Egyptian Informatics Journal 24 (2023) 100414

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Knowledge graph completion method based on hyperbolic representation learning and contrastive learning

Xiaodong Zhang [a], Meng Wang [a,*], Xiuwen Zhong [a], Feixu An [b]

[a] School of Computer Science, Nanjing Audit University, Nanjing 211815, China
[b] University of Toronto, Toronto M2J4A6, 211189, Canada

ARTICLE INFO

Keywords:
Knowledge graph completion
Hyperbolic representation learning
Comparison learning
Adversarial samples

ABSTRACT

Knowledge graph completion employs existing triples to deduce missing data, thereby enriching and enhancing graph completeness. Recent research has revealed that using hyperbolic representation learning in knowledge graph completion yields superior expressive and generalization capabilities. However, the long-tail problem and the presence of hyperbolic metrics make it challenging to effectively learn low-frequency entities or relations, resulting in embedding space distortion and impacting the original semantic relationships. Therefore, this paper proposes a knowledge graph completion method (Att-CL) that integrates hyperbolic representation learning and contrastive learning. In this approach, knowledge is embedded into a hyperbolic space, and samples with limited hierarchical characteristics and insufficient feature information are enhanced by introducing adversarial noise. The loss function of the embedded samples is backpropagated into embedding vectors, perturbations are adjusted in the gradient direction to promote smoothness and locality, and hyperparameters are introduced for fine-tuning the adversarial strength in the construction of adversarial samples for data augmentation to enhance model robustness. To mitigate data distortion due to hyperbolic metrics, a penalty term is introduced in the contrastive loss function to control the distances of the embedding vectors from the origin, thereby reducing the impact of the metrics and further improving the model's completion ability. Experimental results on the WN18RR and FB15K-237 benchmark datasets demonstrate significant improvements in metrics such as MRR, Hits@1, and Hits@3 compared to traditional knowledge graph completion models, providing ample evidence of the model's effectiveness.

1. Introduction

A knowledge graph is a comprehensive and organized network of structured data, designed to illustrate the connections and relationships among various entities. It consists of nodes and edges, consists of nodes that represent distinct entities, and edges that denote the relationships between these entities [1]. At present, knowledge graphs are widely used in many AI fields, such as natural language processing and search recommendation [2]. However, although standard knowledge graphs (e.g., WordNet) are widely applied, they still face a problem of information incompleteness [3]. They contain large amounts of unrevealed semantic information. Therefore, knowledge graph completion, with the goal of automatically inferring the missing content based on the existing information and external data in order to fill in missing and erroneous information to construct a more complete and accurate graph, has become a vital focus of research for enhancing the quality and application value of knowledge graphs [4,5]. The following are some examples of ways to achieve this purpose. Currently, hyperbolic representation learning and contrastive learning have become popular research topics in the field of graph embedding. In hyperbolic representation learning, the nodes in a graph are mapped to a hyperbolic space, and the representation and computation of the nodes are performed in this space. The rich geometric structure of the hyperbolic space itself is utilized to better portray the nonlinear relationships and hierarchical structure of the data [6]. Meanwhile, contrastive learning is also increasingly combined with knowledge embedding models to enable the models to better distinguish different entities and relationships and obtain more semantically informative representations by comparing the similarities and differences between positive and negative sample pairs [7]. Most existing contrastive learning methods rely excessively on pre-existing training data; consequently, they perform poorly when dealing with entirely new, unseen data and generalize

* Corresponding author.
E-mail addresses: 270050@nau.edu.cn (M. Wang), xufei.an@mail.utoronto.ca (F. An).

https://doi.org/10.1016/j.eij.2023.100414
Received 24 September 2023; Received in revised form 22 October 2023; Accepted 30 October 2023
Available online 4 November 2023

poorly to low-frequency data when dealing with long-tailed distributions [8]. Hyperbolic representation learning methods also suffer from an insufficient number of samples during model training due to the long-tail problem, and the hyperbolic nature of the spatial metric causes nodes in embedding vectors that are farther away from the origin to be more susceptible to data distortion, making it challenging to represent the semantic relationships between them accurately [9].

To address the abovementioned issues, this paper proposes the Att-CL method, which integrates hyperbolic representation learning and contrastive learning. This method involves embedding knowledge into a hyperbolic space. For samples with limited hierarchical characteristics and insufficient feature information, a method of overlaying adversarial noise is employed. The loss function of the embedded samples is back-propagated to the embedding vectors. Perturbations are added and adjusted in the gradient direction to make them smoother and more localized. Simultaneously, a hyperparameter is introduced to fine-tune the adversarial strength when constructing adversarial samples for data augmentation, thereby enhancing the model's robustness and mitigating the impacts of insufficient features and insufficient training due to the long-tail problem in knowledge graph completion tasks. To optimize the data distortion caused by hyperbolic metric constraints, a penalty term is introduced in the contrastive loss function to control the distances of the embedding vectors from the origin, reducing the influence of the metrics and further improving the completion ability of the model.

2. Related work

Embedding is a technique commonly employed for addressing the knowledge graph completion problem. It involves mapping entities and relations into a lower-dimensional vector space, allowing for the quantification of entity similarity and relation distance. Effectively capturing the interconnectedness among entities while retaining the integrity of the original information [10,11]. The TransE model [12] uses the Euclidean distance to measure the matching between entities and relations; it combines entity vectors with relation vectors to capture interentity relations, but it has difficulty handling complex relations. The DistMult model [13] uses the dot product to measure entity and relation similarity; it avoids illegitimate entity combinations by restricting the relation vectors, but it is limited in its representation ability. The ComplEx model [14] extends the DistMult model to the complex domain and uses complex vectors to represent entities and relations and compute their similarity, but it has many parameters and is computationally complex. Graph neural network models focus on capturing complex entity and relation interactions to forecast and recover missing data [15]. The ConvE model [16] uses convolutional neural networks to learn the interaction patterns of entities and relationships, which are then applied for information prediction and filling.

In contrast, hyperbolic representation learning uses hyperbolic geometry to map nodes into a hyperbolic space and is better at handling hierarchical structures and nonlinear relationships [17]. Learning the hyperbolic distances between nodes is more suitable for dealing with sparse, high-dimensional data, leading to stronger expressive and generalization capabilities [18]. The MuRP model [19] is an extension of Poincaré embedding [20] in which entities and relations are embedded into the Poincaré sphere and the embedding vectors are optimized using a negative contrastive loss function. AttH [21] considers interentity relations in conjunction with attention mechanisms to enhance or attenuate entity interactions through rotation, reflection and attention. KGAT [22] combines graph neural networks with hyperbolic representation learning, using attention mechanisms to capture entity and relationship importance from knowledge embedded in a hyperbolic space. Methods based on hyperbolic representation learning have demonstrated excellent performance. However, due to the metric properties of hyperbolic space, distances between nodes far from the origin may be excessively amplified, resulting in data distortion in the embedding space, which limits the accurate representation of semantic relationships.

In contrastive learning [23], by comparing the similarities and differences between positive and negative sample pairs and increasing the diversity of the training samples, it is possible to better distinguish between different entities and relationships to obtain more semantically informative representations [24]. SimCTG [25] is a contrastive learning framework proposed for neural text generation tasks that addresses the problems encountered in decoding methods, encourages diversity and coherence, and improves the calibration of the representation space of language models. SimKGC [26] uses a dual encoder architecture, introduces three harmful sample types, and uses the cosine similarity for tail entity prediction to improve the learning efficiency. KGE-CL [27] captures the semantic similarity of related entities and entity–relationship pairs in different triples through contrastive learning and incorporates adaptive weighting, joint sampling, and an optimized sampling distribution to improve the sampling efficiency and model performance. However, when faced with long-tailed data, these methods have limited model generalization capabilities due to insufficient training examples.

To address the aforementioned issues, this paper proposes a knowledge graph completion model that integrates hyperbolic representation learning and contrastive learning. Positive and negative samples are constructed for contrastive learning using triples from the existing knowledge graph. For samples with insufficient feature information, adversarial samples are created by adding adversarial noise, thereby enhancing the model's robustness and addressing the long-tail problem through data augmentation. Additionally, a penalty term is introduced to achieve better control over the contrastive loss function. This optimization mitigates the data distortion caused by the long-tail problem and the use of hyperbolic metrics, thereby enhancing the graph completion ability.

3. Knowledge graph completion method based on hyperbolic representation learning and contrastive learning

3.1. Formal definitions

In this paper, S is used to represent the set of all triples in the knowledge graph, and each triple is denoted by (h,r,t), where h denotes the head entity, r denotes the relationship, and t denotes the tail entity. Z denotes the set of entities, R denotes the set of relationships, and e_Z^H and r^H denote the hyperbolic embeddings of entities and relations, respectively. c is the curvature of the hyperbolic space, and $\oplus$ c denotes Möbius addition. d^c denote the distance measure in hyperbolic space, θ_r and φ_r are formal definitions of rotations and reflections with relation specificity, $G^{\pm}(\theta)$ are the Givens transformations, b_h and b_t are hyperbolic decision boundaries, x^E and y^E denote the embedded data points of a particular relation in the hyperbolic space after reflection and rotation transformations, and the attention weight vector is denoted by α. The formal definitions are expressed as follows:

Exponential arithmetic function:

$$d^c(x,y) = \frac{2}{\sqrt{c}} arctanh(\sqrt{c}\| -x \oplus cy\|) \tag{1}$$

Exponential arithmetic function:

$$exp_0^c(v) = tanh(\sqrt{c}\|v\|)\frac{v}{\sqrt{c}\|v\|} \tag{2}$$

Logarithmic arithmetic function:

$$log_0^c(y) = arctanh(\sqrt{c}\|y\|)\frac{y}{\sqrt{c}\|y\|} \tag{3}$$

Möbius addition of vectors:

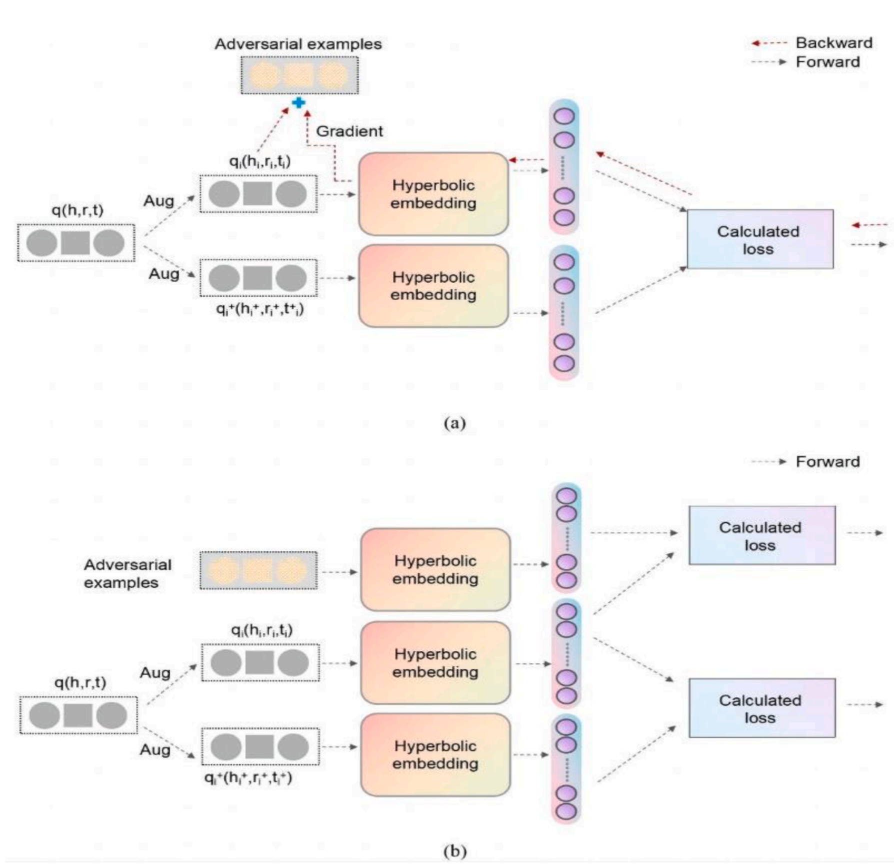

Fig. 1. Construction of adversarial samples: (a) Against the sample building process, (b) Adversarial sample joining training.

$$x \oplus cy = \frac{(1 + 2Cx^T y + C\|y\|^2)x + (1 - C\|x\|^2)y}{1 + 2Cx^T y + C^2\|x\|^2\|y\|^2} \tag{4}$$

Rotation operation:

$$Rot(\theta_r) = diag(G^+(\theta_r, 1), \cdots, G^+\left(\theta_r, \frac{d}{2}\right)\right) \tag{5}$$

Reflection operation:

$$Ref(\varphi_r) = diag\left(G^-(\varphi_r, 1), \cdots, G^-\left(\varphi_r, \frac{n}{2}\right)\right) \tag{6}$$

Hyperbolic attention function:

$$(\alpha_x, \alpha_y) = Softmax(\alpha_x x^E, \alpha_y y^E) \tag{7}$$

Combined vector function:

$$q(h, r) = (\alpha_x q^H_{Rot} + \alpha_y q^H_{Ref}; a_r) \oplus^{c_r} r^H \tag{8}$$

Hyperbolic embedding scoring function:

$$f(h, r, t) = -d^{c_r}(q(h, r), e^H_t)^2 + b_h + b_t \tag{9}$$

3.2. Adversarial sample data enhancement method based on hyperbolic representation learning

Traditional models for hyperbolic representation learning utilize hyperbolic space to embed knowledge graphs during graph completion tasks. The existing triples in the knowledge graph are utilized as positive sample pairs, and mismatched entities and relations are randomly generated as adversarial sample pairs. However, when facing the long-tailed data problem, especially when the number of dataset levels and the overall network branching situation are limited and fragmented, more information about the sample features is needed. Consequently, it is challenging to generate uniformly effective positive and negative samples for data augmentation, leading to a reduction in model performance with respect to its generalization capabilities. Therefore, this paper proposes an adversarial sample data enhancement method based on hyperbolic representation learning. Adversarial samples refer to the creation of samples that are similar to real ones but slightly different. This is achieved by introducing specific perturbations or noise into the original data. After obtaining hyperbolic embedding vectors using the hyperbolic representation learning model ATTH, we superimpose adversarial noise to generate adversarial samples. This approach not only increases data diversity and broadens the training data's scope but also empowers the model to more effectively understand and handle the rare and imbalanced entities and relationships in the knowledge graph. Additionally, it diminishes the influence of the long-tail problem on knowledge graph completion and concurrently alleviates overfitting. Since the introduced adversarial noise is based on the distribution of real data, the model can more effectively learn the intrinsic data features, reducing its dependence on specific sample distributions. This improvement enhances the model's robustness and generalization abilities.

In this paper, we propose a data enhancement method based on hyperbolic representation learning called Generating Adversarial Samples (GAS), which generates adversarial samples for embedded samples $q_i(h_i, r_i)$. For the embedded samples for which the number of relationally connected entities in each triple is less than the average value, adversarial noise is used to construct adversarial samples for data enhancement. That is, the embedded samples are the first pairs of vectors $q_i(h_i, r_i)$. For $q_i^+(h_i, r_i)$, the loss function $\mathscr{L}_{Att}$ is calculated; then, this loss $\mathscr{L}_{Att}$ is backpropagated to the embedded vectors in the input samples, and $\mathscr{L}_{Att}$ is used to calculate the gradient of the pair of input embedded vectors, ∇L. This gradient will indicate the direction in which to increase the loss given the current model parameters. Next, the gradient vector is normalized with respect to $\|\nabla L\|$ to control the range of the gradient. A perturbation δ that is proportional to the calculated gradient is added to the embedding vector in the gradient direction. By scaling the perturbation δ relative to the size of the gradient, a smoother and more localized result is obtained, resulting in a smooth and continuous embedding and thus preserving the original semantic information. To this end, a hyperparameter α for scaling the gradient is introduced to achieve fine-grained control of the adversarial strength; i.e., by adjusting α, the strength of the adversarial perturbation can be fine-tuned. The perturbation δ is also multiplied by the gradient sign $sign(\nabla_L^{'})$ to make the adversarial samples move in the increasing direction of the loss function so that the constructed adversarial samples will have better discriminative properties in the hyperbolic space. The constructed adversarial samples are then mapped back to the hyperbolic space by means of an exponential operation. This approach allows the model to more effectively accommodate various forms of noise and disruptions, thereby enhancing the model's generalization capacity and overall performance. The relevant formulas are as follows:

$$\nabla L_i = \frac{\partial \mathscr{L}_{Att}(q_i, q_i^+)}{\partial(q_i)} \tag{10}$$

$$\|\nabla L_i\| = \sqrt{\sum_{i=1}^{d} (\nabla L_i)^2} \tag{11}$$

$$\nabla L_i^{'} = c\frac{\nabla L_i}{\|\nabla L_i\|} \tag{12}$$

$$\delta = \varepsilon \cdot sign(\nabla L_i) \cdot \alpha \tag{13}$$

$$q_{ad} = exp_0^c(log_0^c(q) + \delta \cdot sign(\nabla L_i^{'})) \tag{14}$$

$$\mathscr{L}_{Att} = \sum_{(h,r,t)\in Z} log(1 + exp(f(h,r,t))) + \sum_{(h,r,t^{'})\notin Z} log(1 + exp(f(h,r,t^{'}))) \tag{15}$$

Due to the nature of the spatial metric in the hyperbolic space, to avoid the constructed adversarial samples being distorted by the influence of nodes that are too far away from the origin, the variable c is set in Eq. (12) to control the range of the gradient and the distance from the origin in the hyperbolic space. In Eq. (13), δ is the perturbation vector, and ε is the magnitude of the perturbation; $\mathrm{sign}(\nabla L)$ is a symbolic function of the gradient ∇L that takes the sign of each component of the gradient, thus retaining the direction information, and α is the control factor that is used to scale the magnitude of the gradient to control the range and size of the perturbation. In Eq. (14), q is the embedding vector of the input sample, and *sign* is the sign function, which ensures that the direction of the added perturbation is consistent with the direction of the gradient. The adversarial samples that are obtained by adding perturbations in the direction of the gradient are then used along with the original positive samples to train the model, thereby enhancing the model's resistance to different types of noise and attacks. In Eq. (15), $f(h, r, t)$ is the hyperbolic embedding scoring function, which compares the resulting embedding with the target tail entity after the isometric transformation through the hyperbolic distance is applied, and $\mathscr{L}_{Att}$ is the final cross-entropy loss function for the embedding. The process of constructing adversarial samples is illustrated in Fig. 1, and the algorithm is given below.

Algorithm 1: Generating Adversarial Samples (GAS)

```
Input: Triplets (q_i, q_j), Avg_rel_numn, dataset, ε, α
Output: adversarial samples
  Step 1: Calculate the Adversarial Loss ℒ_Att;
    ℒ_Att ← Calculate loss (q_i, q_j);
  Step 2: Check the number of entities connected by relations and the average
number;
    foreach triplet (q_i, q_j) in dataset do
      if num_enti_rel(q_i) < Avg_rel_numn then
        proceed to Step 3;
        else
        skip to the next triplet;
      end
    end
  Step 3: Backpropagate to compute the gradient ∇L_i;
    ∇L_i ← ∂ℒ_Att/∂q_i;
  Step 4: Generate counter noise;
    δ ← ε • sign(∇L_i) • α;
  Step 5: Normalize the gradient;
    norm_grad ← normalize(∇L_i);
  Step 6: Generate adversarial samples;
    adversarial_samples ← { };
    foreach embedding q_i in triplets do
      adversarial sample ← q_i + δ • sign(∇L_i);
      adversarial-samples.add(adversarial-sample);
    end
  Step 7: Index mapping;
      adversarial_samples ← Index mapping(adversarial_samples);
  return adversarial_samples
```

3.3. Contrastive loss method incorporating hyperbolic representation learning

This paper proposes a contrastive loss method with a penalty term to integrate contrastive learning with hyperbolic representation learning and avoid distortion of the embedding vectors caused by the hyperbolic spatial metric, which can cause the samples to be overdispersed in the hyperbolic space. By introducing an appropriate penalty term into the contrastive loss function, the hyperbolic distance of the embedding vector from the origin is adjusted to avoid the influence of the metric, which causes embedding vectors to deviate far from the origin, resulting in data distortion. With this penalty term, after positive samples are mapped into the hyperbolic space by means of the contrastive loss function, they will have similar features that retain as much information as possible. At the same time, they will be distributed as evenly as possible over the space.

In this paper, we propose the Contrastive Loss in Hyperbolic Space (CLHS) method for fused hyperbolic representation learning. the first step is to compute the hyperbolic distances $d^{c_r}\left(q_i, q_j\right)$ between two embedded vector samples (Eq. (1) and use the exponentials of the distances, $e^{d^{c_r}(q_i,q_j)}$ and $e^{d^{c_r}(q_j,q_i)}$, to help the model learn the similarities and differences between the samples. This is because, in contrastive learning, it is desired to have smaller distances between pairs of positive samples and larger distances between pairs of negative samples. Therefore, considering the distances in both directions allows for better learning of the similarities and differences between samples. In the second step, normalization factors are calculated, denoted by z_i and z_j. Due to the unique nature of metrics in hyperbolic space, the distances between nodes far away from the origin and other nodes will tend towards infinity. To avoid drastic changes in the output results, leading to unstable calculation results, normalization is needed. In the third step, the hyperbolic distance is converted into a similarity score for subsequent processing using the contrastive loss function. In the fourth step, the distance from the origin is considered in the contrastive loss function. The embedding vector's distance from the origin is adjusted by adding a penalty term to avoid the influence of the hyperbolic spatial metric, which leads to data distortion. When setting the weight of the

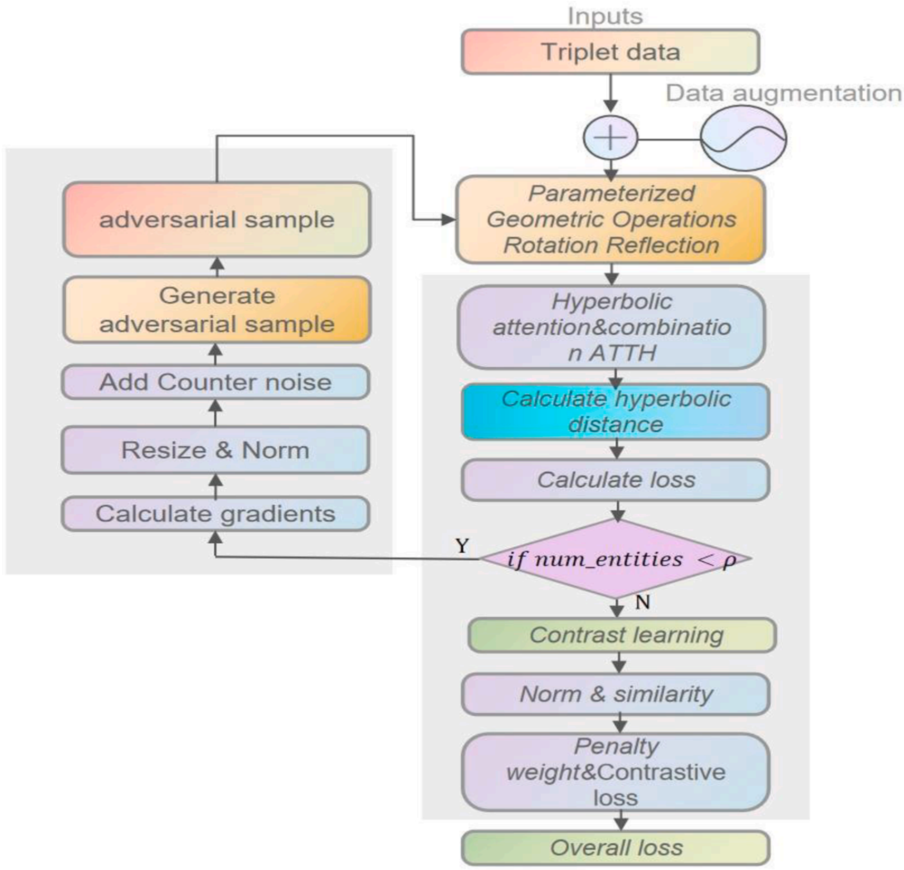

Fig. 2. Overall diagram of the Att-CL framework.

penalty term, λ_r, the relation-specific curvature c_r of the embedding vector is adjusted through a mapping function, and the result is normalized. The output value of the mapping function is divided by a constant K so that the final value of the penalty term weights will lie in the range of [0,1].

$$z_i = \sum_{k=1}^{N} e^{d^{c_r}(q_i,q_k)}, z_j = \sum_{k=1}^{N} e^{d^{c_r}(q_j,q_k)} \tag{16}$$

$$s(q_i, q_j) = -\frac{1}{2}\left(\frac{e^{d^{c_r}(q_i,q_j)}}{z_i} + \frac{e^{d^{c_r}(q_j,q_i)}}{z_j}\right) \tag{17}$$

$$f(c_r') = \frac{1}{(1 + exp(-c_r))} \tag{18}$$

$$\lambda\_r = \frac{f(c_r')}{K} \tag{19}$$

$$\mathscr{L}_{cl} = \frac{1}{|x|x(|x|-1)} \sum_{i=1,j=1}^{|x|} \sum_{j \neq i}^{|x|} max\{0, \rho - s(q_{h_i}, q_{h_i}) + s(q_{h_i}, q_{h_j}) + \lambda\_r \bullet s(q_{h_i}, 0)\} \tag{20}$$

In Eq. (16), q_i and q_j denote the two embedding vectors of the input samples for which the comparison loss calculation is performed, namely, the two samples in the positive (or negative) sample pair. N represents the overall count of samples within the sample set, and $d^{c_r}(q_i, q_k)$ and $d^{c_r}(q_j, q_k)$ denote the hyperbolic distances between the embedding vectors q_i and q_j and the embedding vector of the k-th sample in the sample set. z_i denotes the sum of the exponentials of the hyperbolic distances from the embedding vector q_i to the embedding vectors of all samples in the sample set; similarly, z_j denotes the sum of the exponentials of the hyperbolic distances from the embedding vector q_j to the embedding vectors of all samples in the sample set. In Eq. (17), $e^{d^{c_r}(q_i,q_j)}$ denotes the exponential of the distance from embedding vector q_i to q_j, and $e^{d^{c_r}(q_j,q_i)}$ denotes the exponential of the distance from embedding vector q_j to q_i. In contrastive learning, the exponentials of the distances between pairs of positive samples is desired to be smaller, and the exponentials of the distances between pairs of negative samples should be larger. Therefore, considering the distances in both directions can help the model better learn the similarities and differences between samples. The sigmoid function is used in Eq. (18) to map c_r to the range [0,1] such that relationships with smaller c_r map to values closer to 0 and relationships with larger c_r map to values closer to 1. In this way, the curvatures are converted into penalty term weights that are used to calibrate the embedding vectors in the representation space in the following contrastive loss function. In Eq. (20), λ_r is a hyperparameter used to control the penalty term based on the embedding vector's distance from the origin, which can simultaneously bring positive sample pairs closer and push negative sample pairs farther apart while constraining the embedding vectors to not be too far away from the origin.

The contrastive loss algorithm is given below.

Algorithm 2: Contrastive Loss in Hyperbolic Space (CLHS)

Input: Triplets (q_i, q_j), curvature c, K, $|X|$, ρ
Output: Contrastive Loss $\mathscr{L}_{cl}$
Step 1: Calculate Hyperbolic Distance d^c;
$d^c \leftarrow \frac{2}{\sqrt{c}} \bullet \left(\sqrt{c} - q_i \oplus cq_j\right)$;
Step 2: Calculate Normalization Factors z_i and z_j;
$z_i \leftarrow 0, z_j \leftarrow 0$;
foreach $k = 1$ to N **do**
$z_i \leftarrow z_i + e^{d^c(q_i, q_k)}$;
$z_j \leftarrow z_j + e^{d^c(q_j, q_k)}$;
end
Step 3: Compute Similarity Score s;
$s \leftarrow -\frac{1}{2}\left(\frac{e^{d^c(q_i, q_j)}}{z_i} + \frac{e^{d^c(q_j, q_i)}}{z_j}\right)$;
Step 4: Compute Penalty Term;
$c_r' \leftarrow \frac{1}{1 + \exp(-c_r)}$;
$\lambda_r \leftarrow \frac{c_r'}{K}$;
Step 5: Calculate Contrastive Loss $\mathscr{L}_{cl}$;
$\mathscr{L}_{cl} \leftarrow 0$;
foreach $i = 1$ to $|X|$ **do**
foreach $j = 1$ to $|X|$ **do**
if $j \neq i$ **then**
$s_{ij} \leftarrow s(q_{h_i}, q_{h_i}) - s\left(q_{h_i}, q_{h_j}\right) + \lambda_r \bullet s(q_{h_i}, 0)$;
$\mathscr{L}_{cl} \leftarrow \mathscr{L}_{cl} + \max(0, \rho - s_{ij})$;
end
end
end
$\mathscr{L}_{cl} \leftarrow \frac{1}{|X| \bullet (|X| - 1)} \bullet \mathscr{L}_{cl}$;
return $\mathscr{L}_{cl}$

3.4. Overall model framework

To address the problems of long-tailed data and the influence of the hyperbolic spatial metric, which lead to poor model generalization and easy data distortion for samples with insufficient feature information, this paper proposes a knowledge graph completion method based on hyperbolic representation learning and contrastive learning, called Att-CL. The overall framework is as follows for all triples in the knowledge graph $S(h, r, t)$, positive sample pairs are constructed with the same relationship or the same entity–relationship pair based on the head entity or tail entity, and mismatched entity–relationship combinations are randomly selected from the knowledge graph to construct adversarial sample pairs. To improve the representativeness of the embedding vectors, Att-CL employs the AttH method for hyperbolic representation learning. In this method, the samples are transformed by rotation and reflection operations. An attention mechanism is introduced to weigh the vectors, which are summed with the relation vectors to form the final aggregated vectors. The scores $f(h, r, t)$ between pairs of samples and the loss function $\mathscr{L}_{Att}$ are computed. To cope with the situation in which the number of relationally connected entities in the knowledge graph is low, Att-CL makes a judgement on each triple. Suppose that the number of connected entities is lower than the average. In that case, adversarial samples are generated via the introduction of adversarial noise for data augmentation, and the generated adversarial samples are added to the sample pairs for contrastive learning.

During contrastive learning, a penalty term is introduced into the contrastive loss function of Att-CL, and the weight λ_r of the penalty term is adjusted to control the hyperbolic distances of the embedding vectors from the origin to avoid distortion of data far from the origin due to the influence of the hyperbolic spatial metric. Finally, by means of the contrastive loss function, Att-CL maps positive samples into the hyperbolic space to obtain close features. At the same time, as much information as possible is retained in the features, ensuring that this information is uniformly distributed in the hyperbolic space. The overall training objective $\mathscr{L}_{Att-cl}$ is defined as:

$$\mathscr{L}_{Att-cl} = \mathscr{L}_{Att} + \mathscr{L}_{cl} \tag{21}$$

where $\mathscr{L}_{Att}$ is the loss function used in the adversarial sample data enhancement method based on hyperbolic representation learning, as presented in Section 3.2, and $\mathscr{L}_{cl}$ is the comparative loss function introduced in Section 3.3. The overall framework of Att-CL is shown in Fig. 2, and the Att-CL model algorithm is given below.

Algorithm 3: Att-CL Framework

Input: Knowledge graph triplets $S(h, r, t)$, curvature c, K, N, $|X|$, Avg_rel_num n, dataset, ε, α, ρ
Output: Overall Loss $\mathscr{L}_{Att-cl}$
Step 1: Hyperbolic embedding vector by ATTH;
$q(h, r) \leftarrow ATTH(h, r, t)$;
Step 2: Generate Adversarial Samples using GAS;
$\text{Adversarial\_samples} \leftarrow \text{GAS}(q, n, dataset, \varepsilon, \alpha)$;
$\mathscr{L}_{Att} \leftarrow \text{Calculate Adversarial Loss}(q, \text{adversarial\_samples})$;
Step 3: Calculate Contrastive Loss using CLHS;
$\mathscr{L}_{cl} \leftarrow \text{CLHS}(\text{adversarial\_samples}, \text{c}, \text{K}, |X|)$;
Step 4: Overall Loss Loss $\mathscr{L}_{Att-cl}$;
$\mathscr{L}_{Att-cl} \leftarrow \mathscr{L}_{Att} + \mathscr{L}_{cl}$;
return $\mathscr{L}_{Att-cl}$

4. Experimental analysis

4.1. Datasets

In this paper, we consider two benchmark datasets commonly used for knowledge graph completion tasks, namely, WN18RR [28] and FB15k-237 [29]. The information characteristics of each dataset are shown in Table 1. The link prediction task was used in experiments to evaluate the proposed method and verify its generality over different knowledge graphs. The WN18RR dataset is a rearranged version of the WordNet [30] dataset, used for the WordNet knowledge graph completion task. It contains 18 relationship types (Relations) and 40,943 entities (Entities). Compared to the original WN18 dataset, the WN18RR dataset has been rearranged to increase the task difficulty by eliminating simple triple violations and increasing the number of negative samples in the training set, making it more consistent with real-world knowledge graph completion tasks. The FB15K-237 dataset is a subset of Freebase [31] that contains 237 relationship types and 14,505 entities. FB15K-237 is a streamlined version of the original FB15K dataset obtained by removing some uncommon relationship types and entities to reduce the complexity and size of the dataset.

4.2. Evaluation metrics and parameters

For a comprehensive evaluation of model performance, several evaluation metrics were used in the experiments, including the following main metrics:

Mean reciprocal rank (MRR): MRR serves as a vital evaluation metric for gauging a model's ranking quality for correct answers within the link prediction task.

Table 1
Information on the FB15k-237 dataset and the WN18RR dataset (ρ denotes the average number of entities per relational connection).

dataset	|V|	$\|R\|$	training set	validation set	test set	total triples	ρ
FB15K-237	14,505	237	272,115	17,535	20,466	310,116	3.57
WN18RR	40,943	18	86,835	3034	3134	93,003	4.62

Table 2
Parameters.

Parameter name	Notation	Parameter value
Number of convolutional layers	L	1
Gradient control factor	c	0.003
Adversarial perturbation control factor	α	0.6
Perturbation magnitude	ε	0.1
Vector dimensionality	d	32
Batch size	B	512
Number of model iterations	M	600
Learning rate	γ	0.01

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{22}$$

where $\frac{1}{rank_i}$ denotes the inverse rank of the i-th sample and N denotes the number of test samples. The higher the positions of the correct answers in the sorted list and the greater their inverse ranks, the higher the model's performance.

Top-k accuracy (Hits@k): Hits@k is another important evaluation metric that measures the accuracy of a model in top-k prediction.

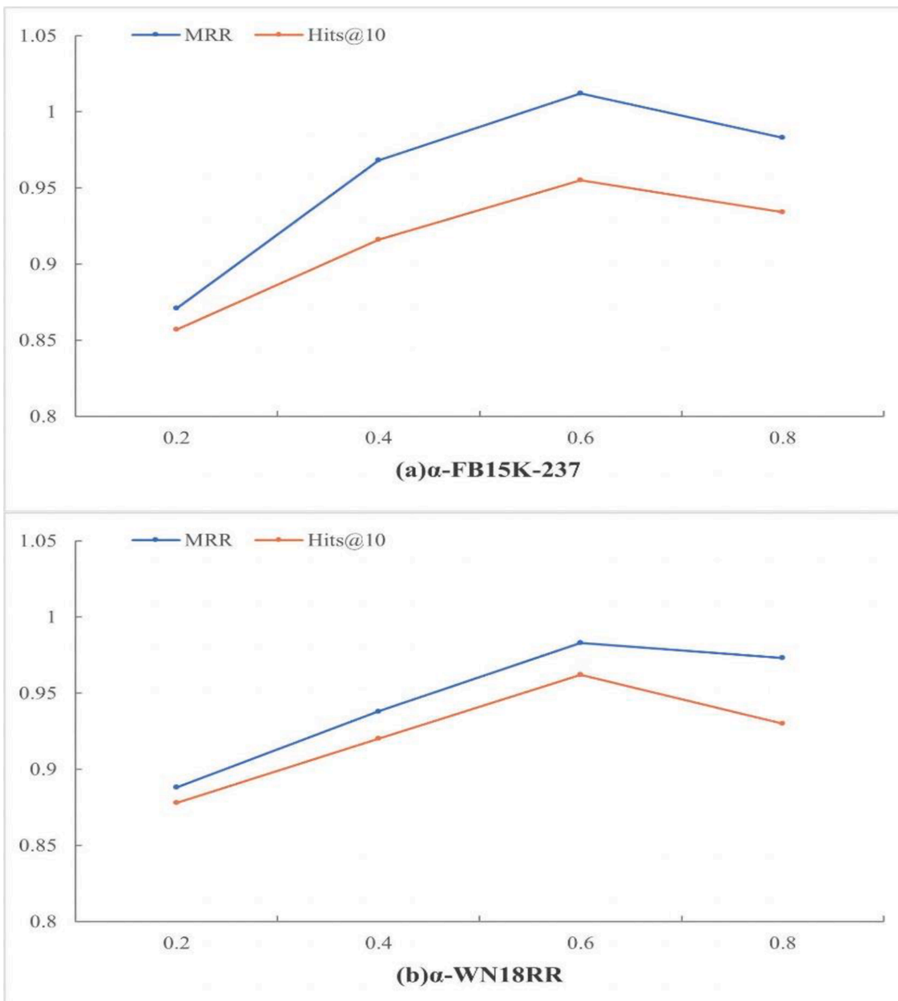

Fig. 3. Parameter sensitivity analysis: (a) FB15K-237, (b) WN18RR.

Table 3
Knowledge graph link prediction results for the FB15k-237 dataset and WN18RR dataset.

Model	FB15K-237				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
TransE	0.207	0.186	0.321	0.377	0.226	0.427	0.441	0.501
DistMult	0.241	0.155	0.263	0.419	0.430	0.390	0.440	0.490
ConvE	0.325	0.237	0.356	0.501	0.430	0.400	0.440	0.520
MuRE	0.336	0.245	0.370	0.521	0.459	0.436	0.487	0.528
ComplEx	0.247	0.158	0.276	0.428	0.440	0.410	0.460	0.510
RotatE	0.338	0.241	0.375	0.533	0.476	0.428	0.492	0.571
MuRP	0.336	0.243	0.367	0.521	0.481	0.440	0.495	0.566
ATTH	0.348	0.252	0.384	0.540	0.486	**0.443**	0.498	**0.573**
Att-CL	**0.351**	**0.255**	**0.388**	**0.541**	**0.489**	0.441	**0.501**	0.569

Table 4
Knowledge graph link prediction results for the FB15k-237 dataset and WN18RR dataset (sparse data).

Model	FB15K-237				WN18RR			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
MuRP	0.262	0.195	0.271	0.307	0.298	0.293	0.289	0.304
ATTH	<u>0.274</u>	<u>0.203</u>	<u>0.293</u>	<u>0.312</u>	<u>0.305</u>	<u>0.301</u>	<u>0.295</u>	**0.314**
Att-CL	**0.291**	**0.237**	**0.299**	**0.358**	**0.319**	**0.303**	**0.307**	<u>0.313</u>

$$Hits@k = \frac{1}{N}\sum_{i=1}^{N} I(rank_i \leq k) \tag{23}$$

$$I(rank_i \leq k) = \begin{cases} 1, rank_i \leq k \\ 0, rank_i > k \end{cases} \tag{24}$$

The value of Hits@k is obtained by averaging the values of the indicator function over all samples.

During the experiments, the RSGD [32] optimizer was used for hyperbolic embedding, while the Adam [33] optimizer was employed for contrastive learning. The model's hyperparameters were selected through a grid search and determined based on the best performance on the validation set, as shown in Table 2. When constructing adversarial samples, α is a crucial hyperparameter primarily used to fine-tune the magnitude of the perturbations, thereby controlling the strength of the adversarial samples. This is because excessively strong perturbations might introduce noise, harming the model's representation ability, while overly weak perturbations might not effectively enhance the model's robustness. To validate the optimal α value, experiments were conducted with$\alpha = 0.2, 0.4, 0.6, 0.8$, using MRR and Hits@10 as the assessment criteria. The outcomes of the experiments are depicted in Fig. 3, the results of the experiments suggest that the model achieved optimal performance with $\alpha = 0.6$.

4.3. Results and analysis of experiments

4.3.1. Comparative analysis of overall model performance

To evaluate the knowledge graph completion method combining hyperbolic representation learning and contrastive learning that is proposed in this paper, the following models were selected for comparative experiments: the translation-based TransE and MuRE models in Euclidean space; the ConvE model, based on a convolutional neural network; the tensor decomposition-based DistMult model; the RotatE and ComplEx models in the complex domain; and the MuRP and ATTH models in hyperbolic space. These models represent the most commonly used and advanced methods for knowledge graph completion tasks. You can find the results of our experiments in Table 3. We've highlighted the best outcomes in bold to make them stand out. On the other hand, we've underlined the suboptimal results.

The experimental results provide valuable insights. They reveal that the Att-CL method, which we introduced in this paper, effectively harnesses the strengths of both hyperbolic representation learning and contrastive learning. Compared with the seven models mentioned above, the Att-CL method stands out with substantial enhancements in the majority of evaluation metrics. On average, we observed remarkable improvements of 5.71 % in MRR, 2.98 % in Hits@1, 4.04 % in Hits@3, and 4.88 % in Hits@10. The Att-CL method utilizes hyperbolic isometric transformations (rotations and reflections) and hyperbolic attention to perform hyperbolic embedding. By employing adversarial noise superposition to construct adversarial samples for data enhancement, the hierarchical structure and logical patterns of knowledge graphs can be more effectively captured, making the proposed method especially suitable for processing knowledge graphs with tree-like hierarchical structures. By introducing a penalty term into the contrastive loss function to adjust the hyperbolic distances between the embedding vectors and the origin, similar entities and relations are represented as similar locations in hyperbolic space, while dissimilar entities and relations are represented as more distant locations. In this way, the model's capacity for expression experiences a notable enhancement, and the model becomes better able to distinguish semantic associations between different entities and relations, thus effectively overcoming the data distortion problem suffered by traditional hyperbolic representation learning models, preserving the knowledge graph's global structure and hierarchical relationships, and improving the model's performance in knowledge graph completion. Owing to the comparatively modest dimensions of the WN18RR dataset and the fact that the relations it contains mainly involve associations between word meanings, it is primarily focused on semantic relations. Therefore, the Att-CL model shows greater improvement on the FB15k-237 dataset than on the WN18RR dataset.

4.3.2. Ablation experiments

To explore the efficacy of constructing adversarial samples for data augmentation, assess the model's performance in handling infrequent and uncommon relationships in a dataset, and conduct a comprehensive assessment of the impact of the method for constructing adversarial samples on enhancing the efficiency of knowledge graph completion. This study selects the ATTH, MURP, and Att-CL models for examination. Link prediction experiments were executed on the triples within each dataset, focusing on which relationships with fewer connected entities than the dataset's average. The outcomes are presented in Table 4, with the best results highlighted and the suboptimal results underscored.

The experimental findings reveal that the utilization of adversarial

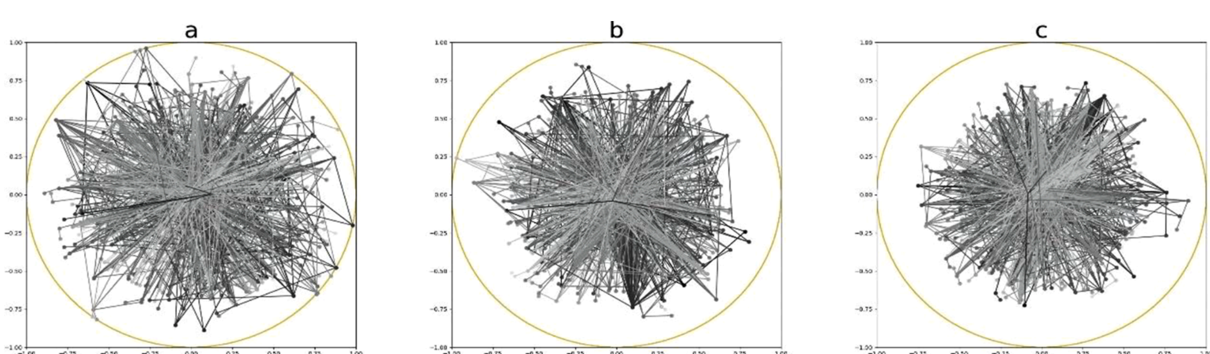

Fig. 4. Embedded vector visualization and analysis diagrams: (a) MURP model, (b) ATTH model, (c) Att-CL model.

Table 5
Comparison of the spatial complexities of knowledge graph completion models.

Method	spatial complexity
TransE	$O(\lvert E\rvert n + \lvert R\rvert n)$
TransH	$O(\lvert E\rvert n + \lvert R\rvert n)$
TransR	$O(\lvert E\rvert n + \lvert R\rvert n^2)$
DistMult	$O(\lvert E\rvert n + \lvert R\rvert n)$
ComplEx	$O(2\lvert E\rvert n + 2\lvert R\rvert n)$
RESCAL	$O(\lvert E\rvert n + \lvert R\rvert n^2)$
RotatE	$O(\lvert E\rvert n + \lvert R\rvert n)$
Att-CL	$O(\lvert E\rvert n + \lvert R\rvert n)$

sample construction for data augmentation effectively enhances the model's capacity to learn from long-tailed data and improves its generalization capabilities. In comparison to both the ATTH and MURP models, which generate affirmative and negative samples, the technique of data enhancement through adversarial sample construction exhibits a pronounced advantage, resulting in average enhancements of 2.03 %, 2.2 %, 1.6 %, and 2.62 % in MRR, Hits@1, Hits@3, and Hits@10, respectively. Furthermore, further enhancement of the embedding space by integrating hyperbolic representation learning and contrastive learning mechanisms can yield superior results and performance in the task of completing knowledge graphs.

In addition, this paper presents visualizations to verify the effectiveness of introducing a penalty term in the contrastive loss function to control the embedding vectors' distances from the origin to avoid node distortion. Fig. 4 illustrates the relevant results.

The MURP model (Fig. 4, a), ATTH model (Fig. 4, b) and Att-CL model (Fig. 4, c) were selected for experiments to compare the resulting embedding vector distortion on the FB15K-237 dataset, and the embedded nodes were mapped onto Poincaré disc for visualization. In hyperbolic space, the radius of the Poincaré disc grows exponentially. Usually, nodes with rich entity–relationship feature information are located near the centre of the Poincaré disc, and the model is fully trained on such nodes. In contrast, nodes representing sparse and long-tailed data are located closer to those with similar angular coordinates. As seen in Fig. 4, the problems of imbalance, undertraining, and hyperbolic embedding distortion observed on the long-tailed data in the sparse FB15K-237 data are greatly mitigated by the construction of adversarial samples for data enhancement as well as the addition of the proposed penalty term in the loss function. These modifications enhance the variety of the training data and, to some degree, enhance the model's capacity for generalization, avoiding the construction of excess outlier pairs of samples in the embedded hyperbolic space, which can lead to node distortion. With the combination of these two approaches, the model pays more attention to samples in the long tail of the distribution during training, thereby balancing the sample distribution, improving the embedding of such long-tail samples, and improving their prediction accuracy.

4.4. Complexity analysis

Table 5 shows the spatial complexity of the Att-CL model compared to several popular models. It can be observed that Att-CL has a similar spatial complexity compared to classical KGC models like TransE, DistMult, and RotatE.

5. Conclusion

Existing hyperbolic representation learning models can preserve the hierarchical structure and logical patterns of nodes in the knowledge graph completion task but still suffer from the long-tail problem and node distortion. This paper introduces the Att-CL model, which integrates hyperbolic representation learning with contrastive learning to more accurately preserve the hierarchical structure and logical patterns of knowledge graphs, thereby improving the effectiveness of knowledge graph completion. Data augmentation is achieved by constructing adversarial samples and increasing the quantity of training samples corresponding to the long tail of the data distribution. This enhancement aims to improve the model's capacity to generalize when faced with long-tailed data, and a penalty term is introduced in the contrastive loss function to effectively avoid excessive deviation of adversarial samples and long-tail data in the hyperbolic space and thus mitigate the node distortion problem. The results of experiments conducted on various datasets consistently demonstrate that this method performs stably and reliably on different datasets and achieves improvements relative to mainstream benchmark algorithms, especially on the FB15k-237 dataset, which has a more pronounced degree of sparsity. In subsequent research, we will compare the effects of multiple different contrastive learning methods in terms of model performance to search for a more suitable contrastive learning strategy for the task of knowledge graph completion, and we will attempt to apply the model introduced in this paper within various contexts, such as recommender systems, text generation, and significant language models to further improve the application scope of this model.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank all those that have contributed to our projects over the years, and the reviewers whose critical feedback helped to improve this article.

Funding

This study was supported by The National Key Research and Development Program of China (No. 2019YFB1404602) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (SJCX23_1096).

Ethical statement

Not applicable

References

[1] Ji S, Pan S, Cambria E, Marttinen P, Philip SY. A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Trans Neural Netw Learn Syst 2021;33:494–514.
[2] Chen X, Jia S, Xiang Y. A review: Knowledge reasoning over knowledge graph. Expert Syst Appl 2020;141:112948.
[3] Zamini M, Reza H, Rabiei M. A review of knowledge graph completion. Information 2022;13:396.
[4] Chen Z, Wang Y, Zhao B, Cheng J, Zhao X, Duan Z. Knowledge graph completion: A review. IEEE Access 2020;8:192435–56.
[5] Shen T, Zhang F, Cheng J. A comprehensive overview of knowledge graph completion. Knowl Based Syst 2022:109597.
[6] Dhingra B, Shallue CJ, Norouzi M, Dai AM, Dahl GE. Embedding text in hyperbolic spaces. ArXiv Preprint ArXiv:180604313 2018.
[7] Moon W, Kim J-H, Heo J-P. Tailoring self-supervision for supervised learning. Eur Conf Comput Vis 2022:346–64.
[8] Purushwalkam S, Morgado P, Gupta A. The challenges of continuous self-supervised learning. Eur Conf Comput Vis 2022:702–21.
[9] Wang Z, Lai KP, Li P, Bing L, Lam W. Tackling long-tailed relations and uncommon entities in knowledge graph completion. ArXiv Preprint ArXiv:190911359 2019.
[10] Tiddi I, Schlobach S. Knowledge graphs as tools for explainable machine learning: A survey. Artif Intell 2022;302:103627.
[11] Wang Q, Mao Z, Wang B, Guo L. Knowledge graph embedding: A survey of approaches and applications. IEEE Trans Knowl Data Eng 2017;29:2724–43.
[12] Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. Adv Neural Inf Process Syst 2013;26.

[13] Yang B, Yih W, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. ArXiv Preprint ArXiv:14126575 2014.
[14] Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. Complex embeddings for simple link prediction. Int Conf Mach Learn 2016:2071–80.
[15] Chami I, Ying Z, Ré C, Leskovec J. Hyperbolic graph convolutional neural networks. Adv Neural Inf Process Syst 2019;32.
[16] Dettmers T, Minervini P, Stenetorp P, Riedel S. Convolutional 2d knowledge graph embeddings. Proceedings of the AAAI conference on artificial intelligence, vol. 32, 2018.
[17] Peng W, Varanka T, Mostafa A, Shi H, Zhao G. Hyperbolic deep neural networks: A survey. IEEE Trans Pattern Anal Mach Intell 2021;44:10023–44.
[18] Sun Z, Chen M, Hu W, Wang C, Dai J, Zhang W. Knowledge association with hyperbolic knowledge graph embeddings. ArXiv Preprint ArXiv:201002162 2020.
[19] Balazevic I, Allen C, Hospedales T. Multi-relational poincaré graph embeddings. Adv Neural Inf Process Syst 2019;32.
[20] Nickel M, Kiela D. Poincaré embeddings for learning hierarchical representations. Adv Neural Inf Process Syst 2017;30.
[21] Chami I, Wolf A, Juan D-C, Sala F, Ravi S, Ré C. Low-dimensional hyperbolic knowledge graph embeddings. ArXiv Preprint ArXiv:200500545 2020.
[22] Wang X, He X, Cao Y, Liu M, Kgat C-S. Knowledge graph attention network for recommendation. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019. p. 950–8.
[23] Wang T, Isola P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. Int Conf Mach Learn 2020:9929–39.
[24] Jaiswal A, Babu AR, Zadeh MZ, Banerjee D, Makedon F. A survey on contrastive self-supervised learning. Technologies (Basel) 2020;9:2.
[25] Su Y, Lan T, Wang Y, Yogatama D, Kong L, Collier N. A contrastive framework for neural text generation. Adv Neural Inf Process Syst 2022;35:21548–61.
[26] Wang L, Zhao W, Wei Z, Liu J. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. ArXiv Preprint ArXiv:220302167 2022.
[27] Luo Z, Xu W, Liu W, Bian J, Yin J, Liu T-Y. KGE-CL: Contrastive learning of tensor decomposition based knowledge graph embeddings. ArXiv Preprint ArXiv: 211204871 2021.
[28] Han X, Cao S, Lv X, Lin Y, Liu Z, Sun M, et al. Openke: An open toolkit for knowledge embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2018. p. 139–44.
[29] Toutanova K, Chen D. Observed versus latent features for knowledge base and text inference. Proceedings of the 3rd workshop on continuous vector space models and their compositionality, 2015, p. 57–66.
[30] Miller GA. WordNet: a lexical database for English. Commun ACM 1995;38:39–41.
[31] 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; 2008. p. 1247–50.
[32] Bécigneul G, Ganea O-E. Riemannian adaptive optimization methods. ArXiv Preprint ArXiv:181000760 2018.
[33] Kingma DP, Ba J. Adam: A method for stochastic optimization. ArXiv Preprint ArXiv:14126980 2014.