships (1-to-N, N-to-1, N-to-N), TransH [7] is proposed to enable an entity to have different representations when involved in various relations. In the TransR model [8], an entity is a complex of multiple attributes, and different relationships focus on different attributes of the entity. Another line of research includes studies based on semantic matching. RESCAL [9] obtains its latent semantics by using a vector to represent each entity. Each relationship is represented as a matrix that is used to model the interaction of potential relationships. It defines the scoring function of the triple (h, r, t) as a bilinear function. DistMult [10] simplifies RESCAL by restricting the relationship matrix to a diagonal matrix, which greatly improves training efficiency. ComplEx [11] extends DistMult by introducing complex number domain embedding to better model asymmetric relationships. In ComplEx, the embedding of entities and relationships no longer exists in real space, but in complex space. The third line

of research includes studies based on graph convolutional neural networks. ConvE [12] employs a multi-layer convolutional network, which enables expressive feature learning, while remaining highly parameter-efficient. Unlike previous works, which focused on shallow, fast models that can scale to large knowledge graphs, ConvE uses 2D convolution over embeddings and multiple layers of nonlinear features to model KGs. Subsequently, the ConvKB [13] model has been used to explore the global relationships among same-dimensional entries of the entity and relation embeddings. However, neither of them models the interactions between various positions of entities and relations. R-GCN [14] is another convolutional network designed for KBs, generalized from GCN [15] for uni-relational data.

A typical KG embedding technique has two necessary elements: (i) an encoder to generate KG embeddings and (ii) a scoring function to measure plausibility for each fact. Entities are usually represented as vectors in low-dimensional space, whereas relations are represented as an operation between entities, resulting in vectors for translational operations [6] or matrices for linear transformation [16]. By doing so, the embedding of KG components can be used to enhance the performance in many downstream tasks. Despite the success those previous algorithms have achieved, we note that those methods have the following defects(1) the n-dimensional representation of the KG component can be regarded as a single point, neglecting the uncertainties for entities and relations; (2) they represent entities as vectors located in low-dimensional space and relations as an operation between entities [6,16], thus ignoring the affinities between entities and relations as they are embedded in different semantic spaces.

To address the issues mentioned above, we propose a co-embedding model for KG, learning low-dimensional representations for entities and relations in the same semantic space so that the affinities between them can be effectively captured. Moreover, we introduce a variational auto-encoder to infer the representations of KG components as Gaussian distributions. The mean of the distributions indicatesthe position in semantic space, and the variance of the distributions indicates the uncertainty for each KG components.

Compared with previous works that regard relations as an operation between entities, co-embedding of entities and relations in the same semantic space can improve the performance of KG representation. For example, in Freebase, the relation 'Perfession' is used in (El Lissitzky, Perfession, Architect) and (Vlad. Gardin, Perfession, Screen Writer) uses two distinct semantic information categories, corresponding to a scientist and a writer, resulting in the finding that the resulting representations calculated using the two triples are not the same. The co-embedding model embeds entities and relations at the same semantic space, thus providing high-quality embeddings for both of them.

In summary, our contributions in this work are as follows:

1. We propose a co-embedding model for knowledge graphs, which learns low-dimensional representations for KG components, including entities and relations in the same semantic space, as a result of measuring their affinities effectively.
2. To address the issue of neglecting uncertainty, we introduce a variational auto-encoder into our model, which represents KG components as Gaussian distributions. The variational auto-encoder consists of two parts: (1) an inference model to encode KG components into latent vector spaces, (2) a generative model to reconstruct random variables from latent embeddings.
3. We conduct experiments on real-world datasets to evaluate the performance of our model in link prediction. The experimental result demonstrates that our model outperforms the state-of-the-art baselines.

2. Related Work

2.1. Knowledge Graph Representation

Knowledge representation is a technique that aims at learning low-dimensional representations for KG entities and relations, consisting of two critical steps: (1) constructing a

scoring function measuring plausibility for triples, and (2) embedding KG components in continuous vector spaces.

TransE [6], the most representative method in KG embedding, represents entities as vectors and relations as an operations between entities, i.e., $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. The scoring function is defined as the distance between entities and relations in latent space, written as $f_{\mathbf{r}}(\mathbf{h}, \mathbf{t}) = -\parallel \mathbf{h} + \mathbf{r} - \mathbf{t} \parallel_{1/2}$. However, TransE fails to deal with one-to-many, many-to-one, and many-to-many relations [7,8]. For example, given a relation holding two facts $(\mathbf{h}, \mathbf{r}, \mathbf{t1})$ and $(\mathbf{h}, \mathbf{r}, \mathbf{t2})$, we can infer $\mathbf{t1} = \mathbf{t2}$ even though they are totally different entities. To overcome the above defects, Z. Wang, J. Zhang, J. Feng, and Z. Chen proposed TransH [7] to obtain distinct representations for entities when dealing with different relations, by projecting entity representations onto a hyperplane, resulting in a normal vector. e.g., $\mathbf{h}_{\perp} = \mathbf{h} - \mathbf{w}_{\mathbf{r}}^{\top} \cdot \mathbf{h} \cdot \mathbf{w}_{\mathbf{r}}$, with $\mathbf{w}^{\mathbf{r}}$ as the normal vector.

TransE and its extensions represent both entities and relations as deterministic points in vector spaces, ignoring the uncertainty for KG components. To resolve this problem, some recent works have introduced uncertainty into KG embedding by representing KG components as distributions, e.g., KG2E [17], proposed by Shizhu He, Kang Liu, and Guoliang Ji and Jun Zhao, represents both entities and relations as distributions via Gaussian embedding. Inspired by the previous works, we tackle the embedding problem for KG by modeling both entities and relations as Gaussian distributions and representing them in the semantic space to effectively measure the affinities between them.

2.2. Gaussian Embedding

Gaussian embedding [18] is a method to embed word types into the space of Gaussian distributions, and learn the embeddings directly in that space, which represents words not as low-dimensional vectors, but as densities over a latent space, directly representing notions of uncertainty and enabling a richer geometry in the embedded space.

In word representation, embedding an object as a single point can not naturally express uncertainty about the target concepts with which the input may be associated, and the relationships between points are normally measured by distances required to obey the triangle inequality. Point vectors are typically compared via their dot products, cosine-distance, or Euclean distance, none of which provide asymmetric comparisons between objects. To overcome the limitations in representing objects as points, Gaussian embedding is proposed to learn representations in the space of Gaussian distributions, advocating for density-based distributed embeddings.

In Gaussian embedding, we learn both means and variances from data, representing them as densities over a latent space instead of low-dimensional vectors. As Gaussians innately represent uncertainty and have a geometric interpretation as an inclusion between families of ellipses, our method adopts KL divergence between Gaussian distributions to measure the relationship between objects, which is straightforward to calculate.

Mapping to a density provides many advantages, including better capturing uncertainty about a representation and its relationships, providing asymmetric comparisons between objects, which is more effective than dot product or cosine similarity, and which enables more expressive parameterization of decision boundaries.

2.3. Variational Auto-Encoder

Variational Auto-encoders [19], abbreviated as VAEs, are proposed to learn probability distributions of data. A typical VAE model is made up of two computational neural networks, an inference model to encode observations into latent variables and a generative model to decode from latent deterministic representations to random variables. Given a dataset $X = \{x^i\}_{i=1}^N$, the VAE regards data as random numbers generated via two steps: (1) the latent variable z_i is sampled from prior distribution $p_\theta(z_i)$, and (2) the random variable x_i is generated by the conditional distribution $p_\theta(x|z)$, where θ is the prior distribution parameter. Using the stochastic gradient variational Bayes (SGVB) estimator

and reparameterization, we can learn approximate distributions for each data point via the VAE.

In the VAE, we treat the encoder and decoder as a whole and train them at the same time. The goal of training is to maximize the evidence lower bound of the likelihood function. Specifically, we first input random variables (randomly initialized node embedding) to the encoder, obtain the output, and calculate the encoder error, then we use the output of the encoder as the input of the decoder and calculate the reconstruction error of the decoder. The two parts of the errors are added together as the overall error of the network and propagated backward, thus realizing the simultaneous training of the encoder and the decoder.

In recent years, the VAE algorithm and its variations have been studied and applied in many downstream tasks such as semi-supervised classification [20], clustering [21,22], and image generation [23].

3. Notations and Problems

In this section, we introduce the notation used and define our studied problem.

3.1. Notations

In this paper, we define scalars as normal alphabets (e.g., the output dimension of latent variables: D), sets as typeface alphabets (e.g., the set of entities: $\mathcal{E}$), and vectors as lowercase alphabets (e.g., the representation of head entities: $\mathbf{h}$). A triple in KG is denoted by τ, whereas it can be written as $\tau = (\mathbf{h}, \mathbf{r}, \mathbf{t})$. Our main notations are shown in Table 1.

Table 1. Main notations in our paper.

Symbol	Description
$\mathcal{G}$	a knowledge graph
$\mathcal{E}$	set of entities
$\mathcal{R}$	set of relations
$\mathcal{O}$	set of triples
$M = \lvert\mathcal{E}\rvert$	size of entities
$N = \lvert\mathcal{R}\rvert$	size of relations
$W = \lvert\mathcal{O}\rvert$	size of triples
D	dimension of latent variables
$\mathbf{O} \in \mathbb{R}^{W\times 3}$	observed data for triples
$\mathbf{Z}^{\mathcal{E}} \in \mathbb{R}^{M\times D}$	latent representation matrix for entities
$\mathbf{Z}^{\mathcal{R}} \in \mathbb{R}^{N\times D}$	latent representation matrix for relations

Given a knowledge graph $\mathcal{G}$, We represent the set of entities as $\mathcal{E}$ and the set of relations as $\mathcal{R}$, whereas $\mathcal{G}$ can be defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{O})$, where $\mathcal{O}$ is the set of triples denoted as $\tau = (\mathbf{h}, \mathbf{r}, \mathbf{t}), \mathbf{h}, \mathbf{t} \in \mathcal{E}$ and $\mathbf{r} \in \mathcal{R}$.

3.2. Problem Definition

Using the notation mentioned above, we define the problem of co-embedding in KG as follows.

Problem 1. *The Co-embedding Model for KG Embedding. Given a knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{O})$, our goal is to learn the representations of KG components, including entities and relations, in the same semantic space as that of a transformation Ξ.*

$$\mathcal{G} \xrightarrow{\Xi} \mathbf{Z}^{\mathcal{E}}, \mathbf{Z}^{\mathcal{R}}, \tag{1}$$

where $\mathbf{Z}^{\mathcal{E}} \in \mathbb{R}^{M\times D}$ and $\mathbf{Z}^{\mathcal{R}} \in \mathbb{R}^{N\times D}$, respectively. The i-th row vector in $\mathbf{Z}^{\mathcal{E}}$, written as $\mathbf{z}_i^{\mathcal{E}}$, is denoted as the resulting embedding of the i-th entity, and the j-th row vector in $\mathbf{Z}^{\mathcal{R}}$ written as $\mathbf{z}_j^{\mathcal{R}}$ is denoted as the resulting embedding of the j-th relation.

4. Model

To address the issues we mentioned above, we propose the co-embedding model, learning representations for both entities and relations as Gaussian distributions in the same semantic space, as Gaussians innately represent uncertainty. To obtain high-quality Gaussian embeddings for both entities and relations, we introduce VAE into our model, learning the distributions from training triples in KG via a stochastic gradient variational Bayes [19] estimator. We introduce the details in the following subsections.

4.1. Variational Lower Bound

For a KG represented as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{O})$, the embeddings of KG components can be represented as $\mathbf{Z}^{\mathcal{E}}, \mathbf{Z}^{\mathcal{R}}$ in latent spaces. To embed both entities and relations in the same semantic space, we first define the log-likelihood of $\mathcal{O}$, notated as the set of triples in KG, as:

$$\begin{aligned} \log p(\mathbf{O}) &= \log p(\mathbf{H}, \mathbf{R}, \mathbf{T}) \\ &= \log p(\mathbf{H}) + \log p(\mathbf{R}) + \log p(\mathbf{T}) \end{aligned} \tag{2}$$

where $\mathbf{O} \in \mathbb{R}^{W \times 3}$ and $\mathbf{H}$, $\mathbf{R}$, and $\mathbf{T}$ are components in $\mathbf{O}$. The log-likelihood of KG components, represented as $\log p(\mathbf{H})$, $\log p(\mathbf{R})$, and $\log p(\mathbf{T})$, can be derived using the Bayesian algorithm:

$$\begin{aligned} \log p(\mathbf{H}) &= \log \sum_{i=0}^{D} \left\{ p_\theta(\mathbf{H} \mid \mathbf{Z}_i^{\mathcal{E}}) \cdot p_\theta(\mathbf{Z}_i^{\mathcal{E}}) \right\} \\ &= \log \sum_{i=0}^{D} \left\{ p_\theta(\mathbf{Z}_i^{\mathcal{E}} \mid \mathbf{H}) \cdot p_\theta(\mathbf{H}) \right\} \\ &= \log \sum_{i=0}^{D} \left\{ p_\theta(\mathbf{Z}_i^{\mathcal{E}} \mid \mathbf{H}) \cdot \frac{p_\theta(\mathbf{Z}_i^{\mathcal{E}}, \mathbf{H}) \cdot q_\phi(\mathbf{Z}_i^{\mathcal{E}} \mid \mathbf{H})}{q_\phi(\mathbf{Z}_i^{\mathcal{E}} \mid \mathbf{H}) \cdot p_\theta(\mathbf{Z}_i^{\mathcal{E}} \mid \mathbf{H})} \right\} \\ &= q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H}) \cdot \log \frac{q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})}{p_\theta(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})} \\ &\quad + q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H}) \cdot \log \frac{p_\theta(\mathbf{Z}^{E}, \mathbf{H})}{q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})} \\ &= D_{KL}(q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H}) \parallel p_\theta(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})) + \mathcal{L}(\theta, \phi; \mathcal{E}) \\ &\geq \mathcal{L}(\theta, \phi; \mathbf{H}) \end{aligned} \tag{3}$$

The conditional probability $q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})$ is the variational posterior to approximate the true posterior $p(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})$, where the parameter ϕ is estimated in the inference model. In Equation (3), the second RHS term $\mathcal{L}(\theta, \phi; \mathcal{E})$ is called the evidence lower bound (ELBO) on the marginal likelihood of the variables $\mathcal{E}$:

$$\begin{aligned} \mathcal{L}(\theta, \phi; \mathbf{H}) &= \mathbb{E}_{q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})} \left[-\log q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H}) + \log p_\theta(\mathbf{H}, \mathbf{Z}^{\mathcal{E}}) \right] \\ &= -D_{KL}(q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H}) \parallel p_\theta(\mathbf{Z}^{\mathcal{E}})) \\ &\quad + \mathbb{E}_{q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})} \left[\log p_\theta(\mathbf{H} \mid \mathbf{Z}^{\mathcal{E}}) \right] \end{aligned} \tag{4}$$

where the D_{KL} term denotes the Kullback–Leibler divergence, a measure of how one probability distribution is different from a second. Respectively, we have:

$$\begin{aligned}\mathcal{L}(\theta,\phi;\mathbf{R}) &= -D_{KL}(q_\phi(\mathbf{Z}^{\mathbf{R}} \mid \mathbf{R}) \parallel p_\theta(\mathbf{Z}^{\mathbf{R}})) \\ &\quad + \mathbb{E}_{q_\phi(\mathbf{Z}^{\mathbf{R}}|\mathbf{R})}\Big[\log p_\theta(\mathbf{R} \mid \mathbf{Z}^{\mathbf{R}})\Big] \\ \mathcal{L}(\theta,\phi;\mathbf{T}) &= -D_{KL}(q_\phi(\mathbf{Z}^{\mathbf{E}} \mid \mathbf{T}) \parallel p_\theta(\mathbf{Z}^{\mathbf{E}})) \\ &\quad + \mathbb{E}_{q_\phi(\mathbf{Z}^{\mathbf{E}}|\mathbf{T})}\Big[\log p_\theta(\mathbf{T} \mid \mathbf{Z}^{\mathbf{E}})\Big]\end{aligned} \tag{5}$$

Substituting Equations (3)–(5) into Equation (2), the variational lower bound of $\log \mathcal{O}$ can be represented with the parameters θ and ϕ:

$$\begin{aligned}\log p(\mathbf{O}) &= \log p_\theta(\mathbf{H}) + \log p_\theta(\mathbf{R}) + \log p_\theta(\mathbf{T}) \\ &\geq \mathcal{L}(\theta,\phi;\mathbf{H}) + \mathbf{L}(\theta,\phi;\mathcal{R}) + \mathcal{L}(\theta,\phi;\mathbf{T}) \\ &= \mathcal{L}(\theta,\phi;\mathbf{O})\end{aligned} \tag{6}$$

where

$$\begin{aligned}\mathcal{L}(\theta,\phi;\mathbf{O}) = &- D_{KL}(q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H}) \parallel p_\theta(\mathbf{Z}^{\mathcal{E}})) \\ &+ \mathbb{E}_{q_\phi(\mathbf{Z}^{\mathcal{E}}|\mathbf{H})}\Big[\log p_\theta(\mathbf{H} \mid \mathbf{Z}^{\mathcal{E}})\Big] \\ &- D_{KL}(q_\phi(\mathbf{Z}^{\mathcal{R}} \mid \mathbf{R}) \parallel p_\theta(\mathbf{Z}^{\mathcal{R}})) \\ &+ \mathbb{E}_{q_\phi(\mathbf{Z}^{\mathcal{R}}|\mathbf{R})}\Big[\log p_\theta(\mathcal{R} \mid \mathbf{Z}^{\mathcal{R}})\Big] \\ &- D_{KL}(q_\phi(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{T}) \parallel p_\theta(\mathbf{Z}^{\mathcal{E}})) \\ &+ \mathbb{E}_{q_\phi(\mathbf{Z}^{\mathcal{E}}|\mathbf{T})}\Big[\log p_\theta(\mathbf{T} \mid \mathbf{Z}^{\mathcal{E}})\Big]\end{aligned} \tag{7}$$

In Equation (7), the conditional probabilities $q(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{H})$, $q(\mathbf{Z}^{\mathcal{R}} \mid \mathbf{R})$ and $q(\mathbf{Z}^{\mathcal{E}} \mid \mathbf{T})$ can be regarded as probabilistic encoders to embed real data into latent space. Similarly, the conditional probabilities $p(\mathbf{H} \mid \mathbf{Z}^{\mathcal{E}})$, $p(\mathbf{R} \mid \mathbf{Z}^{\mathcal{R}})$ and $p(\mathbf{T} \mid \mathbf{Z}^{\mathcal{E}})$ can be regarded as probabilistic decoders, producing corresponding data from latent vector representations. To approximate the real distributions of KG components, we assume that the prior distributions and the variational posterior distributions are Gaussian distributions.

$$\begin{aligned}&p(\mathbf{Z_i}^{\mathcal{E}}) = \mathcal{N}(\mathbf{0},\mathbf{I}) \\ &p(\mathbf{Z_j}^{\mathcal{R}}) = \mathcal{N}(\mathbf{0},\mathbf{I}) \\ &q_\phi(\mathbf{Z_h}^{\mathcal{E}} \mid \mathbf{H}) = \mathcal{N}(\overline{\mathbf{E}}, \sigma^2_{\mathcal{E}} \cdot \mathbf{I}) \\ &q_\phi(\mathbf{Z_r}^{\mathcal{R}} \mid \mathbf{R}) = \mathcal{N}(\overline{\mathbf{R}}, \sigma^2_{\mathcal{R}} \cdot \mathbf{I}) \\ &q_\phi(\mathbf{Z_t}^{\mathcal{E}} \mid \mathbf{T}) = \mathcal{N}(\overline{\mathbf{E}}, \sigma^2_{\mathcal{E}} \cdot \mathbf{I})\end{aligned} \tag{8}$$

Assuming priors and variational posteriors to be Gaussian distributions, the D_{KL} terms in Equation (7) can be formed computationally. In addition, we adopt the Monte Carlo gradient estimator to deal with the $\mathbb{E}_{q_\phi}$ terms:

$$\begin{aligned}\mathcal{L}(\theta,\phi;\mathbf{O}) = &\frac{1}{L}\sum_{i=0}^{L}\sum_{(h_i,r_i,t_i)\in\mathcal{O}}^{W}(\log p_\theta(t_i \mid \mathbf{Z}^{\mathcal{E}}_{t_i}) \\ &+ \log p_\theta(h_i \mid \mathbf{Z}^{\mathcal{E}}_{h_i}) + \log p_\theta(r_i \mid \mathbf{Z}^{\mathcal{R}}_{r_i})) \\ &+ \frac{1}{M}\sum_{e_i\in\mathcal{E}}^{M}\sum_{d=0}^{D}(\mu^2_{e_i,d} + \sigma^2_{e_i,d} - \log\sigma^2_{e_i,d} - 1) \\ &+ \frac{1}{N}\sum_{r_i\in\mathcal{R}}^{N}\sum_{d=0}^{D}(\mu^2_{r_i,d} + \sigma^2_{r_i,d} - \log\sigma^2_{r_i,d} - 1),\end{aligned} \tag{9}$$

where D is the output dimension of latent variables, L is the sampling number in the Monte Carlo estimator, and M, N, and W are the number of entities, relations, and triples. We also adopt the reparameterization trick mentioned in the VAE section to generate samples.

$$\begin{aligned}
\mathbf{z}^{\mathcal{E}}_{h_i} &= \overline{\mathbf{h_i}} + \sigma^2_{h_i} \odot \epsilon, \text{ with } h_i \in \mathbf{H}, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
\mathbf{z}^{\mathcal{R}}_{r_i} &= \overline{\mathbf{r_i}} + \sigma^2_{r_i} \odot \epsilon, \text{ with } r_i \in \mathcal{R}, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
\mathbf{z}^{\mathcal{E}}_{t_i} &= \overline{\mathbf{t_i}} + \sigma^2_{t_i} \odot \epsilon, \text{ with } t_i \in \mathbf{T}, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})
\end{aligned} \tag{10}$$

4.2. Learning

To optimize the parameters in Equation (9), we apply two neural networks in VAE: (1) An inference model f_ϕ with parameter ϕ to map observation data into latent vector spaces. (2) A generative model g_θ with parameter θ to produce random variables from latent embeddings.

Inference model f_ϕ. To encode KG components to Gaussian embeddings, we apply two fully-connected layers to map the entities and relations to the means and log-variances in their resulting Gaussian embeddings. One of the benefits of encoding log-variance instead of variance is that it enables us to avoiding using activation functions, since the variance σ^2 must be a positive number.

$$\begin{aligned}
(\overline{\mathbf{h_i}}, \log \sigma^2_{\mathbf{h_i}}) &= f_{\phi_1}(\mathbf{h_i}) \\
(\overline{\mathbf{r_i}}, \log \sigma^2_{\mathbf{r_i}}) &= f_{\phi_2}(\mathbf{r_i}) \\
(\overline{\mathbf{t_i}}, \log \sigma^2_{\mathbf{t_i}}) &= f_{\phi_3}(\mathbf{t_i})
\end{aligned} \tag{11}$$

where $\phi = [\phi_1, \phi_2, \phi_3]$ and μ and $\log \sigma^2$ are the means and log-variances of learned Gaussian embeddings of KG components:

$$\begin{aligned}
q_\phi(\mathbf{z}^{\mathcal{E}}_{\mathbf{h_i}} \mid \mathbf{h_i}) &= \mathcal{N}(\overline{\mathbf{h_i}}, \sigma^2_{\mathbf{h_i}} \cdot \mathbf{I}) \\
q_\phi(\mathbf{z}^{\mathcal{R}}_{\mathbf{r_i}} \mid \mathbf{r_i}) &= \mathcal{N}(\overline{\mathbf{r_i}}, \sigma^2_{\mathbf{r_i}} \cdot \mathbf{I}) \\
q_\phi(\mathbf{z}^{\mathcal{E}}_{\mathbf{t_i}} \mid \mathbf{t_i}) &= \mathcal{N}(\overline{\mathbf{t_i}}, \sigma^2_{\mathbf{t_i}} \cdot \mathbf{I})
\end{aligned} \tag{12}$$

We apply the reparameterization trick mentioned in Equation (10) to obtain the deterministic variables $\mathbf{Z}^{\mathcal{E}}_h$, $\mathbf{Z}^{\mathcal{R}}_r$, and $\mathbf{Z}^{\mathcal{E}}_t$, transformed from latent random variables, with a noise term ϵ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, which benefit from gradient propagation between the inference model and the generative model. We compute the loss of the inference model by measuring the KL divergence between those conditional probabilities and $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

Generative model g_θ. The generative model decodes from deterministic values to random variables. For example, given resulting embeddings $\mathbf{Z}^{\mathcal{E}}$ and $\mathbf{Z}^{\mathcal{R}}$ from a KG represented as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{O})$, our goal is to reconstruct random variables for each triple $(\mathbf{h_i}, \mathbf{r_i}, \mathbf{t_i}) \in \mathcal{O}$, where:

$$p_\theta(\mathbf{h_i}, \mathbf{r_i}, \mathbf{t_i} \mid \mathbf{z}^{\mathcal{E}}_{\mathbf{h_i}}, \mathbf{z}^{\mathcal{R}}_{\mathbf{r_i}}, \mathbf{z}^{\mathcal{E}}_{\mathbf{t_i}}) = g_\theta(\mathbf{z}^{\mathcal{E}}_{\mathbf{h_i}}, \mathbf{z}^{\mathcal{R}}_{\mathbf{r_i}}, \mathbf{z}^{\mathcal{E}}_{\mathbf{t_i}}) \tag{13}$$

The random distributions of those components can be defined as:

$$\begin{aligned}
p_{\theta_1}(\mathbf{h_i} \mid \mathbf{z}^{\mathcal{E}}_{\mathbf{h_i}}) &= \mathcal{N}(\overline{\mathbf{z}_{\mathbf{h_i}}}, \sigma^2_{\mathbf{z}_{\mathbf{h_i}}} \cdot \mathbf{I}) \\
p_{\theta_2}(\mathbf{r_i} \mid \mathbf{z}^{\mathcal{R}}_{\mathbf{r_i}}) &= \mathcal{N}(\overline{\mathbf{z}_{\mathbf{r_i}}}, \sigma^2_{\mathbf{z}_{\mathbf{h_i}}} \cdot \mathbf{I}) \\
p_{\theta_3}(\mathbf{t_i} \mid \mathbf{z}^{\mathcal{E}}_{\mathbf{t_i}}) &= \mathcal{N}(\overline{\mathbf{z}_{\mathbf{t_i}}}, \sigma^2_{\mathbf{z}_{\mathbf{h_i}}} \cdot \mathbf{I})
\end{aligned} \tag{14}$$

where $\theta = [\theta_1, \theta_2, \theta_3]$, and the reconstruction loss of the generative model can be measured based on the binary cross entropy (BCE) between the generative variables and the real data.

5. Experiment

5.1. Data Sets

In this work, we conducted experiments and evaluated the related methods using real-world databases of KG, commonly used in previous works: WordNet [24] and Freebase [25]. WordNet is an extensive lexical database of English. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept, and with interlinked sysnets employing conceptual-semantic and lexical relations. Freebase is a large collaborative knowledge base consisting of data compiled mainly by its community members. It is an online collection of structured data harvested from many sources, including individual wiki contributions. The most representative dataset in WorldNet is WN18, and FB15k in Freebase.

In those datasets, WN18 contains 18 relations and 40,943 entities, whereas FB15k contains 1345 relations and 14,951 entities. However, both of them suffer from test leakage through inverse relations: a large number of test triples can be obtained simply by inverting triples in the training set. Therefore, we introduced FB15k-237, a subset of FB15k, in which reversible relations were removed. Similarly, WN18 was corrected by WN18RR. Therefore, we selected WN18RR and FB15k-237 as datasets in our experiments.

5.2. Experimental Setup

We compared our models with serveral KG embedding algorithms in our experiments:

1. TransE [6]. TransE was the first model to introduce translation-based embedding, which interprets relations as the translations operating on entities.
2. DistMult [10]. DistMult is based on the bilinear model, where each relation is represented by a diagonal rather than a full matrix. DistMult enjoys the same scalable properties as TransE and it achieves superior performance over TransE.
3. ComplEx [11]. ComplEx extends DistMult by introducing complex-valued embeddings so as to better model asymmetric relations. It has been proven that HolE is subsumed by ComplEx as a special case.
4. ConvE [12]. ConvE is a multi-layer convolutional network model for link prediction [24] of KGs, and it reports state-of-the-art results for several established datasets. Unlike previous work which has focused on shallow, fast models that can scale to large knowledge graphs, ConvE uses 2D convolution over embeddings and multiple layers of nonlinear features to model KGs.
5. ConvKB [13]. ConvKB applies the global relationships among same-dimensional entries of the entity and relation embeddings, so that ConvKB generalizes the transitional characteristics in the transition-based embedding models.
6. R-GCN [14]. R-GCN applies graph convolutional networks to relational knowledge bases, creating a new encoder for link prediction and entity classification tasks.

The experimental results from those baselines were obtained from the codes provided by the authors. In our method, we made configurations by selecting a learning rate α among [0.01, 0.05, 0.10] and an output dimension D among [100, 200, 400]. For WN18RR, the configuration was as follows. The learning rate α was 0.01 and the output dimension D was 400, with 3000 training iterations using the Adam [27] optimizer. For FB15k-237, the configuration was as fallows. The learning rate α was 0.10, and the output dimension D was 200, with 1000 training iterations using the SGD optimizer. We trained the model until it converged.

5.3. Link Prediction

Link prediction, aiming at predicting the missing KG components for incomplete triples, is a typical task in KG embedding. e.g., predicting the head entity for a given triple $(*, \mathbf{r}, \mathbf{t})$ or predicting the tail entity for a given triple $(\mathbf{h}, \mathbf{r}, *)$. Following the protocols in [6], we evaluated the performance of our model. Given a test triple, we replaced the head or tail with all available entities and ranked them by measuring the scoring function defined

in the methods section. Based on the ranking lists, we report the proportion of correct entities in the top N ranked entities, where N = 1, 3, and 10, denoted as Hits@1, Hits@3, and Hits@10.

$$MRR = \frac{1}{|M|}\sum_{i=0}^{M}\frac{1}{rank(e_i)}$$
$$MR = \frac{1}{|M|}\sum_{i=0}^{M} rank(e_i) \quad (15)$$

We also record the average reciprocal rank of correct entities (denoted as MRR) and the average rank of correct entities (denoted as MR) for link prediction, where the function $rank(e_i)$ transforms to the rank of e_i. A good embedding algorithm should obtain a relatively low mean rank and a relatively high mean reciprocal rank.

5.4. Results and Analysis

In this subsection, we report the ability of our model to represent uncertainty, and the experimental results regarding link prediction.

Qualitative Analysis Before evaluating the performance in specific task compared with other methods, we need to discuss the ability of our model to represent uncertainty in KG embedding.

In our method, we measure the uncertainty of KG components by the variances of their embeddings, where an entity/relation with a higher level of uncertainty has a large covariance. We discuss the relations in FB15k-237 with '/education' as the domain, providing a (log) determinant and trace of their covariances as shown in Table 2, from which we have made the following observations:

1. Our method has the ability to measure the uncertainty in KG embedding. The covariance of Gaussian embedding can effectively describe the uncertainties by calculating the determinants and traces of the covariances.
2. The relations with more semantic information (the number of associated heads and tails, type of relation) have larger uncertainty. For example, the 'major_field_of_study' relation has the largest uncertainty, and the 'educational_insitution' relation has the smallest uncertainty in those relations.

Table 2. The relations with /education/ as the domain and their determinants and traces of the corresponding covariances, sorted by descending order of traces.

Relation	#Head	#Tail	Type	log (det)	Trace
major_field_of_study	225	77	m-n	−338.8	38.1
student	183	292	1-n	−340.6	34.8
institution	22	222	m-n	−376.2	32.8
colors	85	19	m-n	−400.9	26.9
fraternities_sororities	20	3	m-1	−406.9	24.9
campuses	13	13	1-1	−411.9	21.3
currency	5	3	m-1	−423.4	19.8
educational_institution	13	13	1-1	−430.6	18.7

6. Results

We compared our method with the state-of-the-art baselines mentioned above, including TransE, DisMult, ComplEx, ConvKB, and R-GCN. First of all, the codes in the baseline we used are provided by other authors. All models were fully trained, and the data sets used were public. Our models, in both the Hits@3 and Hits@10 metrics for this dataset, achieved superior results, which proves that the embedding obtained using our proposed method is of high quality. The experimental results regarding link prediction are shown in Table 3. We observe that:

1. The experimental results on FB15k-237 and WN18RR indicate that our method can learn high-quality representations in KG.
2. Our method outperformed other baselines in terms of the Hits@3 and Hits@10 metrics, but its performance was poor in terms of mean reciprocal rank and the Hits@1 metric on WN18RR. This may be because WN18RR contains a large number of entities and several relations, so most methods can only judge the correctness of a triple but cannot rank it in the top position.
3. On FB15k-237, our method outperformed other baselines in terms of the Hits@3, Hits@10, and mean reciprocal rank metrics, and came second in terms of the Hits@1 and mean rank metrics. The improvements observed in FB15k-237 were greater than those in WN18RR, showing that FB15k-237 contains more relations and thus the uncertainties in its components are larger than those in WN18RR, which indicates that our method can learn valid representations with uncertainties in KG.

Table 3. Experimental results for WN18RR and FB15k-237 test sets. Hits@N values are presented as percentages. The best score is in bold and the second best score is underlined.

	WN18					**FB15k-237**				
	MR	**MRR**	**HITS@N**			**MR**	**MRR**	**HITS@N**		
			1	**3**	**10**			**1**	**3**	**10**
TransE (Bordes et al., 2013) [6]	2300	0.243	4.27	44.1	53.2	323	0.279	19.8	37.6	44.1
DistMult (Yang et al., 2015) [10]	7000	0.444	41.2	47	50.4	512	0.281	19.9	30.1	44.6
ComplEx (Trouillon et al., 2016) [11]	7882	0.449	40.9	46.9	53	546	0.278	19.4	29.7	45
ConvE (Dettmers et al., 2018) [12]	4464	**0.456**	**41.9**	47	53.1	245	0.312	**22.5**	34.1	49.7
ConvKB (Nguyen et al., 2018) [13]	**1295**	0.265	5.82	44.5	55.8	**216**	0.289	19.8	32.4	47.1
R-GCN (Schlichtkrull et al., 2018) [14]	6700	0.123	8	13.7	20.7	600	0.164	10	18.1	30
Our work	1963	0.236	11.4	**48.0**	**57.6**	240	**0.518**	21.8	**42.0**	**52.1**

7. Conclusions

In this paper, we propose the co-embedding model to learn the latent representations of both entities and relations in the same semantic space, embedding them as Gaussian distributions. To obtain high-quality embeddings, we introduced the variational auto-encoder, an auto-encoder model consisting of a probabilistic encoder and a probabilistic decoder, into our model. One of the assets of the technique is that the affinities between entities and relations can be measured effectively since they are embedded in the same semantic space, and we also explain the transformation from observation values to latent representations via the two models using the variational auto-encoder. In our experiments, we evaluated the performance of the co-embedding model and other baselines on several benchmark datasets. From these experimental results, we can conclude that our method can learn high-quality representations of KG components.

In the future, we plan to extend our method by assuming the priors with other distributions and optimizing the variational lower bounds in an effective way.

Author Contributions: Conceptualization, H.H.; data curation, L.X. and Q.D.; formal analysis, L.X.; methodology, L.X. and H.H.; project administration, H.H.; software, Q.D.; supervision, H.H.; validation, H.H.; visualization, L.X. and Q.D.; writing—original draft, L.X.; writing—review and editing, H.H.; funding acquisition, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Wenzhou Science and Technology Planning Project #2021R0082.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, of in the decision to publish the results.

References

1. Berant, J.; Chou, A.; Frostig, R.; Liang, P. Semantic Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1533–1544.
2. Heck, L.; Hakkani-Tür, D.; Tur, G. Leveraging Knowledge Graphs for Web-Scale Unsupervised Semantic Parsing. In Proceedings of the International Speech Communication Association, Lyon, France, 25–29 August 2013.
3. Wang, W.Y.; Mazaitis, K.; Lao, N.; Mitchell, T.; Cohen, W.W. Efficient Inference and Learning in a Large Knowledge Base: Reasoning with Extracted Information using a Locally Groundable First-Order Probabilistic Logic. *arXiv* **2014**, arXiv:cs.AI/1404.3301.
4. Bordes, A.; Weston, J.; Usunier, N. Open Question Answering with Weakly Supervised Embedding Models. *arXiv* **2014**, arXiv:cs.CL/1404.4326.
5. Bordes, A.; Chopra, S.; Weston, J. Question Answering with Subgraph Embeddings. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 615–620. [CrossRef]
6. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; pp. 2787–2795.
7. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec, QC, Canada, 27–31 July 2014; Volume 28.
8. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.
9. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the ICML, Bellevue, WA, USA, 28 June–2 July 2011.
10. Yang, B.; tau Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *arXiv* **2014**, arXiv:cs.CL/1412.6575.
11. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex Embeddings for Simple Link Prediction. *arXiv* **2016**, arXiv:cs.AI/1606.06357.
12. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2D Knowledge Graph Embeddings. *arXiv* **2017**, arXiv:cs.LG/1707.01476.
13. Nguyen, D.Q.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 2, pp. 327–333. [CrossRef]
14. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; van den Berg, R.; Titov, I.; Welling, M. Modeling Relational Data with Graph Convolutional Networks. *arXiv* **2017**, arXiv:stat.ML/1703.06103.
15. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
16. Paccanaro, A.; Hinton, G.E. Learning distributed representations of concepts using linear relational embedding. *IEEE Trans. Knowl. Data Eng.* **2001**, *13*, 232–244. [CrossRef]
17. He, S.; Liu, K.; Ji, G.; Zhao, J. Learning to Represent Knowledge Graphs with Gaussian Embedding. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM'15, New York, NY, USA, 19–30 October 2015; pp. 623–632. [CrossRef]
18. Vilnis, L.; McCallum, A. Word Representations via Gaussian Embedding. *arXiv* **2014**, arXiv:cs.CL/1412.6623.
19. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:stat.ML/1312.6114.
20. Kingma, D.P.; Rezende, D.J.; Mohamed, S.; Welling, M. Semi-Supervised Learning with Deep Generative Models. *arXiv* **2014**, arXiv:cs.LG/1406.5298.
21. Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; Zhou, H. Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering. *arXiv* **2016**, arXiv:cs.CV/1611.05148.
22. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial Autoencoders. *arXiv* **2015**, arXiv:cs.LG/1511.05644.
23. Dosovitskiy, A.; Brox, T. Generating Images with Perceptual Similarity Metrics based on Deep Networks. *arXiv* **2016**, arXiv:cs.LG/1602.02644.
24. Miller, G.A. WordNet: A Lexical Database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
25. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD'08, Vancouver, BC, Canada, 9–12 June 2008; pp. 1247–1250. [CrossRef]