doi:10.3772/j.issn.1006-6748.2021.02.006

NGDcrm: a numeric graph dependency-based conflict resolution method for knowledge graph①

Ma Jiangtao (马江涛)*, Wang Yanjun②**, Chen Xueting***, Qiao Yaqiong****
(*College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, P. R. China)
(**State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, P. R. China)
(***School of Computer & Communication Engineering, University of Science and Technology Beijing, Beijing 100083, P. R. China)
(****School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, P. R. China)

Abstract

Knowledge graph (KG) conflict resolution is to solve knowledge conflicts problem in the construction of KG. Aiming at the problem of KG conflict resolution, a KG conflict resolution algorithm NGDcrm is proposed, which is a numeric graph dependency-based conflict resolution method. NGDcrm utilizes the dependency graph to perform arithmetic calculation and predicate comparison of numerical entity knowledge in the KG. NGDcrm first uses a parallel segmentation method to segment the KG; then, it extracts the features of the KG according to KG embedding; finally, it uses numerical graph dependencies to detect and correct the wrong facts in the KG based on the extracted features. The experimental results on real data show that NGDcrm is better than the state-of-the-art knowledge conflict resolution method. Among them, the AUC value of NGDcrm on the DBpedia dataset is 15.4% higher than the state-of-the-art method.

Key words: dependency graph, knowledge conflict resolution, knowledge graph (KG), numeric graph dependency (NGD)

0 Introduction

Knowledge graph (KG) construction is an active research area[1-3]. Many KGs such as DBpedia[4], Read The Web[5] and YAGO[6] acquire knowledge from structured or semi-structured web resources. In the process of constructing the KG, some inconsistent or wrong knowledge will appear[7-8], which is called knowledge conflict in the KG. The method of automatically correcting these knowledge conflicts is called the knowledge conflict resolution method or KG correction error method. The existing knowledge conflict correction methods mainly utilize entity relationship embedding vector[9-10] or graph structure embedding vector to correct the wrong facts. Although these methods can correct many knowledge conflicts in KGs, they cannot find the inconsistent numerical data in the KG and have poor interpretability.

However, the numerical data in the KG often changes dynamically with time. This article focuses on the problem of inconsistent numerical data in the KG. Many facts in the KG change over time, so the entities and relationships in the KG also change. This change is called the evolution of the KG. During the evolution of the KG, noise, inconsistency, and even wrong facts often appear. Such errors will cause cascading propagation errors in KG completion and inference[11], which will lead to inconsistent data and knowledge conflicts in the KG. Therefore, in order to solve the problem of inconsistent knowledge conflicts during the evolution of KGs, researchers have proposed representation learning methods based on entity relationship embedding and graph structure embedding, and conflict fact correction method based on entity and relationship embedding vector and path embedding vector. These

① Supported by the Henan Province Science and Technology Department Foundation (No. 202102310237, 192102210133, 202102310295), the Doctoral Research Fund of Zhengzhou University of Light Industry (No. 2018BSJJ039) and the Internet Medical and Health Service Henan Collaborative Innovation Center Open Project Fund (No. IH2019006).

② To whom correspondence should be addressed. E-mail: wyj@zzuli.edu.cn

Received on June 17, 2020

methods take the time attribute of facts as an essential feature in the evolution of the KG, and use the disjoint, antecedent, and mutually exclusive relations of time to constrain time-based conflicting facts, and use the method of recurrent neural networks to detect conflicts. A lot of inconsistent facts appear to be adjusted, but these methods cannot resolve the conflict or inconsistency of the numerical data, and the interpretability is poor. Besides, another disadvantage of the existing KG conflict resolution methods is that the dependency relationships between entities are not fully utilized, which is why these methods are not sufficient. Therefore, new method needs to be proposed to solve knowledge conflicts problem in the process of KG evolution and to ensure the consistency of numerical data.

In order to solve the shortcomings of existing KG conflict resolution methods, a dependency graph-based KG conflict resolution method named NGDcrm is proposed, which uses the dependency relationship between entities in the KG to detect and correct knowledge conflicts in the KG. A large number of experimental results on real data sets show the effectiveness of the NGDcrm. In summary, the main contributions of this article are as follows:

(1) A KG conflict detection method NGDcrm is proposed, which finds and corrects conflicts of facts based on the dependency relationship between entities in the KG. This is the first attempt to apply dependency graph relationships to knowledge conflict correction of KGs.

(2) NGDcrm makes full use of the dependency relationship between entities in the KG for conflict discovery and correction, making the KG conflict discovery more interpretable. This solves the poor interpretability problem which is caused by the representation learning-based method.

(3) A large number of experimental results on DBpedia and YAGO2 show that NGDcrm is superior to existing knowledge conflict correction methods. The AUC value of NGDcrm on DBpedia is 15.4% higher than the best-known method CoCKG, and the AUC value of NGDcrm on YAGO2 is 16.1% higher than the best-known method CoCKG.

The rest of this paper is organized as follows. Section 1 summarizes the related work of KG conflict correction. Section 2 describes the KG conflict resolution problem. Section 3 gives the KG conflict detection and correction framework based on the numeric dependency graph. The experimental results and analysis are given in Section 4. At the end, Section 5 summarizes the work and gives the research direction in the future.

1 Related work

The existing KG conflict correction methods are divided into relational assertion-based methods and representation learning-based methods. Relational assertion-based methods require the restrictive knowledge of relationship between entities. Representation learning-based methods require the relationship between entities and the graph structure features of the KG.

1.1 Relational assertion-based methods

Researchers have proposed some error correction methods for the problem of relational assertion error detection in the KG. For example, researchers have proposed some methods for clearing the large-scale linked open data (LOD) error facts for DBpedia and never-ending language learning (NELL), but they still contain many relationship assertion errors that cannotbe detected by inference methods[12]. One of the main reasons for this type of problem is the lack of domain and scope restrictions or overly general restrictions. SDValidate[12] uses the statistical distribution of types and relationships to find false relationship assertions. Ref. [13] applied the type of entity similarity measure to outlier detection to find false relational assertions. These methods can effectively detect errors in DBpedia, but they require the presence of an information type declaration. Besides, more complex wrong entity errors containing the correct type cannot be detected.

Ref. [14] can deal with static fact conflicts, but cannot deal with fact conflicts that change over time. Ref. [15] used Markov logic networks to reduce errors in the temporal KG. They used hand-designed timing restriction rules to encode static facts with a binary relationship. The disadvantage of this method was that it was impossible to encode the timing information to suit the dynamic changes of the KG. Ref. [16] studied the problem of erroneous links in Wikipedia and DBpedia. They modeled Wikipedia links as some directed single-relationship graphs and proposed the LinkRank algorithm for ranking pages, in which links are ranked by the importance of relationships. LinkRank generates candidates for link correction and uses text features to train the support vector machine classifier. The number of association errors can be reduced by improving the quality of the source data, but this method cannot be directly applied to arbitrary KG.

Ref. [17] proposed a temporal inference and conflict resolution system TeCoRe for non-deterministic temporal KGs, which allows users to select rules, establish constraints and resolve conflict facts. This sys-

tem is built on the time expansion modules of the two probability inference engines, nRockIt and PSL solver, which can efficiently deal with time limit problems. TeCoRe transforms non-deterministic temporal KGs, inference rules, and constraints into weighted first-order logic that can be represented by Markov logic network and probabilistic soft logic. According to logical reasoning, inconsistent facts in the KG can be found. On this basis, they use the Markov logic network's numerical expansion method[18] to give uncertain temporal KG syntax and semantic support, and use the Datalog limit set to expand the KG pattern to detect inconsistencies facts in the KG.

1.2 Representation learning-based methods

KG conflict resolution and KG completion are inseparable. The process of KG completion is also the process of KG correcting errors and resolving knowledge conflicts. Recently, researchers have used representation learning methods to complete KGs, and use KG representations in low-dimensional dense vector spaces, that is, KG embedding models to complete KGs[19]. The triples between them are scored with confidence, the triple with the higher score is the correct candidate triple, and the triple with the lower score is likely to be the wrong triple. RESCAL[20] is one of the earliest KG embedding models. It performs tensor decomposition on the adjacent tensor of the KG, and the obtained feature vector corresponds to the entity embedding and core tensor relation matrix. TRESCAL[21] extends RESCAL by leveraging the domain and scope constraints of entity types and relationships to improve data quality and speedup the tensor decomposition process. The neural tensor model (NTN) represents each relationship as a bilinear tensor operator and a linear matrix operator. Other early embedding models include structural embedding (SE)[22], semantic matching energy (SME)[23], and the latent factor model (LFM)[24]. The translation-based embedding method is to express the relationship as a translation between the two entities, subject and object. TransE[25] is the first translation-based model proposed. TransE shares entities and relationships with the same embedding space. TransH[26] and TransR[27] show that transformations are performed in the relationship space and interact with entities spaces are different, so projection matrices are needed to map entities to relational spaces. CTransR[27] contains multiple relationship semantics, where a relationship may have multiple meanings as determined by the entity pair associated with the relationship. HolE[28] uses cyclic correlation as an operator to combine topics and object embeddings. The complex embedding[29] method takes the Hermitian fdot product of subject and object embeddings as a triplet score, which consists of real and imaginary vectors. The discriminant Gaifman model[30] embeds a subgraph pattern containing first-order rules in the model. Recently, some studies have questioned the performance of the new KGC embedding model. Most experiments rely on only two data sets (WN18 and FB15k), which contain many inverse relationships[31], therefore, some models may take advantage of this feature and may not perform well on other KGs. Studies have also shown that the relationship between candidate pairs can be a solid signal in some cases[32]. Besides, recent work has shown that hyperparameter tuning has been ignored, and simple methods (such as DistMult) can achieve the best results after tuning[32].

Researchers have studied various dependencies in the graph[33]. These dependencies are usually defined according to the graph model, aiming to capture inconsistencies between entities in the KG. They can be used for knowledge acquisition, KG augmentation, and spam detection in social networks. However, semantic inconsistencies in realistic graphics often involve numerical data. To catch such errors, arithmetic calculations and predicate comparisons are often necessary. Unfortunately, existing graph dependency studies do not support these computational expressions. Pick[34] is a model for solving conflicts of KG. This model determines the current data according to chronological order and time constraints, and enforces data consistency by using conditional function dependencies. Pick resolves knowledge conflicts by integrating current data and consistency inference into a single process and interacting with users. CoCKG is an automatic method for correcting confusion in KGs, which can resolve relationship declaration errors caused by instance confusion. This method relies on error detection methods and type predictors to evaluate the confidence of a corrected fact. It uses approximate string matching and uses search for entities with similar Internationalized Resource Identifier and Wikipedia disambiguation pages to find candidate instances for correcting facts. Ref. [35] proposed a class of numeric graph dependency (NGD) with arithmetic and comparison expressions to capture semantic inconsistencies in the KG. The author proposes an incremental algorithm based on NGD to detect errors in the KG. This method has good interpretability, but it does not solve the problem of KG conflict resolution. Therefore, based on this, this paper proposes an NGD-based knowledge conflict resolution method to solve the deficiencies of existing knowledge conflict resolution methods.

2 Problem description

Given a KG $G=(V, E, T)$, V is a finite set of entities, each node v in V is labeled as $L(v)$. $E \subseteq V \times V$ is a set of relations between entities, the label of edge e in E is $L(e)$, where (v, v') represents the edge from node v to v'. $T = \{<subject, relation, object>\}$ is a triple set of knowledge, *subject* and *object* represent two entities involved in knowledge, and relationship is the relationship between entities.

The matching of the pattern $Q[x']$ in the graph G is a mapping h from Q to G such that (a) for each node $u \in V_Q$, $L_Q(u) = L(h(u))$; (b) for Q each of $e=(u, u')$, $e'=(h(u), h(u'))$ is an edge in G, and $L_Q(e) = L(e')$. The graph pattern $Q[x]$ is expressed as $e_1 \otimes e_2$, where e_1 and e_2 are linear arithmetic expressions of $Q[x]$ (such as $+$, $-$, $\times$, $\div$), and $\otimes$ is a built-in comparison operator $=$, $\neq$, $<$, $\leqslant$, $>$, and $\geqslant$.

Numeric graph dependency (NGD) is often used for arithmetic expressions and built-in predicates. The form of numeric graph dependencies marked by NGD is $Q[x]$ $(X \rightarrow Y)$, where $Q[x]$ is a graph mode, called φ mode; X and Y are the literals set of $Q[x]$ (maybe empty set). NGDφ is a combination of: (a) topological constraint Q, used to identify entities in the graph; (b) attribute dependencies $X \rightarrow Y$, defined by linear arithmetic expressions connected to built-in predicate to enforce Q identification the entity. NGD supports the following two: linear arithmetic expressions with $+$, $-$, $\times$, $\div$, and $|\cdot|$; comparison with built-in predicates $=$, $\neq$, $<$, $\leqslant$, $>$, $\geqslant$.

As shown in Fig. 1 (left part of Fig. 1 is derived from Ref. [35]), the date related attributes of *BBC_Trust* institution in $G1$ is incorrect as its created date (2007-##-##) is later than its destroyed date (1946-08-28). The goal of KG error correction is to correct such errors. To correct the errors, the place of those two dates should be exchanged. In this way, $G1$ can be changed to $G1'$ without using external data sources.

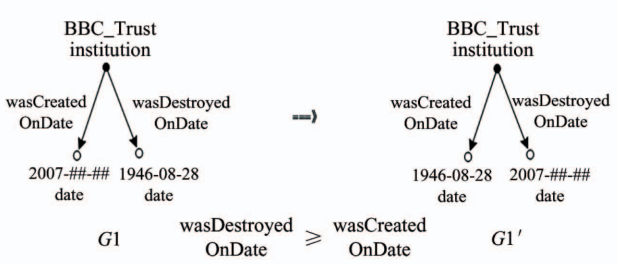

Fig. 1 The example of error correction with NGD in $G1$

As shown in Fig. 2 (the left half of Fig. 2 is derived from Ref. [35]), for *Bhonpur*. (Population Total) in $G2$, it should be equal to *Bhonpur*. (Female Population) + *Bhonpur*. (Male Population). The goal of KG error correction is to change $G2$ to $G2'$ without using external data sources, that is, to change the population total in $G2$ to 1372 (i.e., 600 + 722). The aim of KG conflict resolution is to correct numeric errors with NGD.

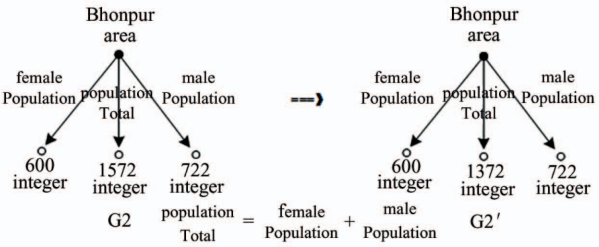

Fig. 2 The example of error correction with NGD in $G2$

3 Proposed solution

This paper proposes to use the NGD-based KG error correction method to correct the wrong knowledge in the KG. Firstly, the KG is segmented by the method of cutting edges, and then pattern matching is performed on the KG. Then the conflict detection is performed according to the graph dependency relationship between entities. Finally, the KG is corrected based on the graph's dependency constraint relationship. This solution uses a cut-edge method to perform parallel graph segmentation on the KG. It then uses KG representation learning to perform graph pattern matching, then discovers knowledge conflicts based on graph dependency constraints between entities, and finally uses NGD to analyse knowledge conflicts for error correction.

3.1 KG partition

Since the scale of the KG is very large, we first segment the KG. KG segmentation takes a simple KG $G=(V, E, T)$ as input, and each edge consists of a pair of unordered vertices (i.e., v, $u \in V$). $|V|=n$ is the number of vertices, and $|E|=m$ is the number of edges. The purpose of graph partitioning is to create k disjoint subsets (partitions) of vertices, $V=V_1 \cup \cdots \cup V_k$ to minimize the number of edges between the vertex sets. The total weight of these partitions across edges is named edge cut:

$$edgecut = \sum_{i=1}^{k} \sum_{v \in V_i} \sum_{u \in \Gamma(v),\, u \notin V_i} \theta\{v, u\} \quad (1)$$

This work focuses on the problem of balanced graph partitioning, which means that the size of each partition set is limited by the balance constraint, namely:

$$k\frac{\max_i |V_i|}{|V|} \leqslant 1+\epsilon \tag{2}$$

In order to facilitate the discussion of the effect of moving vertices, let $d_{int}(v)$ denote the indegree of v, that is, the sum of the weights of edges connecting v to its partition. Let $d_{ext}(v)$ denote the outdegree of v, that is, the sum of the weights of edges connecting v to partitions. Let $d_A(v)$ denote the sum of the weights of edges connecting vertex v to partition A. Finally, let $\Delta(v)$ denote the number of external partitions to which v is connected. Here the k-Way Hill-Scanning method[36] is used. All boundary vertices are inserted into the priority queue. To accurately sort vertices based on vertices in the priority queue, it needs to track:

$$gain = \max_{P_i \in P'} d_{P_i}(v) - d_{int}(v) \tag{3}$$

where, P' is the set of partitions that v is moved to and it will not violate the balance constraint. However, this will require frequent updates of vertex priorities as the partition weights and vertex connectivity change. Instead, the approximate gain associated with moving the vertex v out of its partition is prioritized as

$$priority = \frac{d_{ext}(v)}{\sqrt{\Delta(v)}} - d_{int}(v) \tag{4}$$

where, $\Delta(v)$ is the number of external partitions to which v is connected. For vertices connected to only a single external partition, Eq. (4) can simulate their priority accurately. For vertices connected to multiple external partitions, this facilitates vertices connected to fewer partitions without unduly penalizing vertices connected to too many partitions.

3.2 KG subgraph matching

KG embedding (KGE) method can answer graph pattern queries[37]. The existing KG query methods are limited to answering facts or one-way path queries. However, general graph queries involve bidirectional path queries. Also, considering the multi-hop path, the KGE method is susceptible to cumulative errors because errors at the edges can be amplified after multi-hop[38]. Therefore, a method based on KG embedding is proposed to solve the graph pattern matching of KG.

In this section, KG embedding is used to perform graph queries on incomplete KG without relying on subgraph isomorphism. Subgraph isomorphism (or subgraph matching) has traditionally been considered the key to answering graph pattern queries. However, when KG is incomplete or contains incorrect knowledge, the answer's quality depending on KG will drop sharply. To solve this problem, the subgraph isomorphism constraint in the proposed query model is removed. The KGE method is used to improve the accuracy of queries, and correct wrong knowledge[39]. In the model, we treat graph queries as a set of two-way path queries. To answer graph queries, we first need to answer each bidirectional path query through the KGE method. However, most KGE methods do not consider bidirectional path queries, and the operators used to calculate energy are irreversible. To take advantage of the regular and inverse constraints in the support method, the training process of the KGE method should be improved. The spanning tree from KG is sampled and then each tree is decomposed into a set of two-way paths between two terminal vertices (or 1-degree vertices) in the tree. Then, a set of wrong paths are created by changing one terminal vertex of the real path, and both the real and wrong path sets are used as the training data. This improvement enables the KGE method to learn the embedding of vertices and relationships that are used to answer graph pattern queries.

3.3 KG conflict detection

NGD provides unified rules to capture whether the values in the graph are consistent. Next, NGD is used as the data quality rule to study the error detection in the graph. To illustrate the problem of error detection in KG, the following symbols are obtained from Ref. [40]. Given a KG G and an NGD$\varphi = Q[x'](X \rightarrow Y)$ and we say that if $Gh \nRightarrow \varphi$, then a match $h(x)$ of Q in G violates φ, where Gh is induced subgraph of $h(x)$. For the set Σ of NGD, $Vio(\Sigma, G)$ is used to represent the set of all NGD violations in G, that is, $h(x) \in Vio(\Sigma, G)$. If NGDφ exists in G, let $h(x')$ violate φ in G. That is, $h(x')$ violates at least one NGD in Σ.

The input of the error detection problem is the Σ of a set of NGDs and a graph G; the output is the set of violations $Vio(\Sigma, G)$. That is, when NGD in Σ is used as the data quality rule, the set $Vio(\Sigma, G)$ of all inconsistent entities in G will be found. Given a set of NGDs Σ and graph G, its decision version is a verification problem to determine whether $G \models \Sigma$ is valid, such as $Vio(\Sigma, G) = \varnothing$. The verification problem of GFD is coNP-complete. Using GFDs as data quality rules, Ref. [35] had developed parallel algorithms for error detection. These algorithms are scalable in parallel, which guarantees the reduction of the running time of sequential algorithms when more processors are used. Therefore, when the graph becomes more massive, it can be expanded by adding computing resources. The experimental results of Ref. [35] have verified the parallel scalability and efficiency of the algorithm. This algorithm can be extended to NGDs. Indeed, to make the algorithm work with NGDs on graph G, which is dispersed and distributed across different

processors, the only change is to locally check the NGDs in each segment of G by adding arithmetic and comparison calculations. Graph pattern matching remains unchanged. For NGDs, the workload estimation and balancing strategy of Ref. [40] remains unchanged. These strategies allow the algorithm to scale in parallel. As a result, when algorithms use NGDs instead of GFDs, they maintain parallel scalability. Therefore, a scalable parallel algorithm can be designed for uniformly detecting semantically inconsistent information in a graph (whether it is a number or not) using NGDs.

In the large KG G, the cost of error detection is high, and the graph updates in real-time, which highlights the need to study incremental error detection. NGDcrm calculates $Vio(\Sigma, G)$ at first, and then calculates $Vio(\Sigma, G \oplus G)$ in response to updating ΔG to G incrementally. When ΔG is small, this is more efficient than recalculating $Vio(\Sigma, G \oplus \Delta G)$ from scratch, which is very useful in practice. NGDcrm defines unit updates as inserts or deletes that can simulate some modified edges. Insertion may introduce new nodes with labels, attributes, and values extracted from the entity, while the delete operation will only delete the link, and other nodes will not be affected.

3.4 Knowledge conflict resolution model

A sequential incremental error detection method is proposed to detect error triples in a KG with NGDs. On the NGDs defined by the connected graph pattern Q, there is a path between any two vertices in Q. Calculate tuples for local violations by finding matches for different connected components in Q, combining these partial matches to evaluate attribute dependencies, and then using the triples that identify the violations across multiple connected components.

Given the Σ of a set of NGDs, a KG G, and a batch update ΔG, a single processor is used to calculate $\Delta Vio(\Sigma, G, \Delta G)$. Increment subgraph matching of the KG by following update-driven evaluation, and use arithmetic and logical expressions to check dependencies. The overall framework of subgraph matching of the KG is: given a pattern Q and a graph G, each node u in the first pair of matches Q identifies a set of candidate matches $C(u)$. Then, in each iteration, its subroutine recursively expands the scheme S by matching a pattern node of Q with a node in the graph G, where S is a set of node pairs (u, v), indicating that v and the pattern node u match. For a partial solution S, the subroutine selects a pattern node u that has not yet matched, and refines $C(u)$ by selecting and pruning strategies according to some matching order. For each refined candidate v in $C(u)$, it checks whether v can effectively match u by checking the correspondence between the edge adjacent to u in Q and the edge connected to v in G. Eligible node pairs (u, v) are added to S, and then the subroutine is recursively called for further expansion until all pattern nodes match. When the subroutine is traced back, the partial solution S is restored.

To sum up, the proposed KG error correction method based on NGDs is shown in Algorithm 1. First, the k-way method is used to segment the KG, and then the KG embedded representation method is used to perform similarity subgraph matching in the KG. Then the conflict between the comparison operation and the arithmetic operation in the matching KG are found according to NGDs, and finally the errors are updated in the KG based on the semantic relationship without relying on external data sources.

Algorithm 1 NGDcrm: numeric graph dependency-based conflict resolution method

Input: knowlege graph $G = (V, E, T)$

Output: the rectified knowlege graph G: $G' = (V', E', T')$

1. $\sum_{i=1}^{n} G'_i$ = partition(G);
2. for ($j = 1$; $j < n + 1$; j ++)
3. KGE(G'_j);
4. $\sum_{k=1}^{m} G'_k$ = match (G', KGE (G'));
5. for ($k = 1$; $k < m + 1$; k ++)
6. errorfinder (NGDs(G'_k, $Vio(\Sigma, G'_k)$));
7. for ($k = 1$; $k < m + 1$; k ++)
8. rectify (NGDs(G'_k);
9. for ($k = 1$; $k < m + 1$; k ++)
10. $G' = G' \cup G'_k$;
11. return G'

4 Experimental results

In this paper, the effects of NGDcrm and baseline methods on knowledge conflict error correction of KG are compared on two types of data sets: DBpedia[4] and YAGO2[6]. True positive rate, false positive rate, positive predictive value, and $F1$ value can measure the performance of the proposed scheme. Experiments were performed on a cluster with Intel Celeron E5-2620 V3 CPU, Nvidia Tesla K80 GPU, 128GB memory, and CentOS6.4. The algorithms in the experiment were implemented in TensorFlow by using Python.

4.1 Datasets

(1) DBpedia KG contains 200 entity types with a

total of 1.72 million entities, 160 edge types, and 31 million edges, which can be downloaded from https://wiki.dbpedia.org/Datasets.

(2) YAGO2 is YAGO's upgraded version, there are 1.99 million entity nodes in 13 types of entities, and 5.65 million edges in 36 types of edges, which can be downloaded from https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/downloads/.

4.2 Test criteria and baseline methods

True positive rate, false positive rate, positive predictive value, and $F1$ are employed to evaluate the proposed KG conflict correction method. True positive rate (TPR) indicates the ratio of corrected conflict facts that indeed conflict facts, $TPR = TP/(TP + FN)$, where TP is the number of rectified facts, and these facts are wrong in original KG, and FN is the number of the fact that it is wrong facts but to be considered true fact. False positive rate ($FPR = FP/(FP + TN)$) indicates the proportion of conflicting facts that are incorrectly corrected and should not be corrected, where FP is the number of facts (correct facts) that are incorrectly corrected, TN is the number of correct facts. Positive predictive value ($PPV = TP/(TP + FP)$) represents the ratio of correcting correct conflicting facts to all correcting conflicting facts, where $F1$($F1 = 2PPV \times TPR/(PPV + TPR)$) is the harmonic mean of both PPV and TPR.

In this paper, the mainstream KG conflict resolution methods Pick[34] and CoCKg[41] are selected as baseline methods and compared with the proposed method NGDcrm.

4.3 Experimental results and analysis

The NGDcrm employed AMIE[42] to find 21 and 24 numeric dependency rules (such as wasCreatedOnDate ≤ wasDestroyedOnDate) in DBpedia and YAGO2, respectively. According to these numeric dependency rules, the NGDcrm found 362 and 108 numeric errors in the process of subgraph matching, which corrected 362 and 108 numeric errors in DBpedia and YAGO2, respectively. Fig. 3 shows the ROC comparison results of the proposed NGDcrm method with several baseline methods on the DBpedia dataset. As can be seen from Fig. 3 that NGDcrm is significantly better than several other methods. The *AUCs* of NGDcrm, CoCKG, and Pick were 0.8306, 0.7198, and 0.6598, respectively. The AUC of NGDcrm was 15.4% higher than that of CoCKG, which is in the second place.

Fig. 4 shows the ROC comparison results of the proposed NGDcrm method with several baseline methods on the YAGO2 dataset. Similar to the experimental results on the DBpedia data, NGDcrm is significantly better than the other baseline methods. When the false positive rate is 0.5, the true positive rates of NGDcrm, CoCKG, and Pick are 0.97, 0.91, and 0.85, respectively. Overall, the AUCs of NGDcrm, CoCKG, and Pick are 0.9002, 0.7751, and 0.7021, respectively. The *AUC* of NGDcrm is 16.1% higher than CoCKG's *AUC*.

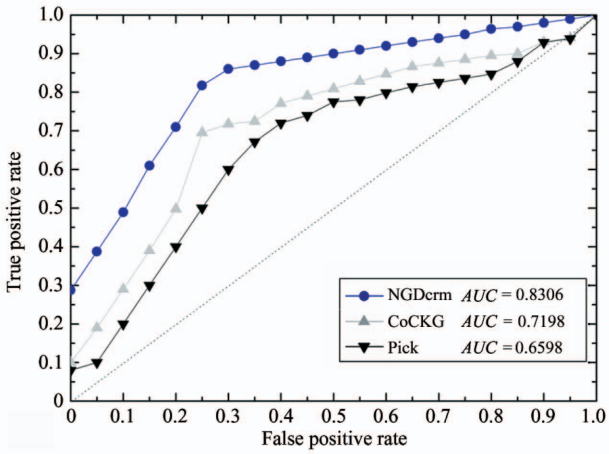

Fig. 3 ROC of NGDcrm and baseline methods on DBpedia dataset

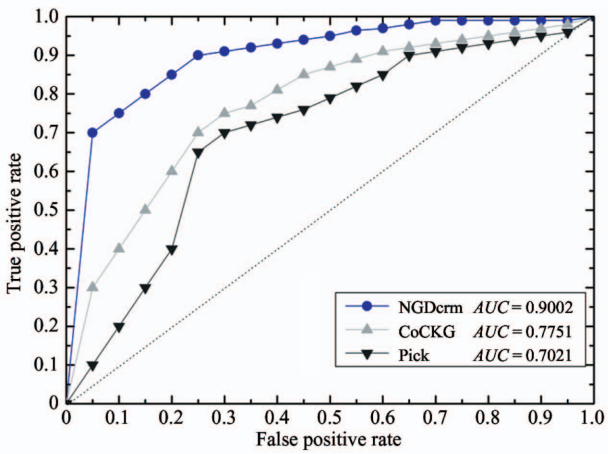

Fig. 4 The ROC of NGDcrm and baseline methods on YAGO2 dataset

Fig. 5 compares the $F1$ of NGDcrm with the baseline method on the DBpedia dataset. It can be seen from Fig. 5 that the $F1$ of NGDcrm is the highest. Among them, when the ratio of the constraint conditions on the x-axis is 0.5, the $F1$ values of NGDcrm, CoCKG, and Pick are 0.88, 0.76, and 0.68, respectively.

Fig. 6 compares the $F1$ of NGDcrm with the baseline method on the YAGO2 dataset. It can be seen from Fig. 6 that the $F1$ value of NGDcrm is the highest. Among them, when the ratio of the constraint conditions on the x-axis is 0.5, the $F1$ values of NGDcrm,

CoCKG, and Pick are 0.88, 0.77, and 0.71, respectively.

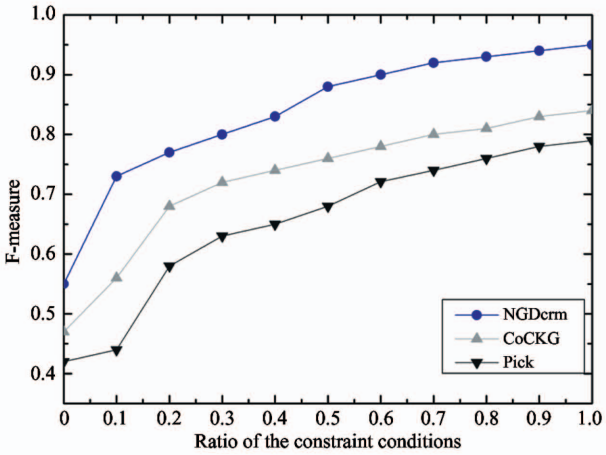

Fig. 5 F-measure of NGDcrm and baseline methods on DBpedia dataset

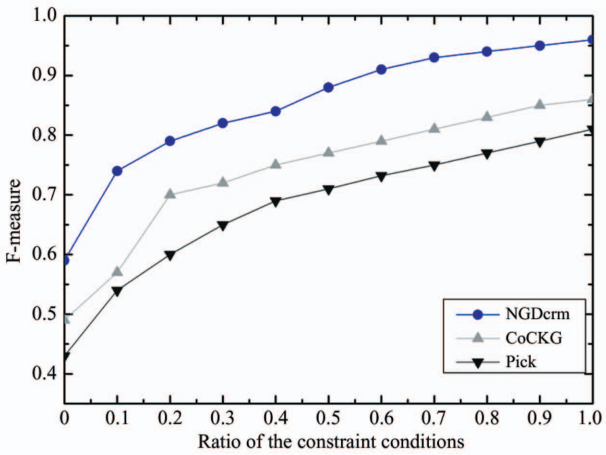

Fig. 6 F-measure of NGDcrm and baseline methods on YAGO2 dataset

Fig. 7 shows the running efficiency effect of NGDcrm and the baseline method on the DBpedia dataset. It can be seen from Fig. 7 that the running time of NGDcrm with the same data amount is the shortest, and its running efficiency is the highest. When the DBpedia dataset is increased by 40%, the running time of NGDcrm is 43%, 72% less than Pick, and CoCKG, respectively.

Fig. 8 shows the running efficiency effect of NGDcrm and the baseline method on the YAGO2 dataset. It can be seen from Fig. 8 that the running time of the same data amount NGDcrm is the shortest and its running efficiency is the highest. When the YAGO2 dataset is increased by 40%, the running time of NGDcrm is 50% and 67% less than that of Pick and CoCKG, respectively.

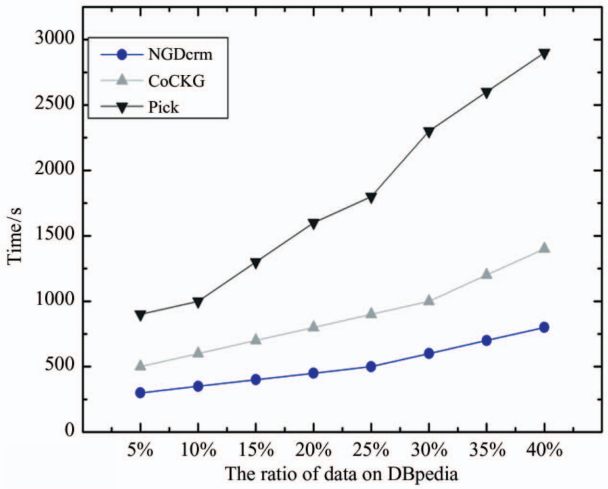

Fig. 7 Running time of NGDcrm and baseline methods on DBpedia dataset

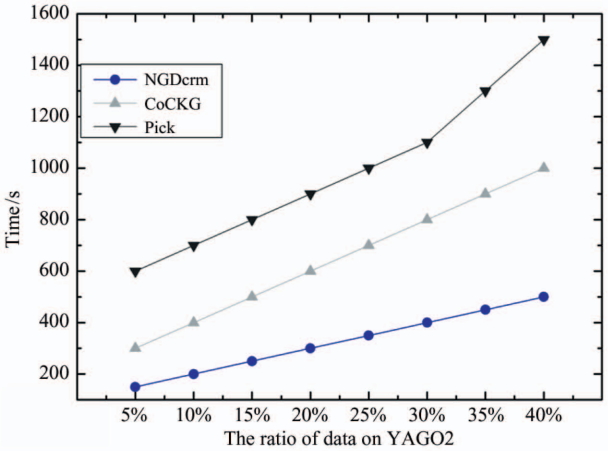

Fig. 8 Run time of NGDcrm and baseline methods on YAGO2 dataset

5 Conclusion

The problem of knowledge conflicts in KGs is analysed. Existing research ignores the dependency relationship between the values in KGs. A knowledge conflict resolution method NGDcrm is proposed, which combines parallel KG segmentation, embedded feature matching of KGs, and numerical graph dependencies. NGDcrm finds errors in the KG based on numerical graph dependencies, and does not rely on external data sources to resolve errors in the KG, thereby ensuring consistency in the KG construction. A large number of experiments on real datasets show that this method is superior to the current mainstream methods. NGDcrm shows that it is feasible to use numerical graph dependence to deal with knowledge conflicts problem in the process of constructing KG. In the future, this idea will be used to solve the dynamic change of entity relationships during the evolution of the KG over time.

References

[1] Dettmers T, Minervini P, Stenetorp P, et al. Convolutional 2d knowledge graph embeddings[C]// Proceedings of the 32nd AAAI Conference on Artificial Intelligence, Louisiana, USA, 2018: 1811-1818

[2] Consoli S, Presutti V, Reforgiato Recupero D, et al. Producing linked data for smart cities: the case of Catania [J]. *Big Data Research*, 2017(7): 1-15

[3] Dai W W, Dubinin V N, Christensen J H, et al. Toward self-manageable and adaptive industrial cyber-physical systems with knowledge-driven autonomic service management[J]. *IEEE Transactions on Industrial Informatics*, 2017(13): 725-736

[4] Lehmann J, Isele R, Jakob M, et al. DBpedia: a large-scale, multilingual knowledge base extracted from Wikipedia[J]. *Semantic Web*, 2015(6): 167-195

[5] Mitchell T M, Cohen W W, Jr E R H, et al. Never-ending learning[J]. *Communication of ACM*, 2018,61(5): 103-115

[6] Rebele T, Suchanek F, Hoffart J, et al. YAGO: a multilingual knowledge base from Wikipedia, Wordnet, and Geonames[C]// Proceedings of the 15th International Semantic Web Conference, Kobe, Japan, 2016: 177-185

[7] Pradhan R, Aref W G, Prabhakar S, Leveraging data relationships to resolve conflicts from disparate data sources [C]// Proceedings of 29th International Conference on the Database and Expert Systems Applications, Regensburg, Germany, 2018: 99-115

[8] Kalchgruber P, Klas W, JnouN b, et al. FactCheck-identify and fix conflicting data on the Web[C]// Proceedings of the 18th International Conference on Web Engineering, Cáceres, Spain, 2018:312-320

[9] Aljefri Y M, Hipel K W, Fang L. General hypergame analysis within the graph model for conflict resolution[J]. *Operations and Logistics*, 2020,7: 1-16

[10] Rêgo L C, dos Santos A M. Upper and lower probabilistic preferences in the graph model for conflict resolution[J]. *APPROX Reason*,2018,98: 96-111

[11] Chen Y, Wang D Z. Knowledge expansion over probabilistic knowledge bases[C]// Proceedings of the International Conference on Management of Data, Snowbird, USA, 2014: 649-660

[12] Paulheim H, Bizer C. Improving the quality of linked data using statistical distributions[J]. *Semantic Web*,2014, 10:63-86

[13] Debattista J, Lange C, Auer S. A preliminary investigation towards improving linked data quality using distance-based outlier detection[C]// Proceedings of the 6th Joint International Conference on Semantic Technology, Singapore, 2016:116-124

[14] Dylla M, Miliaraki I, Theobald M. A temporal-probabilistic database model for information extraction[J]. *VLDB Endowment*, 2013,6: 1810-1821

[15] Chekol M W, Huber J, Meilicke C, et al. Markov logic networks with numerical constraints[C]// Proceedings of the 22nd European Conference on Artificial Intelligence, Hague, Netherlands, 2016:1017-1025

[16] Wang C, Zhang R, He X, et al. Error link detection and correction in Wikipedia[C]// Proceedings of the 25th ACM International Conference on Information and Knowledge Management, Indianapolis, USA, 2016:307-316

[17] Chekol M W, Pirrò G, Schoenfisch J, et al. TeCoRe: temporal conflict resolution in knowledge graphs[J]. *VLDB Endowment*. 2017,10(12): 1929-1932

[18] Chekol M W, Pirrò G, Schoenfisch J, et al. Marrying uncertainty and time in knowledge graphs[C]// Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, USA, 2017: 88-94

[19] Wang Q, Mao Z, Wang B, et al. Knowledge graph embedding: a survey of approaches and applications[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2017,29(12): 2724-2743

[20] Nickel M, Tresp V, Kriegel H-P. A three-way model for collective learning on multi-relational data[C]// Proceedings of the 28th International Conference on Machine Learning, Washington, USA, 2011: 809-816

[21] Chang K W, Yih S W, Yang B, et al. Typed tensor decomposition of knowledge bases for relation extraction[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural, Doha, Qatar, 2014:1568-1579

[22] Bordes A, Weston J, Collobert R, et al. Learning structured embeddings of knowledge bases[C]// Proceedings of the the 25th AAAI Conference on Artificial Intelligence, San Francisco, USA, 2011: 301-306

[23] Bordes A, Glorot X, Weston J, et al. Joint learning of words and meaning representations for open-text semantic parsing[C]// Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, Canary Islands, Spain, 2012: 127-135

[24] Jenatton R, Roux N L, Bordes A, et al. A latent factor model for highly multi-relational data[C]// Proceedings of the 26th Annual Conference on Neural Information Processing Systems, Lake Tohoe, USA, 2012: 3167-3175

[25] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[C]// Proceedings of the 27th Annual Conference on Neural Information Processing Systems, Lake Tohoe, USA, 2013: 2787-2795

[26] Wang Z, Zhang J, Feng J, et al. Knowledge graph embedding by translating on hyperplanes[C]// Proceedings of the 28th AAAI Conference on Artificial Intelligence, Quebec, Canada, 2014:1112-1119

[27] Lin Y, Liu Z, Sun M, et al. Learning entity and relation embeddings for knowledge graph completion[C]// Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, USA, 2015: 2181-2187

[28] Nickel M, Rosasco L, Poggio T A, et al. Holographic embeddings of knowledge graphs[C]// Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, USA,2016:1955-1961

[29] Trouillon T, Welbl J, Riedel S, et al. Complex embeddings for simple link prediction[C]// Proceedings of the 33nd International Conference on Machine Learning, New York, USA, 2016: 2071-2080

[30] Niepert M. Discriminative Gaifman models[C]// Proceedings of the Annual Conference on Neural Information

Processing Systems, Barcelona, Spain, 2016:3405-3413
[31] Toutanova K, Chen D Q. Observed versus latent features for knowledge base and text inference[C]// Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, Beijing, China, 2015: 57-66
[32] Kadlec R, Bajgar O, Kleindienst J. Knowledge base completion: baselines strike back[C]// Proceedings of the 2nd Workshop on Representation Learning for NLP, Vancouver, Canada, 2017:69-74
[33] Fan W, Lu P. Dependencies for graphs[J]. *ACM Transactions on Database Systems*, 2019, 44:1-40
[34] Fan W, Geerts F, Tang N, et al. Inferring data currency and consistency for conflict resolution[C]// Proceedings of the 29th IEEE International Conference on Data Engineering, Brisbane, Australia, 2013: 470-481
[35] Fan W, Liu X, Lu P, et al. Catching numeric inconsistencies in graphs[C]// Proceedings of the 2018 International Conference on Management of Data, New York, USA, 2018: 381-393
[36] LaSalle D, Karypis G. A parallel hill-climbing refinement algorithm for graph partitioning[C]// Proceedings of the 45th International Conference on Parallel Processing, Philadelphia, USA, 2016: 236-241
[37] Hong S, Park N, Chakraborty T, et al. PAGE: answering graph pattern queries via knowledge graph embedding [C]// Proceedings of the 7th International Congress on BigData, Held as Part of the Services Conference Federation, Seattle, USA, 2018: 87-99
[38] Guu K, Miller J, Liang P, et al. Traversing knowledge graphs in vector space[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 2015: 318-327
[39] Ren X, Wu Z, He W, et al. CoType: joint extraction of typed entities and relations with knowledge bases[C]// Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 2017:1015-1024
[40] Fan W, Wu Y, Xu J. Functional dependencies for graphs [C]// Proceedings of the 2016 International Conference on Management of Data, San Francisco, USA, 2016: 1843-1857
[41] Melo A. Automatic Refinement of Large-scale Cross-domain Knowledge Graphs[D]. Germany: University of Mannheim, 2018
[42] Galárraga L A, Teflioudi C, Hose K, et al. AMIE: association rule mining under incomplete evidence in ontological knowledge bases[C]// Proceedings of the 22nd International World Wide Web Conference, Rio de Janeiro, Brazil, 2013:413-422

Ma Jiangtao, born in 1981. He received his Ph. D degree in State Key Laboratory of Mathematical Engineering and Advanced Computing of Information Engineering University in 2018. He also received his B. S. and M. S. degrees from Zhengzhou University of Light Industry in 2004 and 2007 respectively. His research interests include knowledge graph, social network analysis and data mining.