AUT Jouranl of Modeling and Simulation

AUT J. Model. Simul., 52(2) (2020) 269-282
DOI: 10.22060/miscj.2020.18353.5207

New Configurations for the Correction of the RDF knowledge bases

Farhad Abedini [1*], Mohammad Reza Keyvanpour [2,*], Mohammad Bagher Menhaj [3]

[1] Department of Computer Engineering, Roudsar and Amlash Branch, Islamic Azad University, Roudsar, Iran.
[2] Department of Computer Engineering, Alzahra University, Vanak, Tehran, Iran.
[3] Center of Excellence in Control and Robotics, Electrical Engineering Department, Amirkabir University of Technology, 424, Hafez Ave., Tehran, Iran.

ABSTRACT: In each RDF knowledge base, several errors must be corrected by correction methods. Correction methods can be divided into three classes for the correction of outliers, inconsistencies, and erroneous relations. RDF knowledge base outliers can be considered as two types of outlier entities and triples. Inconsistent triples are corrected by inconsistency correction methods and there are many erroneous relation correction methods that each of them is used for a special objective. The variety of these errors is so wide so that no correction method could be able to cover them all. Most of the correction methods have been focused only on some of these errors, so a comprehensive study is mandatory to cover all of these elements for different objectives. Nevertheless, a couple of survey articles on the RDF knowledge base correction exist, but they are out-dated and did not present different configurations of these errors for various objectives. Since there is no configuration in this field, a new general configuration of the RDF knowledge base correction for a different objective is proposed here that can cover these various errors. In this configuration, a new classification of the errors is presented in which they are divided into three classes. The correction of each class is performed in a separate step. Finally, the state-of-the-art approach of each step is identified for each objective and a different configuration of these methods will be proposed for various objectives.

Review History:

Received: Mar. 03, 2020
Revised: Jun. 05, 2020
Accepted: Jun. 06, 2020
Available Online: Dec. 01, 2020

Keywords:

RDF knowledge base correction
inconsistency
outliers
erroneous relations.

1- INTRODUCTION

In each RDF knowledge base like Freebase [1] and WordNet [2], there are many real-world facts. Here, each RDF knowledge base is called briefly KB that has a triple structure for each fact. The structure of a triple is $(e1, R, e2)$ in which $e1$ and $e2$ are KB entities and R is a relation between them [3]. For instance, "Elvis Presley has won Grammy award" is a fact that can be displayed by the triple (*Elvis Presley*, *hasWonPrize, GrammyAward*) that "*Elvis Presley*" is $e1$, "*GrammyAward*" is $e2$ and the "*hasWonPrize*" is R. The structure of knowledge base can also be represented by a graph G (N, E), in which N is the set of nodes and E is the set of edges. In the knowledge base, the entities e_i can be considered as graph nodes ($e_i \in N$) and the relation between them as graph edges ($r_i \in E$). Such a graph is also called the *knowledge graph* [4-6]. For example, the graph of Table 1 is shown in Fig. 1. There are other data models that have knowledge graph structure such as social networks data model, but the focus of this paper is only on the KG that is used as a background knowledge in the semantic web [7, 8].

The errors of a KB are corrected by KB correction methods. These errors may happen after or before the KB creation. After KB creation, new facts may be added to the KB by the enrichment methods that may cause new errors that are corrected by *post-correction* methods [9]. Also, some errors may exist in the KB, previously [3] that must be corrected by *pre-correction* methods. The literature of pre-correction is rich [10-17], but there are few studies on the post-correction methods. The focus of this paper is on the post-correction by RDF mining methods that are divided into non-embedding and embedding approaches [4]. Embedding post-correction methods embed entities and relations to a vector space [4, 18-24]. Through this conversion, learning steps can be performed effectively.

There is a Correction Tower [9] that is able to correct the errors of inconsistencies, outliers, and erroneous relations in the KB by embedding post-correction methods. This tower is useful only for special objectives, but there are some other objectives like numerical correction, non-embedding correction, Pre-correction, Error avoidance, etc. thus, it is mandatory to investigate other configurations of the correction. For this goal, a general configuration is proposed in this paper based on the Correction Tower that is called GKBC. The GKBC includes three steps for the correction of inconsistency, erroneous relations, and outliers. In this paper, suitable methods of each GKBC step for each objective are proposed in different configurations.

Corresponding author's email: abedini.ac

Table 1. A part of an RDF KB

subject	predicate	object
ElvisPresley	hasWonPrize	GrammyAward
ElvisPresley	bornInYear	1935
ElvisPresley	bornIn	Tupelo
Tupelo	locatedIn	Mississippi
Mississippi	locatedIn	USA
ElvisPresley	type	singer
singer	subClassOf	Person

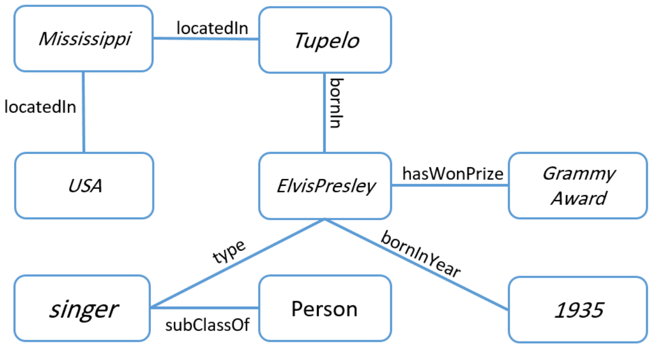

Fig. 1. Knowledge graph structure of the RDF KB

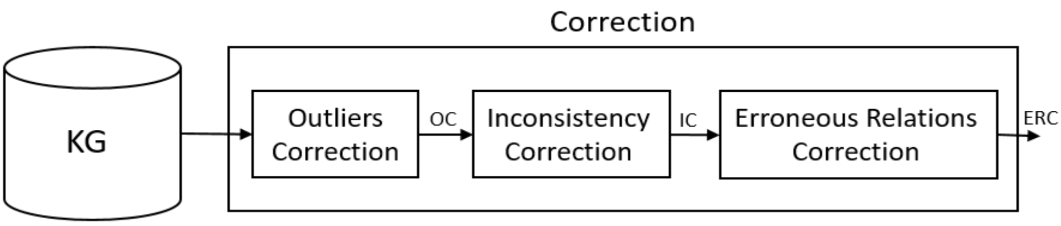

Fig. 2. GKBC: A General configuration of the KB Correction

In related studies, the correction of the inconsistencies has less been considered. Although some of the inconsistencies are corrected by outlier and erroneous relations correction methods, many inconsistency errors cannot be detected by these methods. Thus, the correction of inconsistencies is considered in the proposed GKBC as a separate step. In the proposed GKBC, a suitable approach is detected for each step in each objective. Based on these findings, new correction configurations will be proposed for different objectives. Finally, the evaluation of each configuration will be illustrated.

In the following, Section 2 investigates related studies. In Section 3, the GKBC general configuration is proposed. In Section 4, the methods of outlier correction are studied. These studies are performed for the inconsistencies an erroneous relations in Sections 5 and 6. In Section 7, the evaluation results of investigated methods are presented, and based on them the best configurations of the GKBC are organized for each objective. Finally, the conclusion is presented in Section 7.

2- RELATED STUDIES

There are many KB correction methods in which some errors of a KB are corrected. For example, there are some methods such as [25, 26] for detecting of the KB outliers. These methods mine only numeric entities by the clustering. Other methods such as [27] correct outlier triples by the clustering and classification methods. Also, the method of [28] detects outlier numeric entities by the clustering of numeric RDF data. Outliers of a KB are detected for two objectives. In some methods such as [25-28], outlier detection is used to find other errors. Another objective is to preprocess an operation such as KG embedding [29].

The method of [30] focuses only on the inconsistency errors. So far, existing embedding methods of the correction did not pay to only inconsistency errors, except EPCI [30]. Most of them focus on the correction of the outliers and erroneous relations. For instance, methods of PaTyBRED [31] and SDValidate [32] detect and delete the erroneous relations and outliers. The goal of CoCKG [33] and LinkRank [34] is

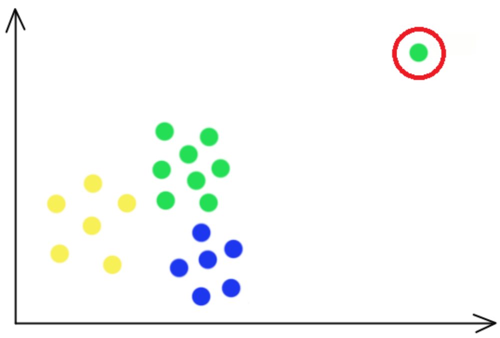

Fig. 3. a sample of the outlier

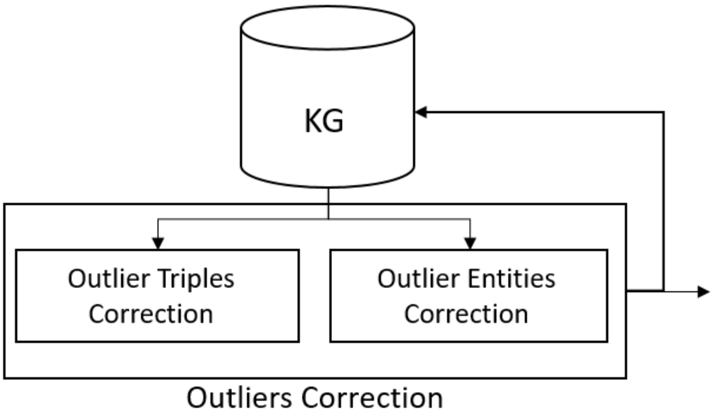

Fig. 4. Proposed outliers correction

to repair erroneous links in addition to error detection, but their repairing results are not acceptable [33]. The method of IncDect [35] has been recently introduced that directly pay to the inconsistency. This method can work only on the numeric data and is a non-embedding method that is not able to repair errors.

On the other hand, PaTyBRED [31] and SDValidate [32] are the methods that correct erroneous relations. Thus, there are few general studies such as Correction Tower [9] that investigate all these errors. In this paper, a general configuration is presented for this reason.

The errors of previous correction studies have been classified by different applications [36]. Here, correction methods can be classified by types of errors. Therefore, three classes are proposed: *outlier entities and triples*, *inconsistent triples*, and *erroneous relations*. Because of these error variety, most previous studies were focused only on some of these errors for the desired application [36]. A general study on these three classes of the errors for different objectives does not exist. Therefore, a new general configuration of the KB correction is proposed which is called *GKBC* in this paper.

3- GKBC: A GENERAL CONFIGURATION

The proposed GKBC general configuration is presented in Fig. 2. In this general configuration, the correction of each class of the errors is performed by a separate step. In this figure, the output of the first step is briefly shown as *OC*. This output includes the triples that its outliers have been corrected. The next step output is *IC* in which the consistencies have been also corrected. The output of the next step is *ERC* in which the erroneous triples are also corrected. There are many different methods for each step of GKBC that each of them is suitable for a special objective. In the next sections, they are investigated and then in Section 6 and after evaluation, suitable methods of each objective will be proposed.

In GKBC configuration, different KB errors are corrected. These errors is divided into three classes of outliers, inconsistencies and erroneous relations. Each class of these errors are corrected by a step of the GKBC, also some errors can be corrected by two or more steps. These steps are explained bellow.

4- FIRST STEP OF GKBC

In the first step of GKBC, outliers of KB must be corrected. Fig. 3 show a sample of an outlier that is different from others [36]. There are some methods such as [25, 26] for detecting the KB outliers that are able to detect only numeric outliers. These methods use clustering algorithms for outlier detection.

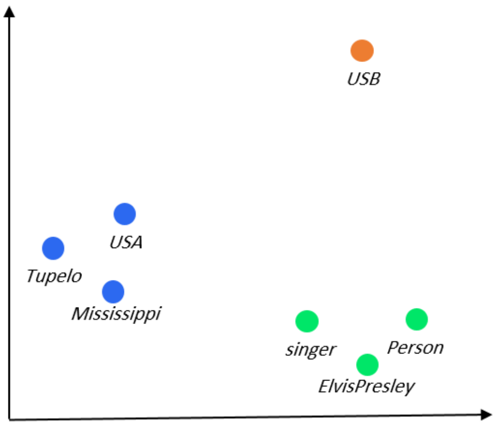

Fig. 5. A sample of outlier entity

Table 2. Comparison of prominent outlier detection methods in the KB

Method	Outlier Type	Triples /Entities	Aim	Technique	Embedding Type	Mining Method
Wienand et al. [26]	Numeric	Entities	Detecting Incorrect Numeric Data	IQR KDE	Non-Embedding	Clustering
Fleischhacker et al. [25]	Numeric	Entities	Detecting Incorrect Numeric liked Data	LOF	Non-Embedding	Clustering
Huiying et al. [28]	Numeric Outlier Interlinks	Triples	Detecting Incorrect Numeric Interlinks	Bayesian Classifier	Non-Embedding	Classification
Paulheim [27]	Outlier Interlinks	Triples	Detecting Wrong Interlinks	LOF LoOP SVM	Embedding	Clustering Classification

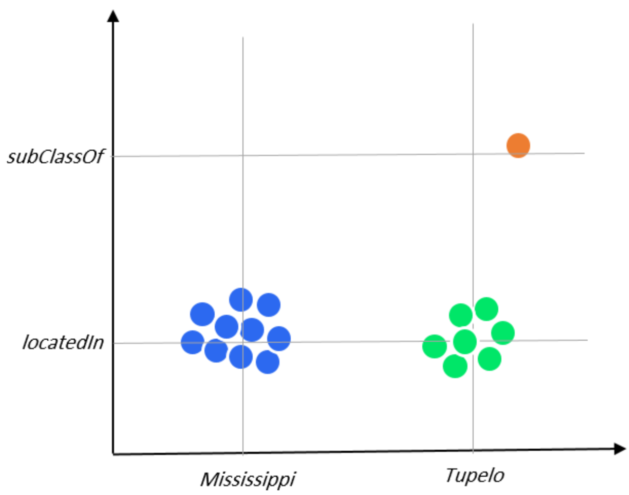

Fig. 6. a sample of outlier triple

Other methods like [27] use also classification algorithms. These outliers can be divided into two objectives. In many methods like [25-28], the detection is done to identify other errors such as inconsistencies. In the second objective, this method is used for the preprocessing of KB embedding [29]. Since there are two types of outliers in KBs, this step is divided into two parts of *outlier triples correction* and *outlier entities correction* that is shown in Fig 4. In [29], a KB is considered

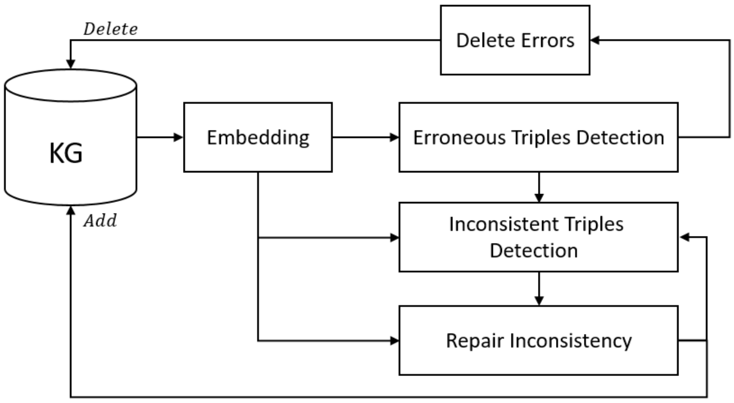

Fig. 7. The steps of an inconsistency correction

as a knowledge graph in which nodes are entities, and linkages can be KB triples. In this reference, two outlier types were nodes and linkages instead of entities and triples. The proposed method of outlier correction is able to correct both outlier triples and outlier entities in two parallel parts.

4.1. The Correction of Outlier Entities

In this step, outlier entities must be found from all entities of a KB. A sample of the outlier entity (USB) is displayed in Fig. 5. Some methods such as [26] are able to detect only numeric outliers so that they cannot correct non-numeric outliers. These methods are non-embedding approaches that by the composition of special clustering methods such as KDE [37] and IQR [26]can detect numeric outliers. These special methods are used because dataset distribution is not normal. Therefore, normal methods have not proper results and the methods of KDE and IQR are replaced. On the other hand, other methods such as [25, 38] were introduced for detecting natural outliers from other outliers. In these approaches, external repositories must be used, but it is not in the field of this manuscript. Numeric outliers can be detected by cross-checking and usage of these external repositories. These approaches use LOF [39] to cluster the outliers and they are not embedding methods.

Although the focus of these methods was on some KBs that there are links between them, but in this paper, we have only one KB. Thus, the same methods can be proposed to find outlier entities, based on the methods of outlier nodes detection [29].

4.2. The Correction of Outlier Triples

To study the outlier triples, there are two points of view. First, erroneous triples can find by detection of outlier triples. In the second point, the results of the next steps can improve by outlier triples detection that the focus of this paper is on this point. As a sample, the correction of outlier triples before the KB embedding can improve the embedding results. In the evaluation section, this improvement will be shown. In Fig. 6, a sample of the triple outlier is presented.

Wienand et al. introduced an important method in [27] for the purpose of outlier triples detection. They detect wrong links between some different datasets. Here, these links can be considered as outlier triples. In this paper, outliers in a KB is detected, but in [27], the outliers of more than two KBs are recognized. In this method, the links of KBs are embedded in the feature vector space. This method utilizes *types* and *properties* features for embedding, but normal embedding methods use latent features by neural networks to embed feature vectors. Outlier detection of this method is an unsupervised method same as LOF [39]. Using this method, we can detect the outliers of a KB.

The method of [28] has suitable results in numeric outliers. In this approach, a non-embedding method is used to find outlier links using arithmetic relations learning by probabilistic modeling [28]. This method can detect only date and numeric outliers and cannot recognize string links such as [27]. In the methods of [28] and [27], the focus is only on interlinks between some KBs and LOD, but the goal of this manuscript is the detection of the outlier triples in only one KB.

Studied methods of outlier detection in the KB are compared in Table **2.**. The methods of [25] and [26] can detect the outliers of numeric entities and cannot recognize string ones. The first method can work with a KB by IQR and KDE techniques, and another method is performed in some KBs by LOF. The mining methods of these approaches is clustering and they don't embed KB in a vector space.

Other methods of this table detect triples as KB outliers. These triples are interlinks between some KBs. In [28], incorrect numeric interlinks are detected, and in [27], all wrong interlinks are investigated. The mining method of [28] is the classification by Bayesian Classifier and the mining method of [27] is both classification and clustering by LOF, LoOP, and SVM. Embedding type of [28] is non-embedding, but its feature space does not obtain from a learning method.

5- SECOND STEP OF GKBC

In the second step of GKBC, inconsistency errors

Table 3. Comparison of prominent related methods for the inconsistency correction

Method	Embedding Type	Correction Type	Post/Pre	Error Type	Mining Method
SOFIE	Non Embedding	Avoid	Pre	Relation Errors Inconsistency	Reasoning
PROSPERIA	Non Embedding	Avoid	Pre	Relation Errors Inconsistency	Reasoning
DBpedia Enrichment	Non Embedding	Avoid	Pre	Inconsistency	Stochastic
PION	Non Embedding	Avoid	Pre	Inconsistency	Reasoning
Temporal Consistency	Non Embedding	Avoid	Pre	Inconsistency	Reasoning
Big RDF Consistency	Non Embedding	Avoid	Pre	Inconsistency	Rule Based
SDValidate	Embedding	Detect	Post	Relation Errors Outliers	Embedding Classifier
PaTyBRED	Embedding	Detect	Post	Relation Errors Outliers	PRA Embedding Classifier
LinkRank	Non Embedding	Repair	Post	Erroneous Links	SVM Classifier
CoCKG	Embedding	Repair	Post	Erroneous Links Outliers	PRA Embedding Classifier
IncDect	Non Embedding	Detect	Post	Inconsistency	Graph Functional Dependency
EPCI	Embedding	Repair	Post	Inconsistency	NTN

Table 4: Methods for correction of erroneous relations

Method	Outlier Type	Triples /Entities	Aim	Technique	Embedding Type	Mining Method
Wienand et al. [26]	Numeric	Entities	Detecting Incorrect Numeric Data	IQR KDE	Non-Embedding	Clustering
Fleischhacker et al. [25]	Numeric	Entities	Detecting Incorrect Numeric liked Data	LOF	Non-Embedding	Clustering
Huiying et al. [28]	Numeric Outlier Interlinks	Triples	Detecting Incorrect Numeric Interlinks	Bayesian Classifier	Non-Embedding	Classification
Paulheim [27]	Outlier Interlinks	Triples	Detecting Wrong Interlinks	LOF LoOP SVM	Embedding	Clustering Classification

are corrected. The amount of KB incompatibilities and contradictions specifies its inconsistencies[40]. Generally, few KB correction methods have been paid to the inconsistency correction, because the inconsistency is investigated often before KB creation as pre-correction [10, 40-44]. In [30], a post-correction method was recently proposed by the authors that is called *EPCI* in which the inconsistency is corrected. The steps of its inconsistency correction are shown in Fig. 7.

Based on what was said above, the correction methods are studied from different points of view. A comparison of prominent related studies based on these perspectives is displayed in Table 3.

. In the first point, the correction methods can either *avoid* the errors or *detect* and *repair* them. Some methods such as [31, 32, 35] only detect and delete errors. On the other hand, a few numbers of approaches such as [33] can repair the errors after the detection step, but these methods have not good results [33]. Nevertheless, related works in the field of inconsistency correction such as [10, 40-44] mostly avoid errors. For these reasons, an inconsistency correction method (EPCI) was introduced by authors in [30] in which inconsistency errors are repaired with suitable accuracy.

From the second point, correction methods are divided into two classes of *non-embedding* and *embedding*. Embedding approaches convert the knowledge base into a vector space [4] to facilitate the operations. Although existing embedding-based KB correction methods such as [4, 31, 33, 36, 45-51] have acceptable accuracy, so far they did not correct the inconsistencies, except EPCI [30]. Also, most existing inconsistencies correction approaches are non-embeddings such as [35, 40-43, 52-55].

Based on the third point, correction methods are either post-correction or pre-correction. The approaches of pre-correction are able to correct errors before the creation of the KB [10, 13, 44, 56-58]. Otherwise, post-correction approaches such as [31, 33, 35, 40, 49, 51, 59, 60] correct errors after KB creation. Existing inconsistencies post-correction approaches focus only on the correction of the numeric errors [25, 26, 28, 35, 55], but the post-correction of inconsistencies approaches are able also to correct non-numeric ones. Based on these points, important correction studies are compared in Table 3.

In this table, The approaches of PROSPERIA [44] and SOFIE [10] and are non-embedding approaches in which the inconsistencies of the YAGO KB are avoided by an extraction step and they are also corrected as the pre-correction approach. By these approaches, some candidate facts are extracted from an external resource, and the facts that are inconsistent are recognized with disambiguation algorithms [14, 15, 57, 61] from these candidate facts. In [41] DBpedia enrichment was introduced. This approach is a pre-correction method and able to correct the inconsistent facts of the DBpedia KB using a learning approach. PION [42] is another pre-correction method in which ontology inconsistencies are avoided. In the method of Big RDF Consistency [40], inconsistencies of RDF data [62] are detected by a rule-based, pre-correction, and non-embedding approach. Also, the method of Temporal Consistency [43] studied timely inconsistencies that is similar to the last investigated methods.

In the related studies of embedding-based correction methods, only EPCI [30] focused on inconsistency errors, exclusively. They have been mostly focused on outlier correction and erroneous relations correction. For example, SDValidate [32] and PaTyBRED [31] can detect and then delete the outliers and erroneous relations. The objective of LinkRank [34] and CoCKG [33] is repairing of erroneous interlinks after error detection, but the results of repairing are not suitable [33]. IncDect [35] is a new method that can directly investigate inconsistencies. IncDect can study only the non-embedding numeric data that cannot repair errors. In this table, post-correction and pre-correction are briefly called “post” and “pre”.

6- THIRD STEP OF GKBC

In the third step of GKBC, erroneous relations must be detected. These errors have different reasons. The first reason is that resources such as Wikipedia have some incorrect data and during fact extraction, erroneous relations may be extracted [3, 11, 56]. The second reason is that some erroneous relations may happen because of KB refinement [7, 36]. These relations must be corrected. Existing methods for this correction are divided into classes of non-embedding and embedding. Most of these methods are compared in Table 4.

Embedding-based correction methods of erroneous relations convert the KG to a vector space. This space gives a score value g [21, 23] to each KB triple. One threshold value τ_i is assigned to each KB relation r_i. For each r_i, if $g(s,r_i,o) > \tau_i$ then the triple (s,r_i,o) has true relation, otherwise r_i is a false relation for this triple [63]. By a classification method, these erroneous relations can be detected. For instance, in the Knowledge Vault [59] creation, the classification step is done by a Multi-Layer Perceptron network. The fusion of the KB is the goal of this approach. In the step of knowledge extraction of this method, extracted knowledge can be classified by the MLP network. In this method, the KB correction is as pre-correction. Thus, when some new facts were entered the KB, this approach can be used. Therefore, these approaches are not appropriate for the correction of existing errors of erroneous relations in a KB.

SDValidate [32] is another method of erroneous relations correction that has three parts: In the first part, RPF (relative predicate frequency) values for each triple are obtained, then lower RPF triples are removed. The repetition of two parts of these triples are less and their accuracy probability is low. In the next part, each entity is converted into a feature vector by a distribution from *Properties* and *Types* feature. Therefore, the relation score of each triple is gained by *cosine similarity* in two entities vectors in this triple. In the third part, a threshold value τ_i is considered for the classification of false and true facts for each relation r_i. Thus using the RPF idea, SDValidate presents a method for the correction of existing erroneous relations in a KB. The vector of each relation r_m of triple (s_i,r_m,o_j) is presented as ϕ_{ijm}^{SDV} for SDValidate. In equation (1), the score function is presented for the relation r_m. This notations are defined in [64] that the weight vector is w_m^T.

$$f_{ijm}^{SDV} \coloneqq w_m^T \phi_{ijm}^{SDV} \tag{1}$$

Some approaches such as [23, 65-67] use path ranking (PRA) algorithms. In PRA-based approaches, paths of triple entities are utilized instead of Properties and Type features. Since the Type relation of many KBs is not rich, the related studies do not work them. By the PRA approach, this problem is resolved. The PRA feature vector is presented as ϕ_{ijk}^{PRA} for each r_m in (s_i,r_m,o_j). In equation (2), the score function of PRA is presented for the relation r_m.

$$f_{ijm}^{PRA} \coloneqq w_m^T \phi_{ijm}^{PRA} \tag{2}$$

PaTyBRED [31] was introduced by Melo et al. to improve the related studies. The objective of PaTyBRED is KB correction of erroneous relations. PaTyBRED combines two methods of SDValidate and PRA. Thus, features of Paths and Types must be used for the creation of the vector $\phi_{ijm}^{PaTyBRED}$. Although in most KBs, there is not Type feature. On the other hand, some errors maybe include false entities by the true feature of the type that is not able to detect using Type-feature-based approaches [68]. Thus, PaTyBRED uses Paths and Types features by the combination of PRA and SDValidate helping a classifier for each r_m, that its score function is displayed in (3).

$$f_{ijm}^{PaTyBRED} \coloneqq w_m^{(1)T} \phi_{ijm}^{SDV} + w_m^{(2)T} \phi_{ijm}^{PRA} \tag{3}$$

Most related studies pay only to error recognition and error deletion, but a few approaches such as CoCKG [33] studies on repairing of the errors. CoCKG repairs errors that caused by entity confusion. In this method, an entity of

Table5 . Accuracy of Outlier Correction Methods

Method	FB15K	WN18
Fleischhacker [25]	0.23	0.21
Wienand [26]	0.22	0.25
Paulheim [27]	0.50	0.52
Huiying [28]	0.29	0.37
OTC	0.61	0.68
OEC	0.67	0.75

Table 6. Accuracy of Inconsistency Correction Methods

Method	FB15K	WN18
EPCI	0.83	0.85
PaTyBRED	0.62	0.66
IncDect	0.27	0.35

Table 7. Accuracy of Erroneous Relation Correction Methods

Method	FB15K	WN18
PaTyBRED	0.48	0.57
CoCKG	0.48	0.57
PaTyMLP	0.53	0.60
PRA	0.45	0.54
SDValidate	0.33	0.40

Table 8. Different Configurations of the GKBC for each Objective

Objectives	**#Conf.**	**First step**	**Second step**	**Third step**	**Accuracy**	
					FB15K	**WN18**
Non Numerical Correction/ Embedding Correction/ Erroneous Triples Correction	1	Paulheim [27]	PaTyBRED	PRA	0.55	0.65
	2	Paulheim [27] & Wienand [26]	EPCI	PaTyBRED	0.58	0.66
	3	**OEC &OTC**	**EPCI**	**PaTyMLP**	**0.66**	**0.73**
Numerical Correction	4	Wienand [26]	PaTyBRED	PaTyBRED	0.31	0.34
	5	Huiying [28]	PaTyBRED	PaTyMLP	0.35	0.36
	6	Huiying [28] & Wienand [26]	IncDect	PaTyMLP	**0.40**	**0.42**
Non Embedding Correction/ Pre-correction/ Error Avoidance	7	Fleischhacker [25]	Big RDF Consistency	Type Assertions	0.45	0.49
	8	Huiying [28] & Wienand [26]	Big RDF Consistency	PROSPERIA	0.48	0.51
	9	Huiying [28] & Wienand [26]	Big RDF Consistency	Type Assertions	**0.57**	**0.61**
Erroneous Interlinks Correction	10	Huiying et al. [28]	PaTyBRED	PRA	0.52	0.61
	11	Paulheim [27]	PaTyBRED	PRA	0.55	0.61
	12	Paulheim [27]	EPCI	SDValidate	**0.60**	**0.64**
Wrong Types Correction	16	Paulheim [27]	EPCI	PRA	0.55	0.66
	17	Wienand [26]	EPCI	PRA	0.51	0.63
	18	OEC &OTC	EPCI	Type Assertions	**0.62**	**0.67**
Error Repairing	19	Paulheim [27]	PaTyBRED	CoCKG	0.63	0.71
	20	Wienand [26]	PaTyBRED	CoCKG	0.60	0.69
	21	OEC &OTC	EPCI	PaTyMLP	**0.66**	**0.73**

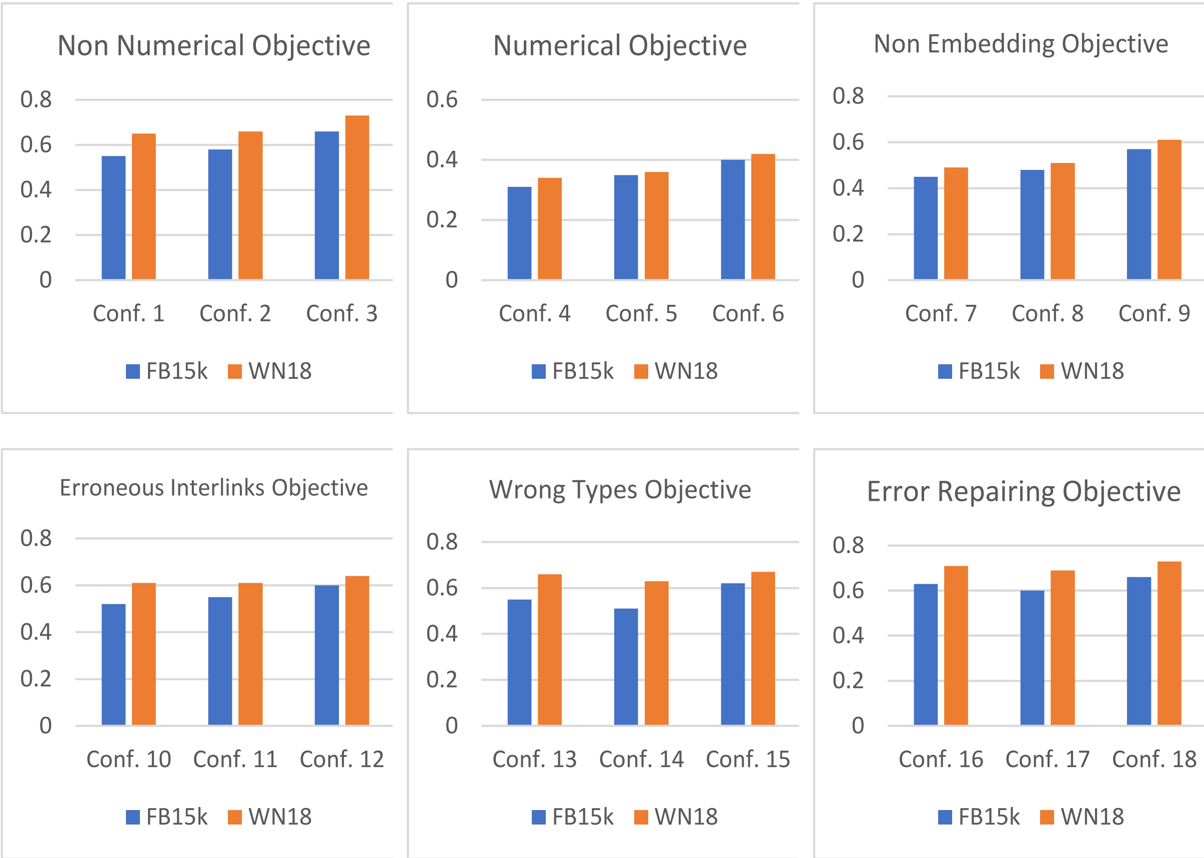

Fig. 8. Comparisons of Different Configurations of the GKBC for each Objective

the triple is changed with another entity for the creation of erroneous relation in that triple. The first step in PaTyBRED detects erroneous relations and then they must be repaired as bellow. In the erroneous triple, a new entity replaces instead of an entity of the triple. Thus, newly changed triples are produced for each erroneous triple that each of them gives a score by the score function. In the next step, the triple with maximum score is selected. If an erroneous triple is repairable, it is replaced by the triple with maximum score. In general, this approach has not acceptable accuracy for repairing the erroneous relations, but its results are better than existing repairing approaches in the field of entity confusion [33]. The score function of CoCKG is identical to PaTyBRED that was displayed in (3). KB embedding methods such as [4, 18-24, 50, 67, 69-74] can help to erroneous relations correction, but these approaches have rarely been used for this purpose, so far [31].

The last reviewed methods were embedding-based error correction, but non-embedding approaches correct these errors by reasoning methods using a rule-based mechanism. In fact, reasoning methods are not proper for real-world KBs [36]. For instance, reasoning on the DBpedia KB have not good results in erroneous relations correction because its rule-based elements are not rich. The method of Type Assertions [60] is a non-embedding approach that can detect wrong types of the KB and in the field of non-embedding approaches have good results [60], but its use in real-world applications is low because its entity types are considered as true relations, by default in the KB [36]. PROSPERIA [44] is another non-embedding approach that reasons KB for avoidance from erroneous relations. This method is done as a pre-correction in the step of knowledge extraction. Thus, it is not appropriate to correct existing errors of the KB.

7- EVALUATION

In this section, different methods of each step in the proposed GKBC is evaluated separately, and then the best configurations of them are proposed for different objectives. The evaluations are performed on the benchmark datasets of FB15k [75] and WN18 [75]. In Table 5, the accuracy of outlier correction methods is shown. Each of these methods can be used for an especial objective. Also, Table6 ,shows the accuracy of prominent methods of the inconsistency correction and in Table 7., the methods of erroneous relation correction are compared.

After evaluations of correction methods, we can find the most appropriate methods in each objective and new configurations can be suggested. To correct the KB errors, a configuration is proposed for each objective based on the proposed GKBC. These configurations are presented in Table

8. Each configuration is presented for a special objective in which a suitable method is proposed for each correction step. The method of each step is selected based on the evaluation results in the desired objective. Finally, a suitable configuration is selected for each objective based on its accuracy value.

Different configurations of each objective are compared in Fig. 8. For non-numerical objective, configuration 3 is selected. In this objective, OEC [9] has suitable results in outlier entities detection, and for this reason, OTC [9] is selected for non-numerical outlier triples detection. The evaluation results show that this method is suitable for this objective. For the inconsistency correction step, a non-numerical method did not exist, but the method of EPCI [30] was recently proposed by authors that can be used for this goal. The evaluation results show that this method can improve the detection and repair results. In the step of erroneous relations correction, the method of PaTyMLP [9] is selected. In this configuration, the methods of all steps are embedding type. Therefore, these methods can be also used for the objectives of the embedding correction and Erroneous Triples Correction.

The next objective is the numerical correction. Configuration 6 is selected for this objective. In the outlier correction step of this configuration, the methods of [28] and [26] are selected to correct numerical outliers because based on the evaluation results. The approach of [28] is a good way in numerical outlier entities detection and the approach of [26] is a suitable method in numerical outlier triples detection. On the other hand, the evaluations showed that IncDect [35] is more useful in numerical inconsistency correction. Therefore, this method is selected for the second step of this configuration. In the step of erroneous relations correction, PaTyBRED is suggested. The evaluation results show that this method outperforms other methods in numerical and non-numerical data.

Configuration 9 is proposed for the objective of the non-embedding correction. The methods of [28] and [26] can be used for the detection of non-embedding outliers. Among non-embedding approaches in the inconsistency correction methods, the Big RDF Consistency is suitable which is a rule-based method [40]. Thus, this method is selected for the step of inconsistency correction. Also, the method of Type Assertions [60] is used for the third step. Non-embedding methods use mostly the reasoning for detecting the erroneous relations in which Type Assertions is suitable for this objective. Existing pre-correction methods are mostly non-embedding approaches. Therefore, its steps are the same as previously. Also, the goal of pre-correction methods is error avoidance. Thus, the configuration of this objective is the same as this configuration.

Another objective is erroneous interlinks correction in which configuration 12 is selected. The approach of [27] is suitable for the correction of outlier interlinks between KBs. Thus, this method is selected for the first step of this configuration. In this objective, there is not a method that investigates inconsistency correction especially, but the EPCI is able to be generalized for this aim. For this reason, this method is proposed for the second step of this configuration. Also, SDValidate is a state-of-the-art erroneous relations correction method for this objective [32] that is selected for the third step. To evaluate SDValidate and Type-based methods, the Type properties of YAGO KB are added into the datasets.

In the objective of Wrong Types Correction, configuration 15 is selected. This configuration is the same as previous, but its first step has the additional method for entity errors. To correct the outliers in this configuration, the OEC and OTC [9] are used and in the third step, Type Assertions is selected. In this step, the method of Type Assertions can have better results for this objective [60]. In the next objective, the repairing is mandatory and the CoCKG [33] is used in comparison to PaTyMLP that the results showed the advantage of PaTyMLP for the third step. Other steps of this objective are the same as previously.

These configurations were proposed based on the evaluation results of the previous section and the results of related studies, but maybe exist other configurations that have similar results. Nevertheless, the proposed configurations are useful for the objectives that help researchers to utilize them for different applications, but there are some problems in these configurations that are investigated bellow. For example, there are no methods to correct both outlier entities and outlier triples. This weakness can be solved in future work. Also, previous methods of erroneous relation correction do not use KB embedding methods such as [63, 76, 77]. While KB embedding methods are very strong, they can be combined with existing methods to increase the accuracy of the correction method. Also, these methods can be optimized by methods such as [78-80].

8- CONCLUSION

In this paper, a new general configuration for the KB correction was proposed that is able to cover various errors and is called GKBC. In the GKBC, a classification of the errors was presented. This classification has three classes of the inconsistent triples, outliers, and erroneous relations. In this general configuration, the correction of each class was performed in a separate step. In related studies, the correction of the inconsistencies has not been considered. Although some of the inconsistencies are corrected by outlier and erroneous relations correction methods, many inconsistency errors cannot be detected by these methods. Thus, the correction of inconsistencies was considered in the proposed GKBC as a separate step. In the GKBC, the suitable approach of each step was identified for each objective and a different configuration of these methods was proposed for various objectives.

In future studies, the methods of KB embedding can be combined with existing methods of erroneous relations correction to increase the accuracy of the correction method. Another future work will be a new method to correct both outlier entities and outlier triples.

REFERENCES

1. Bollacker, K., et al. *Freebase: a collaboratively created graph database for structuring human knowledge*. in

ACM SIGMOD international conference on Management of data. 2008. ACM.
2. Miller, G.A., *WordNet: A Lexical Database for English*, in *Communications of the ACM*. 1995.
3. Suchanek, F.M., *Automated construction and growth of a large ontology*. 2008, Ph.D. thesis, Saarbrücken University.
4. Wang, Q., Z. Mao, and B. Wang, *Knowledge Graph Embedding: A Survey of Approaches and Applications.* IEEE Transactions on Knowledge and Data Engineering, 2017(99).
5. Petrova, A., et al. *Entity Comparison in RDF Graphs*. 2017. Cham: Springer International Publishing.
6. Kim, H., *Building a K-Pop knowledge graph using an entertainment ontology.* Knowledge Management Research & Practice, 2017. **15**(2): p. 305-315.
7. Paulheim, H. *Machine learning with and for semantic web knowledge graphs*. in *Reasoning Web International Summer School*. 2018. Springer.
8. Ristoski, P. and H. Paulheim, *Semantic Web in data mining and knowledge discovery: A comprehensive survey.* Journal of Web Semantics, 2016. **36**: p. 1-22.
9. Abedini, F., M.R. Keyvanpour, and M.B. Menhaj, *Correction Tower: A General Embedding Method of the Error Recognition for the Knowledge Graph Correction.* International Journal of Pattern Recognition and Artificial Intelligence, 2020: p. 2059034.
10. Suchanek, F.M., M. Sozio, and G. Weikum, *SOFIE: a self-organizing framework for information extraction*, in *Proceedings of the 18th international conference on World wide web*. 2009, ACM: Madrid, Spain. p. 631-640.
11. Weikum, G., Johannes Hoffart, and Fabian Suchanek., *Ten Years of Knowledge Harvesting: Lessons and Challenges.* Data Engineering 2016.
12. Preda, N., et al., *Active knowledge: dynamically enriching RDF knowledge bases by web services*. Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. 2010, Indianapolis, Indiana, USA: ACM. 399-410.
13. Abedini, F., F. Mahmoudi, and A.H. Jadidinejad. *A New Disambiguation Method for Semantic Entity Extraction Using YAGO Ontology*. in *In proceedings of IEEE 3th International Conference on Machine Learning and Computing (ICMLC 2011)*. 2011. Singapore.
14. Abedini, F., F. Mahmoudi, and A.H. Jadidinejad, *From Text to Knowledge: Semantic Entity Extraction using YAGO Ontology.* International Journal of Machine Learning and Computing, 2011. **1**(2): p. 113-119.
15. Abedini, F., F. Mahmoudi, and S.M. Mirhashem, *Using Semantic Entity Extraction Method for a New Application.* International Journal of Machine Learning and Computing, 2012. **2**(2).
16. Abedini, F. and M. Mirhashem, *SESR: Semantic Entity Extraction for Computing Semantic Relatedness*, in *International Conference on Advanced Computer Theory and Engineering, 4th (ICACTE 2011)*. 2011, ASME Press: Dubai.
17. Tickoo, O. and R. Iyer, *Knowledge and Ontologies*, in *Making Sense of Sensors: End-to-End Algorithms and Infrastructure Design from Wearable-Devices to Data Center*. 2017, Apress: Berkeley, CA. p. 83-94.
18. Balazevic, I., C. Allen, and T.M. Hospedales, *Hypernetwork Knowledge Graph Embeddings.* arXiv preprint arXiv:1808.07018, 2018.
19. Guan, N., D. Song, and L. Liao, *Knowledge graph embedding with concepts.* Knowledge-Based Systems, 2018.
20. Wang, K., et al., *Knowledge Graph Embedding with Entity Neighbors and Deep Memory Network.* arXiv preprint arXiv:1808.03752, 2018.
21. Zhu, J.-Z., et al., *Modeling the Correlations of Relations for Knowledge Graph Embedding.* Journal of Computer Science and Technology, 2018. **33**(2): p. 323-334.
22. Zhu, Q., et al., *A neural translating general hyperplane for knowledge graph embedding.* Journal of Computational Science, 2019. **30**: p. 108-117.
23. Lin, X., et al., *Relation path embedding in knowledge graphs.* Neural Computing and Applications, 2018.
24. Gao, H., et al., *Triple Context-based Knowledge Graph Embedding.* IEEE Access, 2018: p. 1-1.
25. Fleischhacker, D., et al., *Detecting Errors in Numerical Linked Data Using Cross-Checked Outlier Detection*, in *The Semantic Web – ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, P. Mika, et al., Editors. 2014, Springer International Publishing: Cham. p. 357-372.
26. Wienand, D. and H. Paulheim, *Detecting Incorrect Numerical Data in DBpedia*, in *The Semantic Web: Trends and Challenges: 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, V. Presutti, et al., Editors. 2014, Springer International Publishing: Cham. p. 504-518.
27. Paulheim, H., *Identifying Wrong Links between Datasets by Multi-dimensional Outlier Detection.* WoDOOM, 2014: p. 27-38.
28. Li, H., et al., *Probabilistic Error Detecting in Numerical Linked Data*, in *Database and Expert Systems Applications: 26th International Conference, DEXA 2015, Valencia, Spain, September 1-4, 2015, Proceedings, Part I*, Q. Chen, et al., Editors. 2015, Springer International Publishing: Cham. p. 61-75.
29. Aggarwal, C.C., *Outlier Detection in Graphs and Networks*, in *Outlier Analysis*. 2017, Springer International Publishing: Cham. p. 369-397.
30. Abedini, F., M.B. Menhaj, and M.R. Keyvanpour, *EPCI: An Embedding method for Post-Correction of Inconsistency in the RDF Knowledge Bases.* IETE Journal of Research, 2019.
31. Melo, A. and H. Paulheim, *Detection of Relation Assertion Errors in Knowledge Graphs*, in *Proceedings of the Knowledge Capture Conference*. 2017, ACM: Austin, TX, USA. p. 1-8.
32. Heiko, P. and B. Christian, *Improving the Quality of*

Linked Data Using Statistical Distributions. International Journal on Semantic Web and Information Systems (IJSWIS), 2014. **10**(2): p. 63-86.
33. Melo, A. and H. Paulheim, *An Approach to Correction of Erroneous Links in Knowledge Graphs*, in *1st International Workshop on Quality Engineering Meets Knowledge Graph*. 2017, ACM: Austin, TX. p. 1-4.
34. Wang, C., et al., *Error Link Detection and Correction in Wikipedia*, in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 2016, ACM: Indianapolis, Indiana, USA. p. 307-316.
35. Fan, W., et al. *Catching Numeric Inconsistencies in Graphs*. in *Proceedings of the 2018 International Conference on Management of Data*. 2018. ACM.
36. Paulheim, H., *Knowledge graph refinement: A survey of approaches and evaluation methods.* Semantic Web Preprint, 2016: p. 1-20.
37. Parzen, E., *On estimation of a probability density function and mode.* The annals of mathematical statistics, 1962. **33**(3): p. 1065-1076.
38. Fleischhacker, D., *Detecting errors in linked data using ontology learning and outlier detection*. 2015.
39. Breunig, M.M., et al., *LOF: identifying density-based local outliers.* SIGMOD Rec., 2000. **29**(2): p. 93-104.
40. Benbernou, S. and M. Ouziri. *Enhancing data quality by cleaning inconsistent big RDF data*. in *2017 IEEE International Conference on Big Data (Big Data)*. 2017.
41. Töpper, G., Magnus Knuth, and Harald Sack. *DBpedia ontology enrichment for inconsistency detection*. in *8th International Conference on Semantic Systems*. 2012. ACM.
42. Fang, J. and Z. Huang, *Reasoning with inconsistent ontologies.* Tsinghua Science & Technology, 2010. **15**(6): p. 687-691.
43. Dylla, M., Mauro Sozio, and Martin Theobald, *Resolving Temporal Conflicts in Inconsistent RDF Knowledge Bases.* BTW, 2011.
44. Nakashole, N., M. Theobald, and G. Weikum, *Scalable knowledge harvesting with high precision and high recall*, in *Proceedings of the fourth ACM international conference on Web search and data mining*. 2011, ACM: Hong Kong, China. p. 227-236.
45. Guoliang Ji, K.L., Shizhu He and Jun Zhao, *Knowledge Graph Completion with Adaptive Sparse Transfer Matrix.* Association for the Advancement of Artificial Intelligence, 2016.
46. Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. . *Knowledge Graph Embedding via Dynamic Mapping Matrix*. in *ACL*. 2015.
47. Wang, Z., et al., *Knowledge Graph Embedding by Translating on Hyperplanes.* AAAI, 2014.
48. Zhang, C., Zhou, M., Han, X., Hu, Z., & Ji, Y., *Knowledge graph embedding for hyper-relational data.* Tsinghua Science and Technology, 2017. **22**(2): p. 185-197.
49. Melo, A., J. Völker, and H. Paulheim, *Type Prediction in Noisy RDF Knowledge Bases Using Hierarchical Multilabel Classification with Graph and Latent Features.* International Journal on Artificial Intelligence Tools, 2017. **26**(02): p. 1760011.
50. Ristoski, P., and Heiko Paulheim, *Rdf2vec: Rdf graph embeddings for data mining*, in *International Semantic Web Conference*. 2016, Springer International Publishing.
51. Andr, et al., *Type Prediction in RDF Knowledge Bases Using Hierarchical Multilabel Classification*, in *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*. 2016, ACM: Nîmes, France. p. 1-10.
52. Zhang, D., *Learning through overcoming temporal inconsistencies*, in *14th International Conference on Cognitive Informatics & Cognitive Computing*. 2015, IEEE.
53. Zhang, D., and Meiliu Lu, *Inconsistency-induced learning for perpetual learners.* Advances in Abstract Intelligence and Soft Computing 2012.
54. Sheng, Z., et al. *Checking and handling inconsistency of DBpedia*. in *International Conference on Web Information Systems and Mining*. 2012. Springer Berlin Heidelberg.
55. Flesca, S., F. Furfaro, and F. Parisi, *Querying and repairing inconsistent numerical databases.* ACM Transactions on Database Systems (TODS), 2010. **35**(2): p. 14.
56. Suchanek, F.M., Gjergji Kasneci, and Gerhard Weikum. *Yago: a core of semantic knowledge*. in *Proceedings of the 16th international conference on World Wide Web.* . 2007. ACM.
57. Abedini, F. and S.M. Mirhashem, *Entity Disambiguation in Text by YAGO Ontology.* International Journal of Computer Theory and Engineering, 2013. **5**(3).
58. Bizer, C., et al. , *DBpedia-A crystallization point for the Web of Data.* Web Semantics: science, services and agents on the world wide web, 2009. **7**(3): p. 154-165.
59. Dong, X., et al. *Knowledge vault: A web-scale approach to probabilistic knowledge fusion*. in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014. ACM.
60. Ma, Y., et al., *Learning Disjointness Axioms With Association Rule Mining and Its Application to Inconsistency Detection of Linked Data*, in *The Semantic Web and Web Science: 8th Chinese Conference, CSWS 2014, Wuhan, China, August 8-12, 2014, Revised Selected Papers*, D. Zhao, et al., Editors. 2014, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 29-41.
61. Abedini, F. and M. Mirhashem, *From text to facts: Recognizing ontological facts for a new application.* International Journal of Machine Learning and Computing 2012. **2**(3).
62. Abedini, F., M.R. Keyvanpour, and M.B. Menhaj, *An RDF Based Fuzzy Ontology Using Neural Tensor Networks.* International Journal of Information & Communication Technology Research, 2019. **11**(1): p. 45-56.
63. Abedini, F., M.B. Menhaj, and M.R. Keyvanpour, *An MLP-based representation of neural tensor networks for the RDF data models.* Neural Computing and Applications, 2019. **31**(2): p. 1135-1144.

64. Nickel, M., et al., *A review of relational machine learning for knowledge graphs.* Proceedings of the IEEE, 2015. **104**(1).
65. Lao, N. and W.W. Cohen, *Relational retrieval using a combination of path-constrained random walks.* Machine Learning, 2010. **81**(1): p. 53-67.
66. Lin, X., et al., *A Knowledge Base Completion Model Based on Path Feature Learning.* International Journal of Computers, Communications & Control, 2018. **13**(1).
67. Zhang, M., et al. *Discriminative Path-Based Knowledge Graph Embedding for Precise Link Prediction*. 2018. Cham: Springer International Publishing.
68. Melo, A., *Automatic refinement of large-scale cross-domain knowledge graphs*. 2018, Ph.D thesis, Mannheim University.
69. Chang, L., et al., *Knowledge Graph Embedding by Dynamic Translation.* IEEE Access, 2017. **5**: p. 20898-20907.
70. Goyal, P. and E. Ferrara, *Graph Embedding Techniques, Applications, and Performance: A Survey.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
71. Han, X., et al., *A Triple-Branch Neural Network for Knowledge Graph Embedding.* IEEE Access, 2018. **6**: p. 76606-76615.
72. Kazemi, S.M. and D. Poole, *SimplE Embedding for Link Prediction in Knowledge Graphs.* CoRR, 2018. **abs/1802.04868**.
73. Kristiadi, A., et al., *Incorporating Literals into Knowledge Graph Embeddings.* arXiv preprint arXiv:1802.00934, 2018.
74. Wang, M., et al. *Embedding Knowledge Graphs Based on Transitivity and Asymmetry of Rules*. 2018. Cham: Springer International Publishing.
75. Bordes, A., et al, *Translating embeddings for modeling multi-relational data.* Advances in Neural Information Processing Systems, 2013.
76. Abedini, F., Menhaj, M. B., & Keyvanpour, M. R., *Neuron Mathematical Model Representation of Neural Tensor Network for RDF Knowledge Base Completion.* Journal of Computer & Robotics, 2017. **10**(1): p. 1-10.
77. Abedini, F., M.R. Keyvanpour, and M.B. Menhaj, *Neural Tensor Network Training Using Meta-Heuristic Algorithms for RDF Knowledge Bases Completion.* Applied Artificial Intelligence, 2019. **33**(7): p. 656-667.
78. Sohrabi, M.K. and H. Azgomi, *Parallel set similarity join on big data based on Locality-Sensitive Hashing.* Science of Computer Programming, 2017. **145**: p. 1-12.
79. Sohrabi, M.K. and H. Azgomi, *A Survey on the Combined Use of Optimization Methods and Game Theory.* Archives of Computational Methods in Engineering, 2020. **27**(1): p. 59-80.
80. Azgomi, H. and M.K. Sohrabi, *A novel coral reefs optimization algorithm for smaterialized view selection in data warehouse environments.* Applied Intelligence, 2019. **49**(11): p. 3965-3989.

HOW TO CITE THIS ARTICLE

F.Abedini, M. Keyvanpou, M. Menhaj New Configurations for the Correction of the RDF knowledge bases , AUT J. Model. Simul., 52(2) (2020) 269-282.

DOI: 10.22060/miscj.2020.18353.5207