



Resource Recommendation Based on Industrial Knowledge Graph in Low-Resource Conditions

Yangshengyan Liu^{1,2} · Fu Gu^{1,2,3,4} · Xinjian Gu^{1,2,3} · Yijie Wu¹ · Jianfeng Guo^{5,6} · Jin Zhang^{1,2,3}

Received: 12 July 2021 / Accepted: 10 June 2022
© The Author(s) 2022

Abstract

Resource recommendation is extremely challenging under low-resource conditions because representation learning models require sufficient triplets for their training, and the presence of massive long-tail resources leads to data sparsity and cold-start problems. In this paper, an industrial knowledge graph is developed to integrate resources for manufacturing enterprises, and we further formulate long-tail recommendations as a few-shot relational learning problem of learning-to-recommend resources with few interactions under low-resource conditions. First, an industrial knowledge graph is constructed based on the predesigned resource schema. Second, we conduct schema-based reasoning on the schema to heuristically complete the knowledge graph. At last, we propose a multi-head attention-based meta relational learning model with schema-based reasoning to recommend long-tail resources under low-resource conditions. With the IN-Train setting, 5-shot experimental results on the NELL-One and Wiki-One datasets achieve average improvements of 28.8 and 13.3% respectively, compared with MetaR. Empirically, the attention mechanism with relation space translation learns the most important relations for fast convergence. The proposed graph-based platform specifies how to recommend resources using the industrial knowledge graph under low-resource conditions.

Keywords Resource recommendation · Industrial knowledge graph · Few-shot relational learning · Schema-based reasoning

1 Introduction

The advent of Industry 4.0 requires the smartization of industrial processes, extending smart manufacturing to smart storage and smart logistics [1]. Industrial resources are the

basis of the development and reconstruction of value chains, and the integration and sharing of such resources become increasingly important in the dawn of Industry 4.0 [2].

With the highly specialized division of labor and the continuous upgrading of products, industrial resources experience exponential growth in the entire product life cycle [3]. This growth inevitably leads to several problems. First, scattered systems impede enterprise collaboration, and the innovation and safety of the manufacturing value chain are difficult to guarantee [4]. Second, there is a lack of integration models for product innovation and resource scheduling. At last, massive long-tail resources (i.e., the resources are rarely referenced within a knowledge graph) cause the cold-start problem (i.e., the engine cannot make reliable recommendations due to an initial lack of ratings) in most of the reported recommendation engines [5]. These problems compromise the robustness and generalization of recommender systems, and thereby hinder the sharing and reuse of industrial resources throughout the product life cycle.

The previous works focus on utilizing knowledge graphs to construct interpretable recommender systems [6–8]. Yet,

✉ Fu Gu
gufu@zju.edu.cn

¹ State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

² Department of Industrial and System Engineering, Zhejiang University, Hangzhou 310027, China

³ Key Laboratory of Advanced Manufacturing Technology of Zhejiang Province, Zhejiang University, Hangzhou 310027, China

⁴ Center of Engineering Management, Polytechnic Institute, Zhejiang University, Hangzhou 310015, China

⁵ School of Public Policy and Management, University of Chinese Academy of Sciences, Beijing 100049, China

⁶ Institutes of Science and Development, Chinese Academy of Sciences, Beijing 100190, China

till now, insufficient attention has been paid to building recommender systems for the value chains in the manufacturing industry. Despite several existing works using auxiliary information to overcome the data sparsity problem of user-item interactions [9], their performance is far from satisfaction because auxiliary information is always unavailable for newly arising data and long-tail data [10]. Smart resource aggregation and collaborative recommendation require deep learning models [11, 12], while these models are usually suboptimal under low-resource conditions.

To address the above challenges, a proper meta-model is needed to integrate and share heterogeneous industrial resources, where sufficient annotated data for training is always unavailable. In this article, a novel framework for industrial resource recommendation in low-resource conditions is proposed to improve the seamless collaboration between enterprises and thereby promote the production and operation efficiency of manufacturing enterprises.

Our method consists of the following steps. First, scattered industrial resources are integrated and correlated to form an industrial knowledge graph based on the resource schema. Second, to tackle the sparsity within the graph, we conduct schema-based reasoning to identify the potential implicit relations and heuristically complete the graph. Lastly, we propose a novel multi-head attention-based meta relational learning model that learns the latent representations of relation-meta to solve the cold-start problem caused by the long-tail resources. The predicted links are used to recommend industrial resources to corresponding entity nodes.

Our main contributions are summarized as follows.

- 1) We propose an industrial knowledge graph model to integrate heterogeneous resources throughout the product life cycle. We further formulate long-tail recommendations as a few-shot relational learning problem of learning-to-recommend resources with few interactions.
- 2) We conduct schema-based reasoning to mine the potential implicit relations, as well as to complete the industrial knowledge graph.
- 3) We propose a novel multi-head attention-based meta relational learning model to improve the use of long-tail resources and to address the cold-start problem.
- 4) We develop a graph-based platform in which the proposed recommendation algorithm is incorporated to recommend industrial resources in low-resource conditions. A business metric, net promoter score, is adopted to evaluate the recommender systems in low-resource conditions.

The rest of the paper is organized as follows. Section 2 presents the literature review on related methods. Our method is illustrated in Sect. 3. Section 4 reports the

experimental results. Section 5 displays analysis and discussion. Section 6 concludes this work.

2 Background

2.1 Knowledge Graph-Based Recommendation

Recently, personalized recommender systems based on knowledge graphs are proposed to improve performance and provide interpretability. In general, the recommender algorithms can be categorized into three classes, i.e., feature-based methods, path-based methods, and embedding-based methods [13].

The feature-based methods, as their name indicate, extract the features of users and/or items for the machine learning process [14]. Sohail et al. [15] present an opinion mining-based recommendation technique to provide the university students with promising books for their syllabus; however, this technique could be subjectively biased. Uyangoda et al. [16] apply a user-profile-feature-based approach to improve the recommender system with few user records. Dai et al. [17] propose a feature-based bayesian task recommendation scheme to overcome the challenge of emerging recommendations, but the scheme cannot address the changes in users' interests. Yang et al. [18] propose a meta-feature-based approach, called Explainable Recommendation Framework, to render explainable recommendations for both warm-start and cold-start users/items in a unified framework. Yet, these methods typically neglect the implicit information in graph structures [19]. The structural information in the graph could contribute to capturing the pairwise relations [20], and therefore modeling structural patterns is beneficial to recommender systems [21, 22].

The path-based methods learn the paths between users and items to construct the knowledge graph. Yu et al. [23] investigate the entity recommendation problem in heterogeneous information networks and propose the use of implicit feedback data for developing personalized recommendation models. Zhao et al. [24] introduce the concept of meta-graph to HIN-based recommendation and solve the information fusion problem with a novel approach. Unfortunately, these methods still exhibit poor generalization performance due to the manually designed paths.

The embedding-based methods embed users and items to a low vector space to improve the accuracy of recommendations. Wang et al. [25] develop a deep knowledge-aware network that incorporates knowledge graphs into news recommendations. Zhao et al. [8] leverage heterogeneous information in a knowledge base to improve the recommendation quality. Ripple Network [26] stimulates the propagation of user preferences over knowledge entities by automatically and iteratively extending a user's potential interests along

with links in the knowledge graph. Multi-task feature learning approach [27] is a deep end-to-end framework that uses knowledge graph embeddings to assist resource recommendation. Yet, these embedding-based methods have limited applicability under low-resource conditions, see Sect. 4.2.2.

How to improve the performance of recommender systems based on knowledge graphs is a recent research hotspot. To resolve these problems, researchers have incorporated auxiliary information into recommendation algorithms. These works mine users' historical records for implicit information and improve the model structure for better performance. Contrarily, data sensitivity and knowledge specificity of industrial resources require massive labeled data for training, but manual tagging is extremely costly and time-consuming. In addition, owing to their multi-disciplinary and heterogeneous nature, industrial resources tend to be highly complicated and professional to construct knowledge graphs for the manufacturing value chains, especially in low-resource conditions.

2.2 Industrial Knowledge Graph

There are several well-known knowledge graphs such as CN-DBpedia [28] and Wikidata [29], but limited attention has been paid to knowledge graphs in industrial resource integration such as procurement and manufacturing. The current research on industrial knowledge graphs can be categorized into three parts, including the construction of industrial knowledge graphs, knowledge deduction, and applications of industrial knowledge graphs.

The construction of industrial knowledge graphs aims to aggregate massive data from industrial products and services to create artificial intelligence for industrial applications [30]. Knowledge extraction technologies, such as data mining, natural language processing, and deep learning, are designed to extract entities and relations from unstructured industrial resources [31–33]. Researchers also enhance industrial knowledge graphs with generic knowledge graphs and exploit the storage of industrial knowledge graphs [34–37]. However, at present, industrial knowledge graphs are still entangled with the emerging resources, the knowledge mining of possible implicit relations between resources, and the collaboration of domain experts within the graph construction [38].

The knowledge deduction of industrial knowledge graphs focuses on the multi-hop semantic search and knowledge reasoning based on the industrial knowledge graphs [39–41]. The industrial knowledge deduction can be categorized into attribute deduction and relationship deduction. Practically, these methods map attributes and relations to a low-dimensional vector space and transform the knowledge deduction process into matrix operations [42–45]. Nevertheless, knowledge deduction also requires massive labeled data for

training [38]. Moreover, the few-shot problems under low-resource conditions are usually overlooked.

Industrial knowledge graphs can be applied in a series of scenarios like industrial collaboration, resource representation learning, and intelligent search. The typical representative applications include question answering with knowledge graphs [46], visualization of knowledge graphs [47], and fault diagnosis [48]. However, these practical implementations impose constraints on the links in knowledge graphs by defining a schema or an ontology [49]. Worse still, knowledge representation is always much more complicated in the manufacturing industry [50].

2.3 Few-Shot Link Prediction

Knowledge deduction based on knowledge graph embedding [51] (KGE) usually assumes that there are sufficient triples of entities and relations for training. However, the applicability of this approach is limited due to the two following aspects. First, long-tail resources are widespread in knowledge graphs, and those newly added relations often do not have many known samples for training [52]. Second, emerging resources in the manufacturing industry also cause the few-shot learning problem, while such resources tend to be ignored in prior works [53].

Compared with the methods that require sufficient training data, the performance of few-shot learning problems is poorer [54]. Further, embedding-based methods for few-shot link prediction always perform awkwardly on relations that only have a few associative triples [55]. Therefore, recommender systems are unable to make reliable recommendations due to these long-tail constraints, also known as the cold-start problem.

In sum, the industrial knowledge graph is a promising solution to the integration, sharing, and management of domain knowledge, while industrial resources require a more canonical and formal form to fuse heterogeneous and multi-disciplinary resources for cooperation. Still, the industrial recommender systems are suffering from the cold-start problem caused by the newly arising resources and the long-tail resources.

3 Method

Here, a novel industrial knowledge graph is constructed based on the predefined resource schema. We further conduct rule reasoning on the schema to heuristically complete the knowledge graph. The inference rules are added to the support sets to train the multi-head attention-based meta relational learning algorithm. The industrial knowledge graph aims at integrating resources across the entire product life cycle, and our meta relational learning model

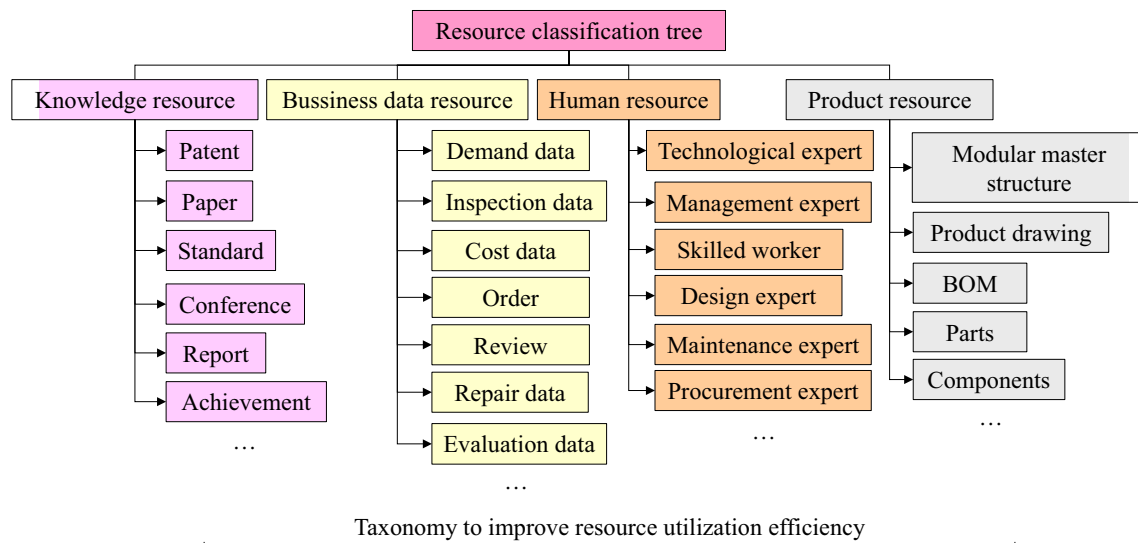


Fig. 1 Resource classification tree

is developed to recommend resources under low-resource conditions.

3.1 Industrial Knowledge Graph

3.1.1 Industrial Knowledge Graph Construction

The industrial knowledge graph is a promising solution to integrate industrial resources and enhance knowledge sharing across different manufacturing sectors [38]. In this paper, an industrial knowledge graph is constructed to support our recommendation algorithm in low-resource conditions. Hence, it is feasible to apply our method to other domains by modifying the resource schema and conducting rule-based reasoning with the proposed meta relational learning algorithm. From Fig. 1, the resource schema is based on the resource classification tree that categorizes industrial resources into four classes, namely knowledge resources, business data resources, human resources, and product resources.

Specifically, knowledge resources, including patents, standards, papers, achievements, regulations, and reports, enhance the knowledge service capabilities for product innovations [56]. Business data resources are accumulated from business processes, from user demands to records of maintenance. These data comprise the implicit and empirical knowledge for intelligent manufacturing. Human resources refer to the information of experts who ensure the orderly management of other resources. Product resources are the entities of modules that enable manufacturing enterprises to focus on the interplay between innovation and design processes, and they also embed a co-creation paradigm between firms and customers for mass customization [57].

The overview of the industrial knowledge graph construction process is shown in Fig. 2. Specifically, the resource schema [58] is designed based on the resource classification tree that describes the hierarchy concepts and their relations. Open Information Extraction (OpenIE) annotator [59] is used to extract open-domain relation triples within structured and unstructured data. Neo4j JDBC driver [35] and RDF2Neo4j interpreter [60] are employed for data mapping, and attribute-based fusion [61] is used to fuse industrial resources from scattered relational databases in a unified paradigm. In such a paradigm, the entity set represents heterogeneous resources, and the edge set represents the relations among industrial resources. The information and knowledge in resources are set as properties of the nodes. The industrial knowledge graph is then integrated and stored in the graph database Neo4j [35]. For KGE, entities and relations form triples that are embedded in a low-dimensional vector space using TransE [62]. Further, the community detection algorithm Cluster-GCN [63] and the few-shot multi-hop reasoning algorithm Meta-KGR [64] are used to support schema-based reasoning (see Sect. 3.1.2). Our few-shot relational learning algorithm (see Sect. 3.2) is proposed to complete the industrial knowledge graph and recommend industrial resources in low-resource conditions. Lastly, a graph-based platform that provides intelligent services like our recommendation engine is developed (as shown in Sect. 4.2).

3.1.2 Schema-Based Reasoning

Schema-based reasoning is an inductive reasoning process that represents a generalized form of case-based reasoning [65]. We conduct schema-based reasoning to complete the industrial knowledge graph and resolve the cold-start

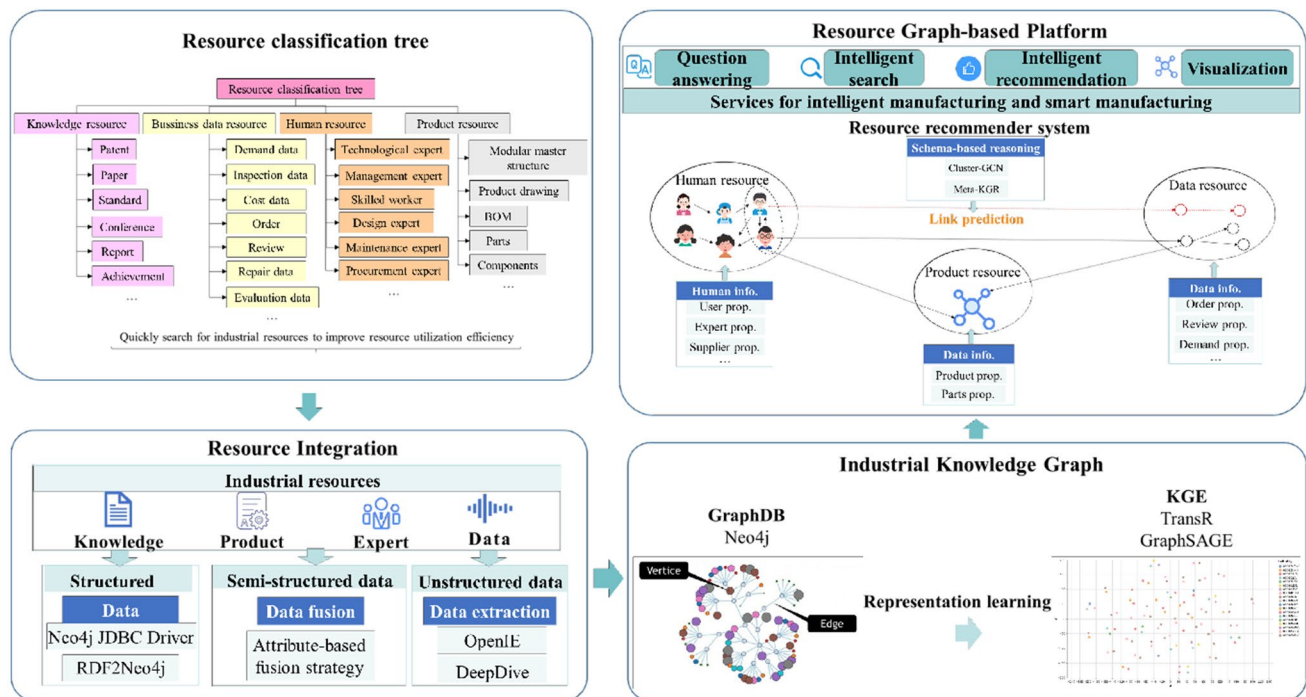


Fig. 2 Overview of the industrial knowledge graph construction

Table 1 Schema-based Reasoning

Rule class	Condition	Result
Historical rule	If b and b' are similar, and a cooperated with b before	Then b' is linked to a
Performance rule	If c and c' perform similarly, and c meets a 's needs	Then c' is linked to a
Community rule	If a , b , and b' are in the same community	Then b and b' are linked to a
Path rule	If a and b are partners, b and b' are also partners	Then b' is linked to a

problem by generating more training data for the few-shot link prediction in Sect. 3.2.

Schema-based reasoning on industrial knowledge graph can be categorized into four classes, namely historical rules, performance rules, community rules, and path rules. The symbols in Table 1 are defined as follows. Assuming a is a manufacturing enterprise that would benefit from the industrial knowledge graph. b and b' are potential suppliers of a . c and c' are parts and components that are required by a .

By their definitions, (1) historical rules associate newly added resources based on the historical or empirical data, i.e. if the enterprise a cooperated with the supplier b in the past, and the supplier b and b' are similar in production, then b' would be linked to a . (2) Performance rules are proposed to link candidate resources that achieve comparable performance.

For instance, if the product c' achieves comparable machining accuracy with the product c that is required in the production process of the manufacturing enterprise a , then c' would be linked to a . (3) Community rules are introduced to associate resources in the same community based on graph structure information, i.e. if industrial enterprises a , b , and b' are densely connected in the knowledge graph, then b and b' would be linked to a . (4) Path rules are generated based on the path in the graph to discover long-tail resources for multi-hop reasoning, i.e. if enterprises a and b are partners, b and b' are also partners, then b' would be linked to a .

In sum, schema-based reasoning can be used to address the sparsity issue of industrial knowledge graphs. Such rules are competent to discover and unveil the implicit relations among resources and associate long-tail resources to complete the

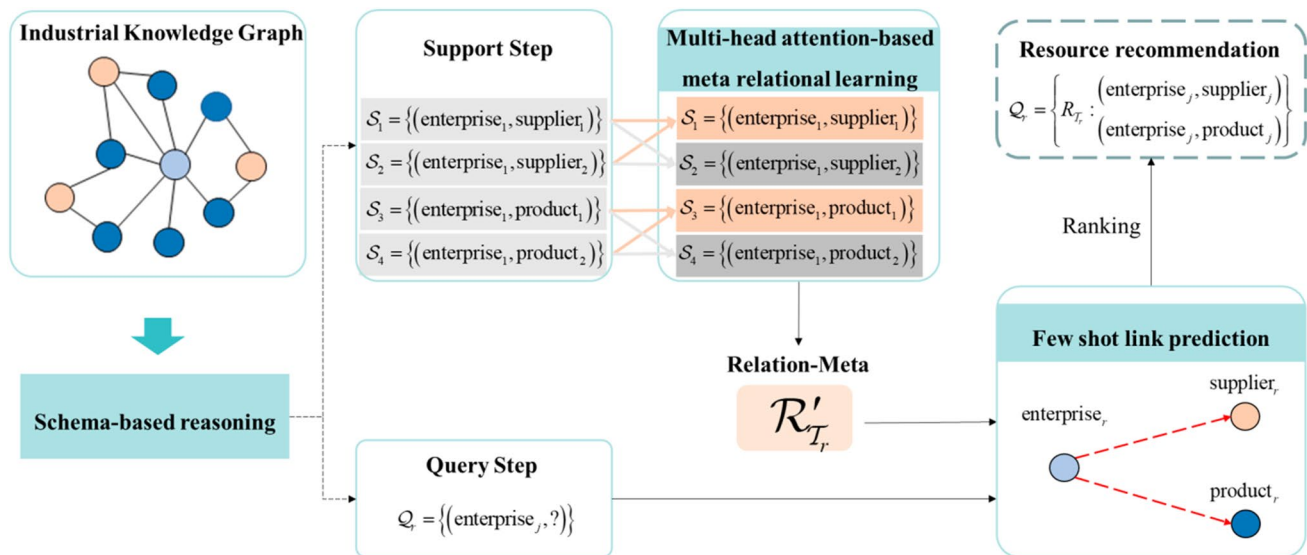


Fig. 3 Resource recommendation algorithm based on the few-shot link prediction

industrial knowledge graph. Moreover, schema-based reasoning generates more training data for few-shot link prediction [55], which is regarded as a promising solution to the cold-start problem in recommender systems [66, 67].

3.2 Multi-head Attention-Based Meta Relational Learning Algorithm with Schema-Based Reasoning

The sparsity of knowledge graphs brings about long-tail data that are the principal cause of the cold-start problem of recommendation algorithms. To fix this problem, we conduct schema-based reasoning to generate more training data and employ the few-shot relational learning on the industrial knowledge graph, which transfers the recommendation task to a simple ranking problem of candidate predictions in low-resource conditions.

An example of the recommendation algorithm based on the few-shot link prediction is shown in Fig. 3. First, the schema-based reasoning strategy is conducted on the industrial knowledge graph to generate more data in the support set and query set for few-shot learning. The meta relational learning model is further proposed to learn the relation-meta. Finally, a recommendation engine that is based on the rankings of predicted links among long-tail resources is introduced to resolve the cold-start problem.

Different from traditional deep learning models that are slow and highly computational expensive, few-shot link prediction aims to gain the capability of predicting new triples about a specific relation by only observing a few triples. Hence, our model is designed to predict any newly added relations without fine-tuning, while existing models

always require massive training data to adapt to newly added relations.

The structure of the multi-head attention-based meta relational learning is shown in Fig. 4, in which a representation layer, a relation-meta encoder, and a training module are included. The representation layer maps entities and relations to a hyper embedding space. A relation-meta encoder is proposed to learn a mapping from head entities to corresponding tail entities in the support set. The training module constructs the objective function and updates the relation-meta that is transferred from the support set to the query set, enabling our model to address the few-shot link prediction.

3.2.1 Representation Layer

Within a support set \mathcal{S}_r and a query set \mathcal{Q}_r , a fact (h_i, t_i) is defined as an entity pair that comprises the head entity h_i and the tail entity t_i . The corresponding relation in the fact is r_i . Entities and relations are first embedded in a hyper vector space, either by random initialization or pre-trained embeddings. [54] Given one few-shot link prediction task $\mathcal{T}\Theta_r = \{\mathcal{S}_r, \mathcal{Q}_r\}$, each fact (h_i, t_i) of the task $\mathcal{T}\Theta_r$ is mapped to get the embeddings of entities and corresponding relations in Eq. (1).

$$\begin{aligned} x_{hi} &= e(h_i) \\ x_{ri} &= e(r_i) \\ x_{ti} &= e(t_i) \end{aligned} \quad (1)$$

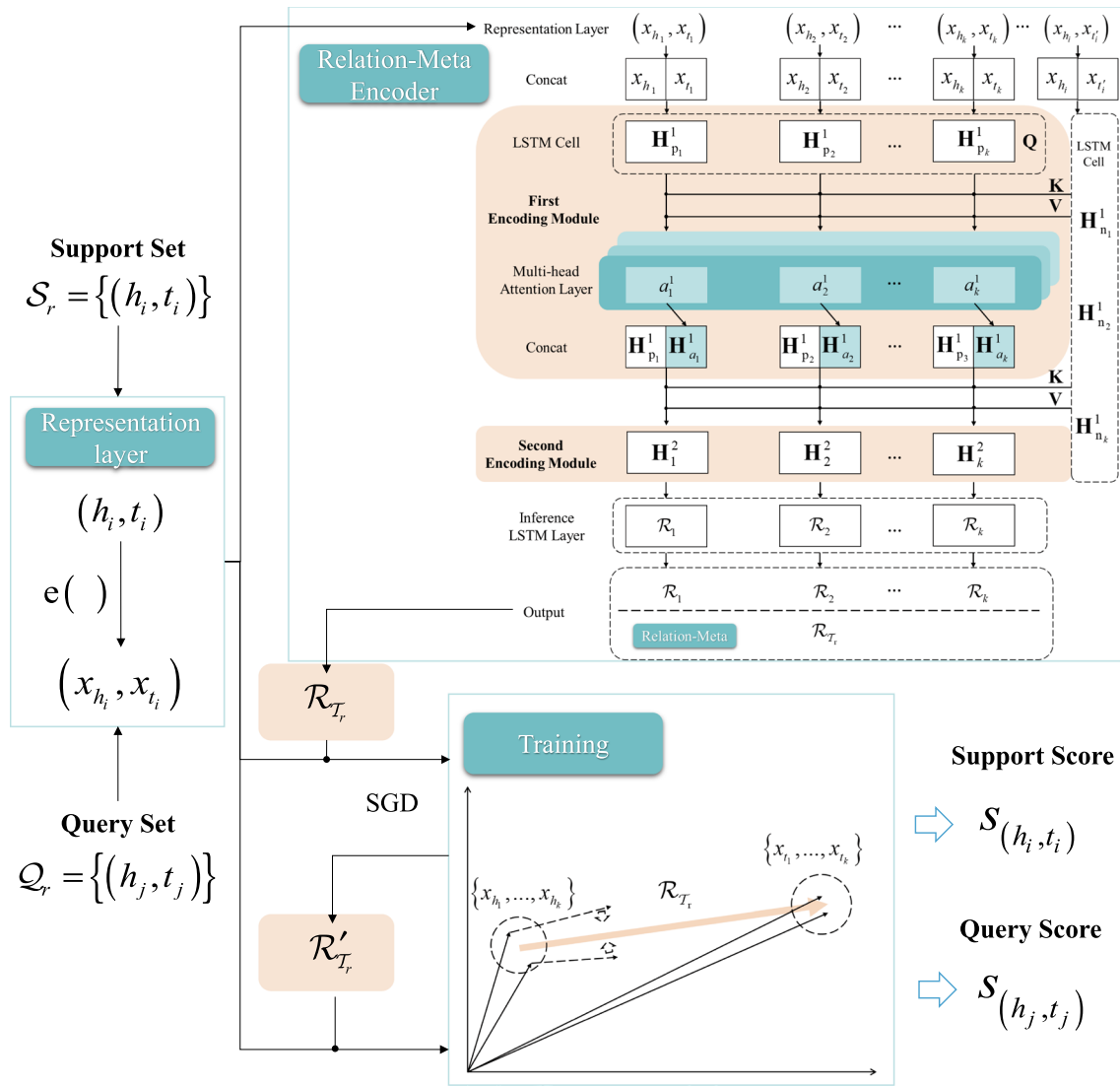


Fig. 4 Multi-head attention-based meta relational learning for few-shot link prediction

where $e(\cdot)$ denotes an embedding lookup table, $k=|\mathcal{S}_r|$ is the shot number, and $i \in \{1, 2, \dots, k\}$.

We concatenate entity embeddings both for the positive fact (h_i, t_i) and the negative one (h_i, t'_i) , generating inputs of the relation-meta encoder in Eq. (2). The symbol \oplus represents the concatenation function.

$$\begin{aligned} x_{p_i}^0 &= x_{h_i} \oplus x_{t_i} \\ x_{n_i}^0 &= x_{h_i} \oplus x_{t'_i} \end{aligned} \quad (2)$$

where x_{h_i} , x_{t_i} , and $x_{t'_i}$ are embeddings of the head entity h_i , the positive tail entity t_i , and the negative tail entity t'_i .

3.2.2 Relation-meta Encoder

The relation-meta encoder consists of two stacked encoding modules and an inference long short-term memory network (LSTM) sublayer. Since the previous implementation is based on simple multi-layer perceptron [55] (MLP), it could lead to vanishing gradient problems when the network goes deeper [68]. In comparison, here we extract fact-specific relation-meta via the multi-head attention-enhanced LSTM, allowing the encoder to capture implicit long-range semantic dependencies [69, 70] between the head entities and tail entities.

As shown in Fig. 4, an encoding module consists of an LSTM sublayer and a multi-head attention layer, learning

a deeper representation of facts for relation encoding. In Eq. (3), both the positive embedding $\mathbf{x}_{p_i}^0$ and the negative one $\mathbf{x}_{n_i}^0$ are fed into an LSTM sublayer that yields an intermediate output representation $\{\mathbf{H}_{p_i}^l, \mathbf{H}_{n_i}^l\} \in \mathbb{R}^{k \times d}$, where k and d denote the shot number of the task \mathcal{T}_{Θ_r} and LSTM hidden size, respectively. l here is the layer number of the encoding module.

$$\begin{aligned}\mathbf{H}_{p_i}^1 &= LSTM(\mathbf{x}_{p_i}^0) \\ \mathbf{H}_{n_i}^1 &= LSTM(\mathbf{x}_{n_i}^0)\end{aligned}\quad (3)$$

Then a multi-head attention sublayer is stacked to the above LSTM sublayer, forcing the encoder to focus on the commonalities of facts. For the multi-head attention mechanism, we assign $\mathbf{Q} = \mathbf{H}_{p_i}^1$, $\mathbf{K} = \mathbf{V} = \mathbf{H}_{n_i}^1$. Hence, the attention mechanism is calculated using Eq. (4).

$$\text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}\quad (4)$$

We use multi-head for capturing the distribution difference between positive facts and negative ones in parallel.

$$\begin{aligned}\text{head}_i &= \text{attention}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \\ \mathbf{H}_{a_i}^1 &= \text{concat}(\text{head}_1, \dots, \text{head}_q) + \mathbf{H}_{p_i}^1\end{aligned}\quad (5)$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d \times \frac{d}{q}}$, $\mathbf{W}_i^K \in \mathbb{R}^{d \times \frac{d}{q}}$, $\mathbf{W}_i^V \in \mathbb{R}^{d \times \frac{d}{q}}$ are trainable parameters, and q is the number of parallel heads. $\mathbf{H}_{a_i}^1$ denotes the intermediate output representation of the first multi-head attention sublayer. Then the output of the first encoding module \mathbf{H}_i^1 is the concatenation of $\mathbf{H}_{p_i}^1$ and $\mathbf{H}_{a_i}^1$.

$$\mathbf{H}_i^1 = \mathbf{H}_{p_i}^1 \oplus \mathbf{H}_{a_i}^1\quad (6)$$

We stack two encoding modules in the relation-meta encoder. Here, $\mathbf{H}_i^2 = \mathbf{H}_{p_i}^2 \oplus \mathbf{H}_{a_i}^2$ denotes the output of the second encoding module. Then, an inference LSTM sublayer that only modifies dropout to be 0 is added to generate the fact-specific relation-meta R_i in Eq. (7)

$$R_i = LSTM(\mathbf{H}_i^2)\quad (7)$$

At last, the relation-meta in Eq. (8) is the average of all the fact-specific relation-meta R_i . k denotes the shot number of the task \mathcal{T}_{Θ_r} .

$$\mathcal{R}_{\mathcal{T}_{\Theta_r}} = \frac{\sum_{i=1}^k R_i}{k}\quad (8)$$

3.2.3 Training

A score function is used to evaluate the relation-meta and the loss function is employed to update the whole model. The score function of the support set is defined in Eq. (9) based on TransE [62]. Inspired by Lin et al. [71], a penalty term is added to ensure the relation-meta $\mathcal{R}_{\mathcal{T}_{\Theta_r}}$ not too far away from the original relation embedding x_{r_i} . λ is the weight parameter for this constraint, and $\|\cdot\|_2^2$ is the squared L_2 norm operation.

$$s_{(h_i, t_i)} = \|x_{h_i} + R_{T_r} - x_{t_i}\|_2^2 + \lambda \|R_{T_r} - x_{r_i}\|_2^2\quad (9)$$

In Eq. (10), margin ranking loss [72] is conducted to compute the loss of the support set \mathcal{S}_r for parameter updating.

$$\mathcal{L}(\mathcal{S}_r) = \sum_{(h_i, t_i) \in \mathcal{S}_r} \left[m + s_{(h_i, t_i)} - s_{(h_i, t'_i)} \right]_+\quad (10)$$

where m represents the margin hyperparameter. $s_{(h_i, t_i)}$ is the score of the positive fact and $s_{(h_i, t'_i)}$ is the score of the corresponding negative fact. $[\cdot]_+$ denotes the positive part of the function.

In Eq. (11), we compute the gradient of parameters, denoted as $\nabla_{\mathcal{R}_{\mathcal{T}_{\Theta_r}}} \mathcal{L}(\mathcal{S}_r)$, in the support set \mathcal{S}_r to update the relation-meta and transfer it to the query set \mathcal{Q}_r to get the updated relation-meta $\mathcal{R}'_{\mathcal{T}_{\Theta_r}}$ and the updated relation embedding \tilde{x}_{r_j} . β here indicates the updating step size.

$$\begin{aligned}\tilde{R}_{T_r} &= R_{T_r} - \beta \nabla_{R_{T_r}} \mathcal{L}(\mathcal{S}_r) \\ \tilde{x}_{r_j} &= x_{r_j} - \beta \nabla_{x_{r_j}} \mathcal{L}(\mathcal{S}_r)\end{aligned}\quad (11)$$

Thus, the scores and loss of query set in Eq. (12) and Eq. (13) follow the same procedure in the support set.

$$s_{(h_j, t_j)} = \|x_{h_j} + \tilde{R}_{T_r} - x_{t_j}\|_2^2 + \lambda \|\tilde{R}_{T_r} - \tilde{x}_{r_j}\|_2^2\quad (12)$$

$$\mathcal{L}(\mathcal{Q}_r) = \sum_{(h_j, t_j) \in \mathcal{Q}_r} \left[m + s_{(h_j, t_j)} - s_{(h_j, t'_j)} \right]_+\quad (13)$$

Our training objective is to minimize the loss of both the support set and the query set. Finally, the updated relation-meta is used to predict tail entities, and recommendations are based on the rankings of the predictions \mathcal{Y} in the test set. From Eq. (14), top- k querying [73] is adopted to return the k samples with the highest scores as the candidate recommendations.

$$P_k(s) \in \{y \in \mathcal{Y}^{(k)} : \forall i \in \{1, \dots, k\}, s_{y_i} \geq s_{[k]}\}\quad (14)$$

where $\mathcal{Y}^{(k)}$ represents the set of k -tuples with k distinct samples of \mathcal{Y} . \mathbf{y} denotes a tuple distinguished from a single prediction $y \in \mathcal{Y}$. s refers to the score computed as Eq. (12), and $s_{[k]}$ is the k th maximum score.

Table 2 Statistics of datasets. The *Pre-Train* setting represents using background to train entity embedding in advance. The *In-Train* setting fits the background graph into training tasks. # Train, # Dev, and

Test count the number of relations in training, validation, and test set. Regardless of the different settings above, # Ent, # R, and # Triples denote the number of entities, relations, and triples

Dataset	Setting	# Train	# Dev	# Test	# Ent	# R	# Triples
NELL-One	Pre-Train	321	5	11	68,545	358	181,109
	In-Train	51	5	11			
Wiki-One	Pre-Train	589	16	34	4,838,244	822	5,859,240
	In-Train	133	16	34			

4 Experiment

4.1 Multi-head Attention-Based Meta Relational Learning

In this section, we conduct experiments¹ on two public data sets to demonstrate the validity of our method. Our meta relational learning model outperforms previous methods, GMatching [54], MetaR [55], and GANA [74], which achieve state-of-the-art results on the few-shot link prediction benchmarks. We further develop an industrial knowledge graph-based platform embedded with our meta relational learning model to support the recommender system in low resource conditions.

4.1.1 Datasets and Evaluation Metrics

4.1.1.1 Datasets In the experiments, two public datasets, namely NELL-One and Wiki-One, are employed to evaluate the performance of our meta relational learning model. These datasets are first constructed by Xiong et al. [54] and then reused by Chen et al. [55]. They are commonly used in few-shot link prediction [54, 55, 74]. The statistics of datasets are shown in Table 2.

Since GMatching considers both learned embeddings and one-hop graph structures, a background graph is constructed with relations out of training/validation/test sets to obtain the pre-trained embeddings and local graph. Except for one-shot task relations, GMatching selects the rest of the relations as background relations that provide crucial background knowledge for training. MetaR further uses the background graph to get two dataset settings. (1) *Pre-Train* setting represents the background used for pretraining. (2) *In-Train* setting denotes that the background graph is fitted into the training tasks with random initialization.

4.1.1.2 Evaluation For evaluation, MRR (mean reciprocal rank) and Hits@N are widely adopted for evaluating few-

shot link prediction algorithms. MRR is an average measure of reciprocal ranks over all the true triples [55, 75, 76]. Hits@N counts the proportion of correct entities ranked in the top N in link prediction [54, 55].

4.1.2 Implementation and Baselines

4.1.2.1 Implementation Our model uses the Adam optimizer [77] for adaptive convergence, and we use the Leaky ReLU function [78] to address the dying ReLU problem. The initial learning rate is 0.001 and gradually drops to 0.0001. The dropout rate is set to be 0.5. We set the margin $m = 1$ and the updating step size $\beta = 5$. The training would be stopped when the performance on Hits@10 drops 30 times.

Parameters of the relation-meta encoder in Sect. 3.2.2 are as follows. Following GMatching, [54] the initial embedding dimension of NELL-One is 100 and Wiki-One is 50. For the first LSTM sublayer, the input dimension of NELL-One is 200 and Wiki-One is 100, with the size of the hidden state being 200 for NELL-One and 100 for Wiki-One. For the multi-head attention sublayers, the input size is 200 for NELL-One and 100 for Wiki-One. The number of parallel attention heads is 5. Due to the concatenation operation, for the inference LSTM sublayer, the input dimension of NELL-One is 400 and Wiki-One is 200, with the output size being 100 for NELL-One and 50 for Wiki-One.

4.1.2.2 Baselines Since GMatching and MetaR are two notable baselines of few-shot link prediction task, we compare our method with the embedding-based methods below, following the implementations of Chen's report [55] for comparison:

- (1) TransE [62]: a classical method that transforms the triples into a low-dimensional vector space.
- (2) TransH [79]: a hyperplane-based translation model.
- (3) RESCAL [80]: a factorization-based method that combines the collective learning of a three-way tensor with the factorization to learn the latent relation space of the triples.

¹ The source code of Sect. 3.2 is available at <https://github.com/RusselMcGrady/Multi-head-attention-based-MetaR>.

Table 3 Results of few-shot link prediction on NELL-One. Underline numbers are the best results of our model. Our model and MetaR test on both the *Pre-Train* setting and the *In-Train* setting, specified in (bracket)

NELL-One	MRR		Hits@10		Hits@5		Hits@1	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
TransE	0.105	0.168	0.226	0.345	0.111	0.186	0.041	0.082
TransH	0.168	0.279	0.233	0.434	0.160	0.317	0.127	0.162
RESCAL	0.140	–	0.229	–	0.186	–	0.089	–
DisMult	0.165	0.214	0.285	0.319	0.174	0.246	0.106	0.140
ComplEx	0.179	0.239	0.299	0.364	0.212	0.2153	0.112	0.176
GMatching_RESCAL	0.188	–	0.305	–	0.243	–	0.133	–
GMatching_TransE	0.171	–	0.255	–	0.210	–	0.122	–
GMatching_DisMult	0.171	–	0.301	–	0.221	–	0.114	–
GMatching_ComplEx	0.185	0.201	0.313	0.311	0.260	0.264	0.119	0.143
GMatching_Random	0.151	–	0.252	–	0.186	–	0.103	–
CogKR	0.256	–	0.353	–	0.314	–	0.205	–
GANa	0.307	0.344	0.483	0.517	0.409	0.437	0.211	0.246
MetaR (<i>Pre-Train</i>)	0.164	0.209	0.331	0.355	0.238	0.280	0.093	0.141
MetaR (<i>In-Train</i>)	0.250	0.261	0.401	0.437	0.336	0.350	0.170	0.168
Our model (<i>Pre-Train</i>)	0.236	0.243	0.359	0.377	0.299	0.315	0.168	0.171
Our model (<i>In-Train</i>)	<u>0.325</u>	<u>0.337</u>	<u>0.487</u>	<u>0.521</u>	<u>0.418</u>	<u>0.447</u>	<u>0.237</u>	<u>0.234</u>

Table 4 Results of few-shot link prediction on Wiki-One. Underline numbers are the best results of our model. Both the *Pre-Train* setting and the *In-Train* setting results are also specified in (bracket)

Wiki-One	MRR		Hits@10		Hits@5		Hits@1	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
TransE	0.036	0.052	0.059	0.090	0.024	0.057	0.011	0.042
TransH	0.068	0.095	0.333	0.177	0.060	0.092	0.027	0.047
RESCAL	0.072	–	0.082	–	0.062	–	0.051	–
DisMult	0.046	0.077	0.087	0.134	0.034	0.078	0.014	0.035
ComplEx	0.055	0.070	0.100	0.124	0.044	0.063	0.021	0.030
GMatching_RESCAL	0.139	–	0.305	–	0.228	–	0.061	–
GMatching_TransE	0.219	–	0.328	–	0.269	–	0.163	–
GMatching_DisMult	0.222	–	0.340	–	0.271	–	0.164	–
GMatching_ComplEx	0.200	–	0.336	–	0.272	–	0.120	–
GMatching_Random	0.198	–	0.299	–	0.260	–	0.133	–
CogKR	0.288	–	0.366	–	0.334	–	0.249	–
GANa	0.301	0.351	0.416	0.446	0.350	0.407	0.231	0.299
MetaR (<i>Pre-Train</i>)	0.314	0.323	0.404	0.418	0.375	0.385	0.266	0.270
MetaR (<i>In-Train</i>)	0.193	0.221	0.280	0.302	0.233	0.264	0.155	0.178
Our model (<i>Pre-Train</i>)	0.333	0.354	0.445	0.470	0.397	0.432	0.282	0.296
Our model (<i>In-Train</i>)	0.215	0.247	0.310	0.352	0.262	0.307	0.172	0.193

(4) DisMult [81]: setting the matrix to diagonal matrices and constructing the object function to simplify the computation of RESCAL.

(5) ComplEx [82]: a latent factorization method that uses the composition of complex embeddings to handle both the symmetric and antisymmetric relations.

(6) GMatching [54]: a one-shot relational learning model that is trained to match local graph patterns. Notably, GMatching_RESCAL, GMatching_TransE, GMatching_DisMult, GMatching_ComplEx, and GMatching_Random in Table 3 and Table 4 represent different selections of the matching processor in the GMatching model.

- (7) CogKR [83]: a cognitive graph based on coordinating retrieval and reasoning.
- (8) GANA [74]: a global–local framework.

4.1.3 Results

We use two initialization strategies, *Pre-Train* and *In-Train*, for 1-shot and 5-shot tasks on NELL-One and Wiki-One following MetaR [55]. The results of the test set are shown in Table 3 and Table 4. Results of the baselines with different initialization of KGE are copied from the original paper in this paper.

In Table 3 and Table 4, our model performs better with all evaluation metrics compared with MetaR on both datasets. (1) With the *Pre-Train* setting, our model increases 1-shot link prediction on NELL-One by 43.9, 8.5, 25.6, and 80.6% on MRR, Hits@10, Hits@5, and Hits@1, with an average improvement of 39.1%. The improvements of 1-shot link prediction on Wiki-One are 6.1, 10.1, 5.9, and 6.0%, with an average improvement of 7.0%. Besides, our model increases 5-shot link prediction on NELL-One by 19.1, 6, 12.5, and 21.3% on MRR, Hits@10, Hits@5, and Hits@1, with an average improvement of 14.9%. The improvements in 5-shot link prediction on Wiki-One are 9.6, 12.4, 12.2, and 9.6% on Wiki-One, with an average improvement of 11.0%. (2) With the *In-Train* setting, our model increases 1-shot link prediction on NELL-One by 30%, 21.4, 24.4, and 39.4%, with an average improvement of 28.8%. The improvements of 1-shot link prediction on Wiki-One are 11.4, 10.7, 12.4, and 11%, with an average improvement of 11.4%. Besides, our model increases 5-shot link prediction on NELL-One by 29.1, 19.2, 27.7, and 39.3%, with an average improvement of 28.8%. The improvements in 5-shot link prediction on Wiki-One are 11.7, 16.6, 16.3, and 8.4%, with an average improvement of 13.3%.

Moreover, we conclude that the performance of few-shot link prediction is affected by the quantity and quality of training data. On the one hand, the *In-Train* dataset setting is better than the *Pre-Train* dataset setting on large and sparse datasets because the relations are too sparse to learn the relation-meta. Controversially, the *Pre-Train* setting provides a few supervised signals for the sparse data for fast convergence, and we conduct schema-based reasoning based on the same intuition. On the other hand, the *In-Train* dataset setting performs better on the small-scale dataset because our meta relational learning model focuses on learning task-specific relation-meta. Bi-LSTM layers are also employed in the relation-meta encoder, while this implementation

performs even worse because the reverse relations of facts should be mapped as a different relation-meta.

4.2 Case Study

In this section, a graph-based platform is developed to demonstrate the feasibility of our method. The proposed recommendation algorithm is embedded in the platform with the graph searching engine to support product development and service innovation for Zhejiang Yueli Electrical CO.LTD (YL) [84]. YL is one of the top three home appliance manufacturing bases in China, with annual sales exceeding 1.7 billion yuan. Their three factories have more than 50 assembling lines, 7 automatic painting lines, 286 patents, and 258 injection molding machines.

4.2.1 Evaluation

To evaluate the impact of our recommendation algorithm on the industrial knowledge graph of YL, a business metric, net promoter score (NPS) [85, 86], is introduced in this section. NPS is a measure of customer loyalty that evaluates users' preferences for the recommended results. However, NPS based on customer surveys may lead to subjectivity and randomness of results. Assuming the threshold between promoters and passives is m , and the threshold between passives and detractors is n . The number of candidate resources is k for each query entity. Thus, we define the promoters, the passives, and the detractors as follows.

- (1) Promoters are correct predictions ranked in the top m .
- (2) Passives are correct predictions that are ranked from m th to n th.
- (3) Detractors are correct predictions that are ranked from n th to k th.

Based on our definition, NPS is calculated by subtracting the percent of detractors from the percent of promoters, as shown in Eq. (15).

$$\text{NPS} = \frac{1}{|k|} \sum_{i=1}^{|k|} \mathbb{I}(\text{rank}_i \leq m) - \frac{1}{|k|} \sum_{i=1}^{|k|} \mathbb{I}(n \leq \text{rank}_i \leq k) \quad (15)$$

where rank_i is the ranking of the i th prediction, and $\mathbb{I}()$ is an indicator function counting rankings that meet the inequality constraints.

Referenced to the construction of NELL-One, we select 50/5/10 task relations that are collected from the industrial knowledge graph constructed for YL for training/validation/testing, considering sufficient triples for evaluation [54]. Here

Table 5 NPS results with different ranking thresholds

Threshold settings		MRR		Hits@10		Hits@5		Hits@1		NPS	
m	n	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
1	5	0.321	0.329	0.484	0.500	0.411	0.431	0.234	0.230	0.161	0.161
1	8	0.324	0.331	0.492	0.516	0.423	0.442	0.227	0.226	0.201	0.201
5	8	0.323	0.331	0.481	0.509	0.413	0.437	0.234	0.228	0.395	0.420

Table 6 Comparative Experiment

Method	MRR		Hits@10		Hits@5		Hits@1		NPS	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
CKFG	0.026	0.029	0.055	0.062	0.024	0.029	0.004	0.009	0.024	0.031
CKE	0.033	0.043	0.061	0.075	0.044	0.058	0.006	0.014	0.038	0.048
NFM	0.037	0.053	0.069	0.087	0.061	0.068	0.009	0.034	0.057	0.053
KGAT	0.040	0.074	0.087	0.092	0.073	0.081	0.013	0.058	0.069	0.078
MetapathRS	0.049	0.077	0.090	0.101	0.074	0.088	0.030	0.065	0.070	0.083
Our model	0.323	0.331	0.481	0.509	0.413	0.437	0.234	0.228	0.395	0.420

we use random initialization. The results are shown in Table 5, and the performance of NPS is related to the ranking threshold settings of m and n .

Table 5 shows that NPS keeps going up with higher threshold settings of m and n , which denotes users would require more recommended resources. If the manufacturing process requires more accurate recommendations, then a strict threshold should be considered. Considering our recommendation engine is based on the few-shot link prediction, the threshold settings of m and n in Table 5 refer to the widely used evaluation metrics Hits@N [55], including 1 and 5, 1 and 8, and 5 and 8, with 10 recommendations for each query. It is based on the intuition that redundant and mismatched recommendations are worthless. Additionally, due to the random initialization adopted for testing here, slight fluctuations of MRR and Hits@N demonstrate that our meta relational learning model could predict any newly added relations without fine-tuning.

4.2.2 Comparative Experiment

To compare with prior recommendation approaches under low-resource conditions, the following algorithms are employed as baselines using the data of YL in Sect. 4.2.1:

- (1) CKFG [19]: an explainable recommendation approach using knowledge-based embeddings.
- (2) CKE [8]: a collaborative knowledge base embedding approach that uses the heterogeneous information in a

knowledge base to improve the quality of recommender systems.

(3) NFM [87]: a novel neural factorization machine for predicting recommendations under sparse settings.

(4) KGAT [88]: a framework that investigates the utility of the knowledge graph to provide explainable recommendations.

(5) MetapathRS [89]: a unified recommendation method with embedding-based learning and graph-based learning.

Here, we reimplement all the approaches by setting the threshold settings of m and n to be 5 and 8. We randomly initialize all the embeddings and the learning rate is 0.001. Evaluation metrics introduced in Sect. 4.1.1 are also adopted for evaluation. Experimental results are shown in Table 6.

From Table 6, our model outperforms previous approaches on all the evaluation metrics. Compared with our model, there are significant gaps in long-tail recommendations for prior knowledge graph-based recommendation approaches like KGAT. We postulate that the performance divergence generally falls into two factors. First, long-tail recommendations that are formulated as a few-shot learning problem of learning-to-recommend long-tail resources with few interactions pose extreme difficulties to the previously reported recommendation approaches. Second, the industrial knowledge graph of YL includes resources with various labels and diverse relations, but the methods like KGAT and MetapathRS that are developed to address user-item graphs



Fig. 5 Graph-based platform: home page of our recommender system

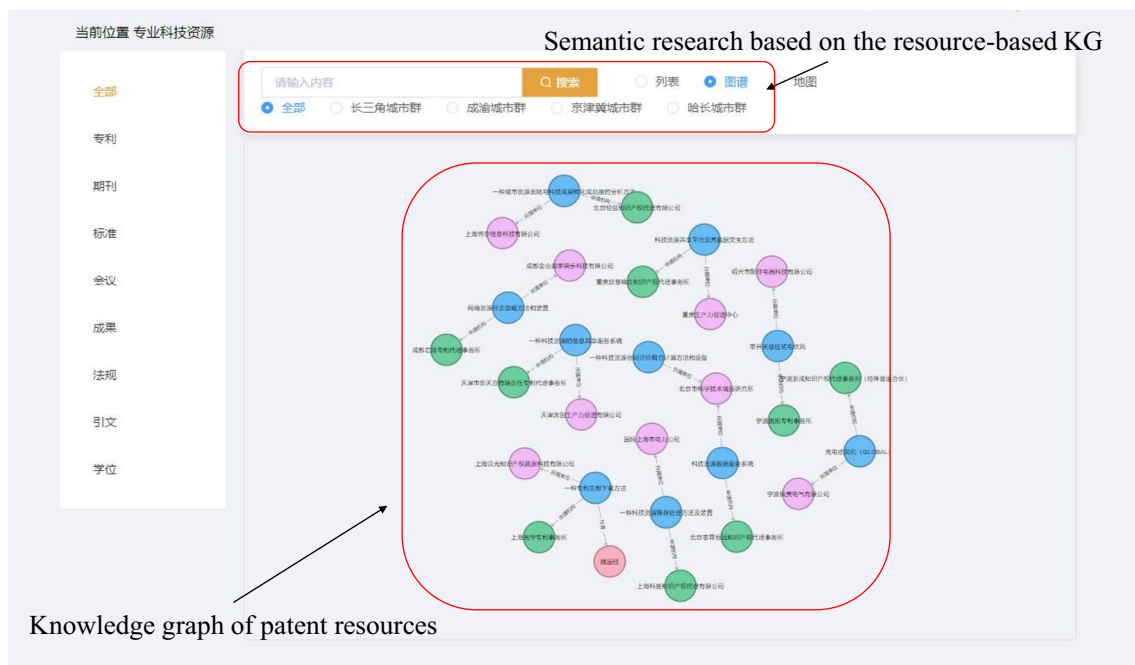


Fig. 6 Graph-based platform: knowledge graph search

Table 7 Outsourcing Procurement of YL

Procedure	Sub-process	Department	Resource
1	Release requirements	Planning department	Purchase requisition Specification Drawing Standard, Patent
2	Receive requisition information	Purchasing department	Purchase requisition
3	Identify suppliers		Approved supplier list
4	Create a purchase order		Purchase order letter
5	Receive the purchase order	Supplier	Purchase order letter Specification
6	Proofing		Drawing Standard, Patent
7	Sample Information	Purchasing department	Sample confirmation letter Product sample
8	Sample confirmation	Research and develop- ment department	Sample confirmation letter
9	Quality inspection		Sample confirmation letter Standard
10	Issue test reports, sample confirma- tion, and the sealed sample		Sample confirmation letter Test report
11	Store and management resources	Purchasing department	Sample confirmation letter

**Fig. 7** Outsourcing procurement of YL: Process engine

cannot be directly adapted to this scenario. Comparatively, based on these observations, we propose handling such problems by learning the robust relation-meta between potentially associated resources.

4.2.3 Applications

The platform integrates resources throughout the product life cycle based on the industrial knowledge graph. Additionally, newly arising resources produced by the business process system are added to the graph, forming a closed loop. Once YL restarts a business process, candidate resources are

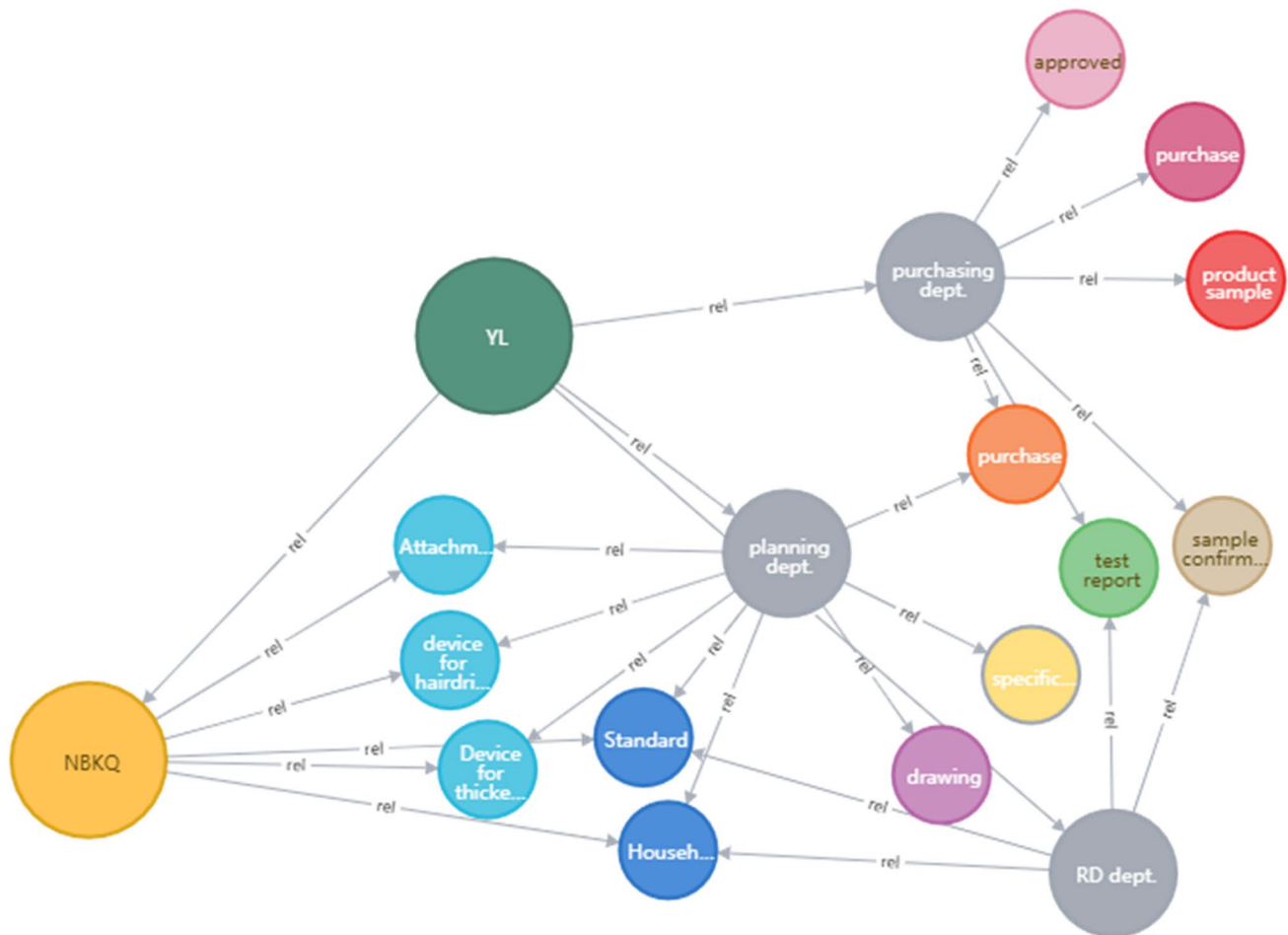


Fig. 8 Outsourcing procurement of YL: Knowledge graph visualization

recommended to corresponding nodes according to the links in the knowledge graph.

Figure 5 and 6 are applications of the graph-based platform, including the resource map, graph search engine, and recommendation engine based on the proposed approach in Sect. 3.

We further analyze an outsourcing procurement process of the YL enterprise for a case study. It is a collaborative step in the product design process. Resources involved in the 11 sub-processes are shown in Table 7. Knowledge resources are standards and patents. Business data resources include purchase requisitions, purchase order letters, test reports, sample confirmation letters, and specifications. Product resources are product samples and drawings.

The core enterprise YL can manage the entire outsourcing process by the process engine in the platform, as shown in Fig. 7. Meanwhile, the industrial resources involved in the process are continuously added to the graph-based platform. The core enterprise releases resources to cooperative suppliers that upload resources back to the core enterprise for

industrial process control. Figure 8 shows the knowledge graph that integrates all the resources involved in the outsourcing process. Yet, such data are still static and sparse. In this sense, our recommendation algorithm would recommend resources under low-resource conditions.

From Fig. 9, black links annotated as *rel*, specify the correlations among resources based on the process engine. The platform further links the resources based on schema-based reasoning (blue links in Fig. 9), forming the training set for few-shot link prediction.

Specifically, the supplier BJHC (denoted as BJHC in Fig. 9) could be recommended to the core enterprise because of the link annotated as *rel* based on the historical rule. Besides, according to the performance rule, the platform links the supplier ZJLB (denoted as ZJLB in Fig. 9) to the core enterprise YL (denoted as YL in Fig. 9) because both the supplier ZJLB and supplier NBKQ (denoted as NBKQ in Fig. 9) are satisfied with the requirements of the specification. Third, the patents and standards (nodes in black ovals in Fig. 9) are linked to supplier NBJH (denoted as NBJH in

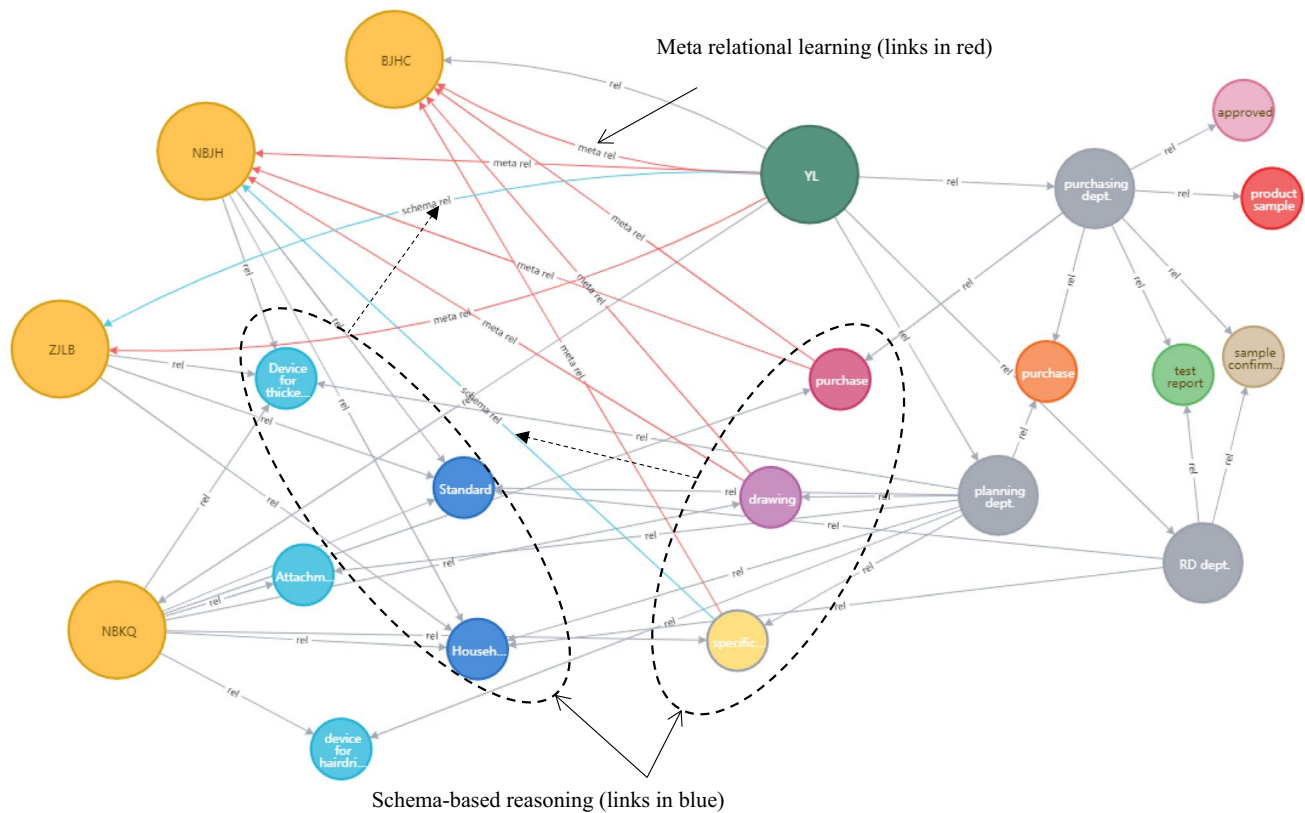


Fig. 9 An example of recommending industrial resources for the outsourcing procurement of YL

Fig. 9) based on the community rule. Lastly, with the path rule, several resources like drawings can be recommended to the core enterprise YL according to the two-hop paths in the graph.

Also, schema-based reasoning generates relations and triples for few-shot recommendations. The platform conducts multi-head attention-based meta relational learning to link implicit resources for the correlative recommendation. As shown in Fig. 9, the model learns the relation-meta (red links in Fig. 9) to correlate long-tail resources. For example, the supplier NBJH is linked to the core enterprise YL, because both suppliers (ZJLB and NBJH) are satisfied with the knowledge resource requirements and the schema-based relation in the graph serves as the heuristic rule for meta training. Moreover, the meta relational learning model learns the relation-meta for product specification, drawing, purchase order, and suppliers shown in the graph. These industrial resources are recommended to the suppliers (NBJH and BJHC) because of their strong relevance in the outsourcing process. All the predicted links practically improve the YL's outsourcing process and further complete the industrial knowledge graph.

5 Analysis and Discussion

To the best of our knowledge, we are among the first to construct an industrial knowledge graph and use the few-shot relational learning to address the cold-start problem in the recommender system. We develop a meta relational learning model for recommender systems based on the intuitive that the close correlations among industrial resources are conducive to the conditions where massive labeled data are unavailable. The case study demonstrates that the industrial knowledge graph contributes to resource ordering, and our meta relational learning model with schema-based reasoning helps to solve the cold-start problem in recommender systems.

Table 8 Ablation Study on NELL-One with Hits@10 Metric

Ablation Configuration	1-shot (Pre-Train/In-Train)	5-shot (Pre-Train/In-Train)
standard	0.359/0.487	0.377/0.521
-mh	0.253/0.368	0.296/0.463
-g	0.245/0.353	0.281/0.447
-g -r	0.226/0.226	0.345/0.345

Table 9 Abbreviations

Abbreviation	Meaning
Long-tail resource	Resources are less commonly referenced within a knowledge graph
Cold-start problem	Engines cannot make reliable recommendations due to an initial lack of ratings
KGE	knowledge graph embedding
LSTM	Long short-term memory
MRR	Mean reciprocal rank
Pre-Train	Pre-Train dataset setting
In-Train	In-Train dataset setting
YL	Zhejiang Yueli Electrical CO.LTD
MetaR	Meta Relational Learning for Few-Shot Link Prediction in Knowledge Graphs 55.
GMatching	One-Shot Relational Learning for Knowledge Graphs 54.
ZJLB	Supplier ZJLB
NBKQ	Supplier NBKQ
BJHC	Supplier BJHC

Inevitably, our method also suffers from multiple limitations. Specifically, the construction of the resource graph is hugely influenced by the specialists using top-down methods. Contrarily, the performance of bottom-up methods in most industrial scenarios, such as unsupervised learning, is unsatisfactory. Moreover, the meta relational learning model greatly affects the performance of the recommender systems that are troubled with the cold-start problem. Thus, we discuss the components and modules of our model through the ablation study.

The ablation study on NELL-One selects metric Hit@10 with four settings for evaluation. The results are shown in Table 8 where the marks (*Pre-Train/In-Train*) represent two different dataset settings in experiments. First, we remove the multi-head attention sublayers, denoted as *-mh*. Second, we remove the training module, denoted as *-g* following MetaR for convenience. Third, we remove the entire relation-meta encoder and the penalty term in the training module, which makes the model rebase to a simple TransE model, denoted as *-g -r*. is the result of TransE is copied from GANA [74], which is neither *Pre-Train* nor *In-Train*. The last one is our complete model denoted as standard.

First, removing the multi-head attention layers for *Pre-Train* and *In-Train settings*, our model decreases by 29.5 and 24.4% for the 1-shot task, and 21.5 and 11.1% for the 5-shot task. Second, by removing the translating embedding module for *Pre-Train* and *In-Train settings*, our model decreases by 31.8 and 27.5% for the 1-shot task, and 25.5 and 14.2% for the 5-shot task respectively. At last, removing the entire relation-meta encoder *Pre-Train* and *In-Train settings*, our model decreases by 37 and 48.7% for the 1-shot task, and 8.5 and 33.8% for the 5-shot task.

The results demonstrate that all the components of our model contribute dramatically, and the relation-meta encoder contributes more than translating module. Moreover, more shot samples contribute to few-shot learning algorithms because such training samples enhance the robustness of models. This also illustrates that the few-shot link prediction is a practical solution to the cold-start problem of recommender systems.

However, all evaluation metrics of our model still have room for improvement, especially compared with deep learning models that have sufficient data for training. Thus, refinement and optimization of meta relational learning models are expected in the future.

6 Conclusion and Future Work

Here we propose an industrial knowledge graph to recommend resources in low-resource conditions for industrial collaboration. We construct the industrial knowledge graph with the predefined schema to enhance resource sharing and improve resource reuse. The nodes in the graph represent different resources, and the links in the graph are correlations among resources. Then the schema-based reasoning links resources to heuristically construct the training data for few-shot learning. The multi-head attention-based meta relational learning further learns the relation-meta that supports the correlative recommendation within the manufacturing value chain.

In the future, we will employ our recommendation algorithm in other domains based on the domain knowledge graph. In addition, reinforcement learning and generative

Table 10 Symbol Table

Symbol	Meaning
(h_i, t_i)	Entity pair, including the head entity and the tail entity
$\mathcal{T}_r = \{\mathcal{S}_r, \mathcal{Q}_r\}$	A link prediction task with data in the support set and the query set
k	Shot number in a task
$e()$	Embedding lookup table
$\mathbf{x}_{p_i}^0, \mathbf{x}_{n_i}^0$	Positive embedding and negative embedding in the representation layer
$\{\mathbf{H}_{p_i}^l, \mathbf{H}_{n_i}^l\} \in \mathbb{R}^{k \times d}$	Intermediate output representation of LSTM sublayer in the relation-meta encoder. k is the shot number, and d is the hidden size of LSTM. l is the layer number of the encoding module
$\mathbf{Q} = \mathbf{H}_{p_i}^1, \mathbf{K} = \mathbf{V} = \mathbf{H}_{n_i}^1$	Key-value pair of the multi-head attention sublayer
$\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$	Trainable parameters
q	The number of parallel heads
$\mathbf{H}_{a_i}^l$	Intermediate output representation of the multi-head attention sublayer
\mathbf{H}_i^l	The output of the l th encoding module
R_i	Fact-specific relation-meta
$\mathcal{R}_{\mathcal{T}\Theta_r}$	Relation-meta
λ	Weight parameter
β	Updating step size
m	Margin hyperparameter

adversarial learning would be used to improve the performance of few-shot relational learning. Finally, multi-modal knowledge graphs with industrial applications are to be developed to enhance industrial cooperation.

7 List of Abbreviations and symbols

A list of abbreviations in this paper is provided in Table 9.

A symbol table for Sect. 3.2 is provided in Table 10.

Acknowledgements Not applicable.

Author contributions YL and FG conceived of the presented idea. YL developed the theory and carried out the experiments. XG and YW verified the analytical methods. FG and JG supervised the project. All authors have read and approved the final manuscript.

Funding This study is financially supported by the National Natural Science Foundation of China (No. 71901194 & No. 51775493), Natural Science Foundation of Zhejiang Province (No. LZ15E050003), Ministry of Industry and Information Technology of the People's Republic China's 2016 Green Manufacturing System Integration Project "Construction of Green Supply Chain System for Home Appliance Industry", Eyas Program Incubation Project of Zhejiang Provincial Administration for Market Regulation (CY2022221), and Ningbo Science and Technology Innovation 2025 Major Project (No. 2018B10069).

Availability of Data and Materials The datasets generated during and/or analyzed during the current study are available in the public repository, <https://github.com/AnselCmy/MetaR> [55].

Declarations

Competing Interests The authors declare that they have no competing interests.

Ethics Approval Not applicable.

Consent for Publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Zheng, P., wang H, Sang Z, et al.: Smart manufacturing systems for Industry 4.0: conceptual framework, scenarios, and future perspectives. *Front Mech Eng* **13**(2), 137–150 (2018)
2. Kulkarni VN, Gaitonde VN, Kotturshettar BB. Product Lifecycle Management (PLM): A Key Enabler in Implementation of Industry 4.0. In: Hussain CM, Di Sia P, eds. *Handbook of smart materials, technologies, and devices: applications of industry 4.0*. Cham: Springer International Publishing :1–32 (2020).
3. Bader SR, Grangel-Gonzalez I, Nanjappa P, Vidal M-E, Maleshkova M. 2020 A Knowledge graph for Industry 4.0. paper presented at: The semantic Web; //,; Cham (2020)

4. Steele, J., Son, Y.-J., Wysk, R.A.: Resource modeling for the integration of the manufacturing enterprise. *J. Manuf. Syst.* **19**(6), 407–427 (2001)
5. Kosmopoulos, D., Styliaras, G.: A survey on developing personalized content services in museums. *Pervasive Mob. Comput.* **47**, 54–77 (2018)
6. Sun R, Cao X, Zhao Y, et al. Multi-modal knowledge graphs for recommender systems. proceedings of the 29th acm international conference on information & knowledge management; Virtual Event, Ireland (2020)
7. Wang H, Zhao M, Xie X, Li W, Guo M. Knowledge graph convolutional networks for recommender systems. The World Wide Web Conference (2019)
8. Zhang F, Yuan NJ, Lian D, Xie X, Ma W-Y. Collaborative Knowledge Base Embedding for Recommender Systems. Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining. San Francisco, California, USA (2016)
9. Lin, X., Zhang, M., Liu, Y., Ma, S.: Enhancing personalized recommendation by implicit preference communities modeling. *ACM Trans Inf Syst.* **37**(4), 1–2 (2019)
10. Zhang, W., Zhang, X., Wang, H., Chen, D.: A deep variational matrix factorization method for recommendation on large scale sparse dataset. *Neurocomputing* **334**, 206–218 (2019)
11. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
12. Dong, S., Wang, P., Abbas, K.: A survey on deep learning and its applications. *Computer Sci Rev* **40**, 100379 (2021)
13. Guo, Q., Zhuang, F., Qin, C., et al.: A Survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge Data Eng* (2020). <https://doi.org/10.1109/TKDE.2020.3028705>
14. Rendle, S.: Factorization machines with libFM. *ACM Trans Intell Syst Technol* **3**(3), 1–22 (2012)
15. Sohail, S.S., Siddiqui, J., Ali, R.: Feature-based opinion mining approach (FOMA) for improved book recommendation. *Arab. J. Sci. Eng.* **43**(12), 8029–8048 (2018)
16. Uyangoda L, Ahangama S, Ranasinghe T. User profile feature-based approach to address the cold start problem in collaborative filtering for personalized movie recommendation. paper presented at: 2018 thirteenth international conference on digital information management (ICDIM). 24–26 (2018)
17. Dai, W., Wang, Y., Ma, J., Jin, Q.: BTR: a feature-based bayesian task recommendation scheme for crowdsourcing system. *IEEE Transactions on Computational Social Systems.* **7**(3), 780–789 (2020)
18. Yang, N., Ma, Y., Chen, L., Yu, P.S.: A meta-feature based unified framework for both cold-start and warm-start explainable recommendations. *World Wide Web.* **23**(1), 241–265 (2020)
19. Ai, Q., Azizi, V., Chen, X., Zhang, Y.: Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms.* **11**, 137 (2018)
20. Wang C, Pan S, Hu R, Long G, Jiang J, Zhang C. Attributed graph clustering: a deep attentional embedding approach. *ArXiv.* (2019) abs/1906.06532
21. Zhisheng, Y., Jinyong, C.: Recommendation algorithm based on knowledge graph to propagate user preference. *Int J Computational Intelligence Systems.* **14**(1), 1564–1576 (2021)
22. Li J, Xu Z, Tang Y, Zhao B, Tian H. Deep hybrid knowledge graph embedding for top-n recommendation. paper presented at: web information systems and applications, Cham (2020)
23. Yu X, Ren X, Sun Y, et al. Personalized entity recommendation: a heterogeneous information network approach. Proceedings of the 7th ACM international conference on Web search and data mining. New York, New York, USA (2014)
24. Zhao H, Yao Q, Li J, Song Y, Lee DL. Meta-graph based recommendation fusion over heterogeneous information networks. proceedings of the 23rd acm sigkdd International conference on knowledge discovery and data mining: Halifax, NS, Canada (2017)
25. Wang H, Zhang F, Xie X, Guo M. DKN: Deep knowledge-aware network for news recommendation. proceedings of the 2018 world wide web conference Lyon, France (2018)
26. Wang H, Zhang F, Wang J, et al. RippleNet: propagating user preferences on the knowledge graph for recommender systems. proceedings of the 27th acm international conference on information and knowledge management; Torino, Italy (2018)
27. Wang H, Zhang F, Wang J, et al. RippleNet: propagating user preferences on the knowledge graph for recommender systems. proceedings of the 27th ACM international conference on information and knowledge management; Torino, Italy (2018)
28. Xu B, Xu Y, Liang J, et al. CN-DBpedia: a never-ending chinese knowledge extraction system (2017)
29. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. *Commun ACM.* **57**, 78–85 (2014)
30. Wang, T., Gu, H., Wu, Z., Gao, J.: Multi-source knowledge integration based on machine learning algorithms for domain ontology. *Neural Comput. Appl.* **32**(1), 235–245 (2020)
31. Gan J, Yang J, Jiao D, Li S, Chen G. Entity and relation extraction with dynamic memory cells. paper presented at: 2018 international conference on network infrastructure and digital content (IC-NIDC); 22–24 (2018)
32. Xiao J, Zhou Z. Chapter-level entity relationship extraction method based on joint learning. paper presented at: 2020 12th international conference on intelligent human-machine systems and cybernetics (IHMSC); 22–23 (2020)
33. Zhen L, Qing C, Youmin Z, Sen B. Research on entity semantic relation extraction in fusion domain. paper presented at: 2018 2nd international conference on data science and business analytics (ICDSBA); 21–23 (2018)
34. Lu H, Hong Z, Shi M. 2017 Analysis of film data based on Neo4j. Paper presented at: 2017 IEEE/ACIS 16th International conference on computer and information science (ICIS); 24–26 (2017)
35. Drakopoulos, G., Gourgarris, P., Kanavos, A.: Graph communities in Neo4j. *Evol. Syst.* **11**(3), 397–407 (2020)
36. Zou Y, Liu Y. The Implementation knowledge graph of air crash data based on Neo4j*. Paper presented at: 2020 IEEE 4th Information technology, networking, electronic and automation control conference (ITNEC); 12–14 (2020)
37. Liu D, Li MM. A performance optimization scheme for migrating hive data to Neo4j database. Paper presented at: 2018 International symposium on computer, consumer and control (IS3C); 6–8 (2018)
38. Li, X., Lyu, M., Wang, Z., Chen, C.-H., Zheng, P.: Exploiting knowledge graphs in industrial products and services: a survey of key aspects, challenges, and future perspectives. *Comput. Ind.* **129**, 103449 (2021)
39. Wang Z, Li L, Zeng DD, Chen Y. Attention-based Multi-hop Reasoning for Knowledge Graph. Paper presented at: 2018 IEEE International conference on intelligence and security informatics (ISI); 9–11 Nov. (2018)
40. Wu Y, Jiang A, Tang Y, Kwan HK. GRNet: Deep convolutional neural networks based on graph reasoning for semantic segmentation. Paper presented at: 2020 IEEE International conference on visual communications and image processing (VCIP); 1–4 Dec. (2020)
41. Nie K, Zeng K, Meng Q. Knowledge reasoning method for military decision support knowledge graph mixing rule and graph neural networks learning together. Paper presented at: 2020 Chinese Automation Congress (CAC); 6–8 (2020)
42. Ai-Xia C, Ning L, Li Z, Hong-Tao Z, Guo-Fang Z. An inexact reasoning algorithm based on interaction with fuzzy rule matrix

- transformation. Paper presented at: 2009 International conference on machine learning and cybernetics; 12–15 (2009)
43. Gu J-y. A fuzzy reasoning method based on weight of mixed matrix. Paper presented at: 2010 International conference on artificial intelligence and education (ICAIE); 29–30 (2010)
 44. Ji G, Liu K, He S, Zhao J. Knowledge graph completion with adaptive sparse transfer matrix. Proceedings of the thirtieth AAAI conference on artificial intelligence: Phoenix, Arizona (2016)
 45. Ji G, He S, Xu L, Liu K, Zhao J. Knowledge graph embedding via dynamic mapping matrix. Beijing, China (2015)
 46. Bao J, Duan N, Yan Z, Zhou M, Zhao T. Constraint-Based Question Answering with Knowledge Graph Osaka, Japan (2016)
 47. Dou, J., Qin, J., Jin, Z., Li, Z.: Knowledge graph based on domain ontology and natural language processing technology for Chinese intangible cultural heritage. *J. Vis. Lang. Comput.* **48**, 19–28 (2018)
 48. Shi H-B, Huang D, Wang L, et al. An information integration approach to spacecraft fault diagnosis. *Enterprise information systems*. 1–34 (2019)
 49. Arista R, Mas F, Vallengano C. Initial Approach to an Industrial Resources Ontology in Aerospace Assembly Lines. Paper presented at: Product Lifecycle Management Enabling Smart X, Cham (2019)
 50. Noy N, Gao Y, Jain A, Narayanan A, Patterson A, Taylor J. Industry-scale Knowledge graphs: Lessons and Challenges: Five diverse technology companies show how it's done. *Queue*. 17(2):Pages 20 (2019)
 51. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **29**(12), 2724–2743 (2017)
 52. Zhang N, Deng S, Sun Z, Chen J, Zhang W, Chen H. Relation adversarial network for low resource knowledge graph completion. Proceedings of the web conference. Taipei, Taiwan (2019)
 53. Kampffmeyer M, Chen Y, Liang X, Wang H, Zhang Y, Xing EP. Rethinking knowledge graph propagation for zero-shot learning. paper presented at: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR); 15–20 (2019)
 54. Xiong W, Yu M, Chang S, Guo X, Wang WY. One-shot relational learning for knowledge graphs. *ArXiv*. (2018) abs/1808.09040.
 55. Chen M, Zhang W, Zhang W, Chen Q, Chen H. Meta Relational Learning for Few-Shot Link Prediction in knowledge graphs. *arXiv: Computation and Language*. (2019).
 56. Li C, Li W-q, Li Y, Na H-z, Shi Q. Research and Application of Knowledge Resources Network for Product Innovation. *The Scientific World Journal*. 2015:495309 (2015)
 57. Concilio, G., De Götzen, A., Molinari, F., et al.: Innovation and Design. In: Concilio, G., Tosoni, I. (eds.) *Innovation capacity and the city: the enabling role of design*, pp. 61–83. Springer International Publishing, Cham (2019)
 58. Qiao, B., Fang, K., Chen, Y., Zhu, X.: Building thesaurus-based knowledge graph based on schema layer. *Clust. Comput.* **20**(1), 81–91 (2017)
 59. Manning CD, Surdeanu M, Bauer J, Finkel JR, Bethard S, McClosky D. The Stanford CoreNLP natural language processing toolkit. Paper presented at: ACL2014
 60. Yun H, He Y, Lin L, Pan Z, Zhang X. Construction Research and Application of Poverty Alleviation Knowledge Graph. Paper presented at: Web Information Systems and Applications. Cham (2019)
 61. Baskar, S., Selvaraj, R., Kuthadi, V.M., Shakeel, P.M.: Attribute-based data fusion for designing a rational trust model for improving the service reliability of internet of things assisted applications in smart cities. *Soft. Comput.* **25**(18), 12275–12289 (2021)
 62. Bordes A, Usunier N, Garcia-Durán A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. Proceedings of the 26th International conference on neural information processing systems - volume 2. Lake Tahoe, Nevada (2013)
 63. Chiang W-L, Liu X, Si S, Li Y, Bengio S, Hsieh C-J. ClusterGCN: An efficient algorithm for training deep and large graph convolutional networks. proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining (2019)
 64. Lv X, Gu Y, Han X, Hou L, Li J, Liu Z. Adapting meta knowledge graph information for multi-hop reasoning over few-shot relations. Hong Kong, China (2019)
 65. Seel, N.M.: Schema-Based Reasoning. In: Seel, N.M. (ed.) *Encyclopedia of the sciences of learning*, pp. 2951–2953. Springer, US, Boston, MA (2012)
 66. Sankar A, Wang J, Krishnan A, Sundaram H. ProtoCF: Prototypical collaborative filtering for few-shot recommendation. In: fifteenth ACM conference on recommender systems. Association for computing Machinery. 166–175 (2021)
 67. Hao B, Zhang J, Li C, Chen H. Few-Shot Representation Learning for Cold-Start Users and Items. Paper presented at: Web and Big Data. Cham (2020)
 68. Kolbusz J, Rozycki P, Wilamowski B. The Study of architecture MLP with linear neurons in order to eliminate the “vanishing gradient” problem (2017)
 69. Feng, S., Wang, Y., Liu, L., Wang, D., Yu, G.: Attention based hierarchical LSTM network for context-aware microblog sentiment classification. *World Wide Web*. **22**(1), 59–81 (2019)
 70. Cui L, Zhang Y. Hierarchically-refined label attention network for sequence labeling. *ArXiv. abs/1908.08676* (2019)
 71. Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. Proceedings of the Twenty-Ninth AAAI conference on artificial intelligence Austin, Texas (2015)
 72. Nayyeri M, Zhou X, Vahdati S, Yazdi HS, Lehmann J. Adaptive margin ranking loss for knowledge graph embeddings via a corentropy objective function. *ArXiv* (2019) abs/1907.05336
 73. Berrada L, Zisserman A, Kumar MP. Smooth Loss Functions for Deep Top-k Classification. *ArXiv*. (2018) abs/1802.07595
 74. Niu G, Li Y, Tang C, et al. Relational Learning with Gated and Attentive Neighbor Aggregator for Few-Shot Knowledge Graph Completion. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery. 213–222 (2021)
 75. Craswell N. Mean Reciprocal Rank. In: Liu L, Özsu MT, eds. *Encyclopedia of database systems*. Boston, MA: Springer US:1703–1703 (2009)
 76. Chen H, Li G, Jiang W, Sun Y. Dynamic dual quaternion knowledge graph embedding. *Applied intelligence* (2022)
 77. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. *CoRR*.abs/1412.6980 (2015)
 78. Xu J, Li Z, Du B, Zhang M, Liu J. Reluplex made more practical: Leaky ReLU. Paper presented at: 2020 IEEE Symposium on computers and communications (ISCC); 7–10 (2020)
 79. Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. Proceedings of the twenty-eighth AAAI conference on artificial intelligence. Québec City, Québec, Canada (2014)
 80. Nickel M, Tresp V, Kriegl H-P. A three-way model for collective learning on multi-relational data. Proceedings of the 28th International conference on International conference on machine learning. Bellevue, Washington, USA (2011)

81. Yang B, Yih W-t, He X, Gao J, Deng L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. CoRR (2015) abs/1412.6575
82. Trouillon T, Welbl J, Riedel S, Gaussier É, Bouchard G. Complex embeddings for simple link prediction. Proceedings of the 33rd International conference on international conference on machine learning. 48. New York, NY, USA (2016)
83. Du Z, Zhou C, Ding M, Yang H, Tang J. Cognitive knowledge graph reasoning for one-shot relational learning. ArXiv (2019) abs/1906.05489
84. Hu, B.: Linking business models with technological innovation performance through organizational learning. Eur. Manag. J. **32**(4), 587–595 (2014)
85. Sauro, J., Lewis, J.R.: Chapter 4 - Did we meet or exceed our goal? In: Sauro, J., Lewis, J.R. (eds.) Quantifying the user experience, 2nd edn., pp. 39–60. Morgan Kaufmann, Boston (2016)
86. Reichheld FF. The one number you need to grow. Harvard business review. 81(12):46–54, 124. (2003)
87. He X, Chua T-S. Neural Factorization Machines for Sparse Predictive Analytics. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. Shinjuku, Tokyo, Japan (2017)
88. Wang X, He X, Cao Y, Liu M, Chua T-S. KGAT: Knowledge Graph Attention Network for Recommendation. Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining. (2019)
89. Hao Q, Xu Q, Yang Z, Huang Q. Learning unified embeddings for recommendation via meta-path semantics. In: Proceedings of the 29th ACM International conference on multimedia. association for computing machinery. 3909–3917 (2021)