IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 4 April 2023, accepted 2 June 2023, date of publication 7 June 2023, date of current version 19 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3283925

RESEARCH ARTICLE

Continuous Knowledge Graph Refinement With Confidence Propagation

ALISETTAR HUSEYNLI AND **M. ALI AKCAYOL**
Department of Computer Engineering, Faculty of Engineering, Graduate School of Natural and Applied Sciences, Gazi University, 06560 Ankara, Türkiye
Corresponding author: Alisettar Huseynli (alisettar@dolusoft.com)

This work was supported in part by the Scientific and Technological Research Institution of Turkey (TÜBİTAK) through the 2244 Industrial Ph.D. Program under Project 118C127.

ABSTRACT Although Knowledge Graphs (KGs) are widely used, they suffer from hosting false information. In the literature, many studies have been carried out to eliminate this deficiency. These studies correct triples, relations, relation types, and literal values or enrich the KG by generating new triples and relations. The proposed methods can be grouped as closed-world approaches that take into account the KG itself or open-world approaches using external resources. The recent studies also considered the confidence of triples in the refinement process. The confidence values calculated in these studies affect either the triple itself or the ground rule for rule-based models. In this study, a propagation approach based on the confidence of triples has been proposed for the refinement process. This method ensures that the effect of confidence spreads over the KG without being limited to a single triple. This makes the KG continuously more stable by strengthening strong relationships and eliminating weak ones. Another limitation of the existing studies is that they handle refinement as a one-time operation and do not give due importance to process performance. However, real-world KGs are live, dynamic, and constantly evolving systems. Therefore, the proposed approach should support continuous refinement. To measure this, experiments were carried out with varying data sizes and rates of false triples. The experiments have been performed using the FB15K, NELL, WN18, and YAGO3-10 datasets, which are commonly used in refinement studies. Despite the increase in data size and false information rate, an average accuracy of 90% and an average precision of 98% have been achieved across all datasets.

INDEX TERMS Continuous refinement, knowledge graph refinement, propagation, triple confidence.

I. INTRODUCTION

Knowledge bases refer to sets of knowledge formed by cases that include commonsense or specialized knowledge in a particular field. Studies on knowledge bases date back to the 1970s. The Cyc project [1], created by Lenat et al., can be shown as an example of the earliest knowledge base studies. With the "Google Knowledge Graph [2]", project announced by Google in 2012, knowledge bases were transformed into knowledge graphs (KGs) and gained popularity. KGs play an important role in creating infrastructure in many downstream applications, from artificial intelligence studies to decision support systems, from question answering applications to search engines [3], [4], [5], [6]. The accuracy of the information contained in KGs is just as essential as its importance [7].

Different refinement studies were conducted to ensure the accuracy of KG. In general, refinement studies can be categorized as completion and correction approaches [8], [9]. Completion studies are closely related to link prediction studies and aim to generate new relations in existing KGs with closed-world assumptions [10], [11], [12], [13]. Error correction studies, on the other hand, deal with correcting relations, ontological types, and literal values.

One of the new topics that have gained popularity in recent KG creation and refinement studies is the consideration of confidence values for relations. The confidence value depends on the relation of triple and have been used in KGs such as NELL [14] and in refinement studies based on

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale de Meo.

fact-checking [15]. Both from closed-world approaches (*will refer as offline*) rule-based models determine confidence values using t-norm or co-norm with probabilistic soft logic [16], [17], [18], or Markov Logic Network methods [19], while translation-based models use a cost function [20]. There are also studies with an open-world approach (*will refer as online*) for calculating confidence values over external sources [15], [21]. In these studies, this value refers to the reliability of the source [21], the accuracy of the information [14], and the combined value obtained by combining both values [15].

In this study, a new model based on confidence values and the connection of triples with each other is proposed for refining KG. This model focuses on the propagation of confidence values within KGs. These values can be generated through offline methods within the graphs themselves, or obtained through online methods from external sources. The model enables the propagation of these values throughout the graph, extending beyond the constraints of existing triples. The propagation ensures that the confidence values on the triangular structures in the graph are updated by increasing or decreasing them. In this way, it strengthens strong relations with high-confidence values and eliminates weak relations with low ones.

Refining KGs is not a one-time operation. It should be considered that it is necessary to work with a dynamic and ever-growing graph structure. Therefore, the size of the data and topology of the graph must be taken into account. Additionally, as emphasized in Paulheim's study [22], considering the refinement process without taking into account the performance criteria will cause errors when evaluating the accuracy of the proposed method. In this study, the performance criterion such as total spent time and scalability of refinement was taken into account, regardless of data size, as well as the success of clearing noisy triples, to ensure accurate evaluation. In this regard, the proposed method, with its innovative approach, does not process refinement as a one-time operation and provides continuous refinement while the KG is created and expanded.

The sections of the article are organized as follows. The next section will present current work related to error correction and noise removal. In the subsequent sections, the problem definition and challenges will be described, and the methodology and novelty of the proposed method for the correction process will be discussed. Then, the preparation of the experimental environment and the evaluation criteria will be described. In the experimental results section, the obtained results will be presented and discussed. Finally, the article will conclude by mentioning the future studies planned on this subject.

II. RELATED WORK

Studies on correctness and fact-checking in KGs are among the popular research topics [23]. In particular, considering that KGs have reached enormous sizes, automated verification models rather than manual methods have started to occur in the literature. In this paper, studies on error correction and noise removal in KGs will be evaluated under the closed-world assumption and open-world assumption categories.

A. CLOSED-WORLD ASSUMPTION METHODS

Closed-world assumption methods are based on the assumption of completeness of the KG [10]. These methods verify, correct errors, and remove noise by considering the existing data in the KG. Closed-world assumption methods can further be classified into translation-based and rule-based methods.

Translation-based methods, starting with TransE [20], express the relation in the triple as the translation between subject and object, as $head + relation \approx tail$. The work on TransE has been continued in different ways using various types of spatial and language transformations [24], [25], [26], [27], [28], [29], [30]. Although TransE offers easy implementation, it exhibits limited performance when modeling complex relations. PTransE [27] extended the TransE to encode complex relations using the path information in KG. However, PTransE is also limited in modeling different relationship patterns. PaTyBRED [31] combines information about the paths between entities and the types of entities and includes these properties in the classification. UKGE [32] includes the confidence value in its triple assessment along with the conversion value defined in the TransE. However, the limitations of TransE are also relevant for this study. The CKRL [33] extended the TransE differently by offering a framework based on confidence value. In this study, the confidence is not limited to the relations between entities. This value is calculated on both the local and global paths. In addition to CKRL, DSKRL [34] performs noise removal by adding hierarchical type information and dissimilarity information of entities. INDIGO [35] can be shown as another example of current error correction studies. Using Graph Neural Networks, INDIGO expresses the KG as GNN. Then, new triples are created using this neural network.

Rule-based information extraction methods are preferred for information embedding and new information extraction. These methods are also used to verify existing information based on logic rules. Rule-based error correction studies make extractions with logic rules or detect erroneous information with the help of predefined rules as *(Y, sonOf, X)* $\wedge$ *(X, hasChild, Y)* $\wedge$ *(Y, gender, Male)*. For example, the "Correction Tower" [36] offers easy implementation and provides success in detecting outliers and erroneous relations. In contrast, this study failed in large KG structures. The KALE [17] also extracts new information by calculating logic operations such as union, intersection, and inverse with t-norm fuzzy logic rules. Another study by Jeyaraj et al. [18] calculates the value of the rules according to the probability distribution using probabilistic soft logic framework. pLogicNet [19] prefers the Markov logic network for defining rules and calculating new values. Similarly, studies prefer a Bayesian network as well [37].

The PTrustE [38] presents a different approach by combining rule-based and translation-based studies in a single model. To reveal the features of KG, it first embeds the path information of the entities and then calculates global and local confidence values with the help of the probability logic network. Finally, it creates a path confidence value with the help of Bi-directional Gated Recurrent Unit and uses this value to determine the trustworthiness of the triple.

B. OPEN-WORLD ASSUMPTION METHODS

The open-world approach is based on the assumption of incompleteness in KGs [39]. Therefore, the verification of triples and detection of errors in these studies are not limited to the internal resources of the KG. The accuracy of triples is verified by outsourcing web resources, social media, and other KGs. When extracting information using the web resources, NELL [14] assigns confidence values to candidate relations with heuristic methods. The Knowledge Vault [21], which also uses web resources, calculates the confidence value by using data fusion from the sources with data extraction methods.

The source of evidence, and methods to be used for verification are important parameters in the verification process. In his work [40] dealing with these concepts, Wang considers metadata about the source, such as by whom and through which media the information is provided for the verification process. Although this method is not sufficient to confirm the information, it provides additional information to increase classification success.

Another study "Tracy" [41], that performs fact-checking using existing KGs, is conducted by Mohamed. In this study, the fact to be verified is broken down, and facts related to the entities in the KGs are collected, and logical inferences are made on the basis of these facts. Gerber et al. proposed another validation approach based on web search engines called the "DeFacto - Deep Fact Validation" [15]. In this study, the query obtained is searched on a web search engine, and the resulting pages are evaluated based on their relevance and confidence level. The results are then presented to the user for decision-making.

The verification of information is also performed using natural language processing methods and evidence on texts, along with confidence values obtained from the web and KG resources. For example, in the "FactCheck" [42], study by Zafar et al., validation is performed on RDF triples with text-based operations. In another example with text processing methods, combining logical and text-based evidence was performed by Du et al. [43]. Along with the methods mentioned for the verification process, other approaches deal with data mining, and natural language processing [44].

Recently, detailed surveys on the refinement process in KGs have also been carried out. For more detailed research on the field, Paulheim's survey [22], Hogan's "Knowledge Graphs" study [8], and Wang's work [10] can be examined.

In the proposed method, logical inference techniques suggested by rule-based models and confidence-aware KG refinement methods are combined. Confidence values obtained from external sources or from the KG itself can propagate on the graph with rule-based inferences, and they can affect other triples. This effect aims to strengthen strong relations and eliminate weak ones by weakening them. The novelty revealed by the study is that the error correction and noise removal process is easy to implement and continuous. Considering the theoretical effectiveness of the propagation effect, it is predicted that the proposed method will also yield successful results in real-world KGs.

III. PROBLEM DEFINITION

It has become imperative to correct errors and eliminate noisy triples in automatically generated and continuously updated KGs. Errors can be defined as establishing relations between unrelated entities, inaccuracies in literal values, or adding false triples to KG. These errors can negatively affect downstream applications such as intelligent question answering systems and semantic search engines. While studies using an offline approach have revealed successful methods for eliminating errors, these models need to be run periodically for the entire KG. Specifically, representation-based models that reveal the attributes in the graph structure of KG, translation-based models that map the semantic connection between entities and relations, or rule-based models that provide the determination of the logical rules of triples for refinement deal with the entire KG. As expected, these methods lead to complex and computationally expensive operations in large KG structures.

Online methods can calculate the confidence value based on the reliability of the source when extracting triples from each piece of information. In this approach, since each triple is processed individually, there is no scaling problem as in offline models. However, these models are also lacking in reducing noise and eliminating errors. The confidence of the newly added or updated triple is limited only to this triple and has no effect on cleaning the entire KG.

The confidence value is assumed as predefined for triples in the proposed method. As mentioned before, the confidence can be obtained from external sources that are structured like other KGs or semi-structured, like web resources. This value can also be obtained from the KG through offline methods. The confidence value obtained is not limited to the current triple, but also contains meaningful information for other triples in KG. To better express this situation, the sample graph structure shown in Fig. 1 can be examined.

Example 1. Suppose that the confidence value of *"Albert Einstein - WinnerOf - Nobel Prize in Physics"* triple is updated. Since the *"Nobel Prize in Physics"* node has a relationship with the *"Physics"* node through the *"AwardIn"* relation, it is meaningful that the updated confidence value also affects the confidence value of *"Albert Einstein - ExpertIn - Physics"* triple.

The phenomenon illustrated in Example 1 can be referred to as the *"propagation effect of confidence"*, as it will persist across all triples connected on the KG. The main idea in

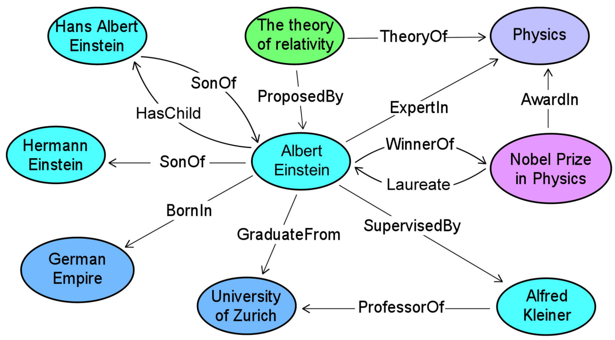

FIGURE 1. Example KG diagram.

creating the propagation effect is to stabilize KG in terms of reliability by creating an effect on both the obtained triple and the other associated triples.

Another feature that stands out in rule-based methods is the assumption that all relationships in the triple also have an inverse one [19]. This feature can be explained with the following examples on the graph structure in Fig. 1.

Example 2. The following examples show the validity of the inverse relation for both pairs of triples.

- *(Albert Einstein, hasChild, Hans Albert Einstein)* → *(Hans Albert Einstein, sonOf, Albert Einstein)*
- *(Albert Einstein, winnerOf, Nobel Prize)* → *(Nobel Prize, laureate, Albert Einstein)*

This property is generalized for the set of entities E as $\forall x, y \in E, \boldsymbol{v}(x, r_i, y) \Rightarrow \boldsymbol{v}(y, r_j, x)$ where r_i and r_j are inverse relations, allowing the KG to be processed as an undirected graph.

IV. PROPOSED METHOD

To continuously refine KGs, this paper recommends distributing confidence values across associated triples. This can be achieved by propagating values over the relationships defined in Example 1. The confidence values are predefined for triples and are added to the KG regardless of their individual confidence levels. As more triples are added, related weak links are identified and removed, leading to a more reliable and trustworthy KG. This process also strengthens the system for strong confidence values. To accurately express the propagation effect, the parameters shown below must be defined:

1. *Propagation rules:* define the rules under which the confidence value will spread over KG.

2. *Propagation calculation*: reveals how the propagation effect will change existing confidence values.

3. *Propagation width*: Determines how wide and by what method the propagation effect on KG will be limited.

Some preliminary information will be presented to define these parameters. This information discusses propagation rules, suggested approaches for recalculating confidence values, and the characteristics of propagation.

A. PROPAGATION RULES

Inference rules are logical rules that are used to extract information from a knowledge graph [19]. These rules are determined in rule-based information extraction models and can be applied to the graph to determine the propagation rules of the confidence value. Essentially, the propagation rules define how the confidence value will spread over the associated triples, and they are based on the logical connections and relationships between the entities and attributes in the graph.

- *Composition rule*: For any three x, y, z entities, r_k relation is a combination of r_i, and r_j relations. $\forall x, y, z \in E, \boldsymbol{v}(x, r_i, y) \wedge \boldsymbol{v}(y, r_j, z) \Rightarrow \boldsymbol{v}(x, r_k, z)$
- *Inverse rule*: If there is a r_i relation between x and y entities, there is also r_j relation between y and x. $\forall x, y \in E, \boldsymbol{v}(x, r_i, y) \Rightarrow \boldsymbol{v}(y, r_j, x)$
- *Symmetry rule*: If the relation r_i is the same between any two entities x and y, and the relation r_j between y and x, there is relation symmetry between these entities. $\forall x, y \in E, \boldsymbol{v}(x, r, y) \Rightarrow \boldsymbol{v}(y, r, x)$
- *Sub-relation rule*: If there are two different relations r_i, and r_j between any two entities x and y, the relations r_i, and r_j are sub-related. $\forall x, y \in E, \boldsymbol{v}(x, r_i, y) \Rightarrow \boldsymbol{v}(x, r_j, y)$

As it is clearly seen, there is no spread of confidence value for other rules except the composition rule. In other cases, only the value of the existing triple can be updated with the new value. In the composition rule, it is possible to spread the value over the triangle structure described in Example 1.

B. CALCULATING THE CONFIDENCE VALUE

The propagation calculation defines a method for calculating the new confidence values for the composition rule. This method is also used to calculate the updated confidence values in the case of updating the existing triple.

The proposed model considers both the updating and spreading of confidence values. Therefore, it determines how the confidence value calculated by offline approaches or obtained by online approaches will update the confidences in KG. This calculation will also be applied to confidence values updated during propagation. In this sense, the calculation of confidence values shares some common aspects with existing studies. However, calculating new confidence values requires a different approach than calculating truth values. Although the confidence is related to the truth of the information, it is clear that these two values have different meanings due to the nature of the information.

To update the new confidence value, the result can be determined by combining the confidence value of the triple in KG and the newly obtained confidence value. Here, reference can be made to the change of confidence (x) in a particular fact against newly obtained confidence (y) regarding this fact from an external source. If the old confidence is high and the newly obtained value is also high, our final confidence in the current fact increases rapidly. If the old confidence is high and the new confidence is low, the final confidence decreases slowly. In the opposite case, if the old confidence is low and

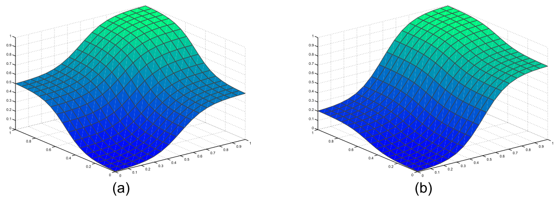

FIGURE 2. Logistic function with equal weights (a) and with weight multipliers (b).

the newly obtained value is high, the final confidence in the fact increases slowly. If both the old and the new confidences are low, the final confidence decreases rapidly. The threshold value (λ) can be used to define whether the confidence values are low or high.

This kind of relationship between input and output reflects the sigmoid function. Sigmoid functions tend to weaken low input values and strengthen high ones. This depends on whether the input is above or below the defined threshold. Logistic functions, inverse trigonometric functions, hyperbolic functions, etc. from the family of sigmoid functions are preferable. In this study, the logistic function in terms of transformation was preferred.

If it is assumed that the confidence values are in the range of 0, 1, the logistic function is expected to be in this range as well. As a result of the transformation, the desired output is obtained by narrowing the logistic function to $\{0, 1\}$ on the x and y-axis.

Another parameter in the creation of the sigmoid function is that the number of inputs is defined on two axes. This is forms the plane shown in Fig. 2(a).

The vertical parameters of the sigmoid function can be adjusted to ensure a stable system based on the predefined threshold value. Another important parameter that defines the output values is the weight multiplier for the old and new confidences used as inputs. The weight value can be adjusted by multiplying the effect of the current confidence on the output by the weight multiplier. An example of a logistic function with defined vertical parameters and weight multipliers is shown in Eq. 1.

$$f(x, y) = w_1 \cdot \frac{1}{1 + e^{-10x+5}} + w_2 \cdot \frac{1}{1 + e^{-10y+5}} \quad (1)$$

If $w_1 = 0.8$ and $w_2 = 0.2$ will be defined for the function in Eq. 1, the logistic function to be obtained will turn into the plane in Fig. 2(b).

C. PROPAGATION PROCESS

The propagation of confidence values in KG is initiated by detecting the triangles defined by the adjacent and opposite neighbors of the triple whose confidence is updated. The triangle structure is defined as $(h, r_1, t) \rightarrow (t, r_2, x) \rightarrow (x, r_3, h)$ for the triple (h, r_1, t), where (h, r_1, t) is the last updated triple, (x, r_3, h) is *adjacent* because it shares the common head entity with the updated triple, and (t, r_2, x) is *opposite* because it is located across the head entity h and does not share any common entity with it. The updating process of calculated confidences for those triples continues until the adjacent and opposite triples of each updated one in the queue are finished. Relations with confidence values below the defined threshold are deleted from KG. The pseudocode for the propagation is shown in Algorithm 1 as the *"PropagateConfidence"* and *"ChangeConfidence"* functions.

Algorithm 1 Propagation Algorithm

```
Triple[] visitedTriples
Queue tripleQueue
Graph G
procedure PropagateConfidence(triple)
    enqueue triple to tripleQueue
    add triple to visitedTriples
    while tripleQueue ≠ ∅ do
        t ← dequeue tripleQueue
        newConf ← confidence of t
        triangles ← triangles for triple t
        if triangles ≠ ∅ then
            for all triangle ∈ triangles do
                adjTriple ←adjacent Triple of triangle
                opTriple ←opposite Triple of triangle
                ChangeConfidence(adjTriple, newConf)
                ChangeConfidence(opTriple, newConf)
            end for
        end if
    end while
end procedure

procedure ChangeConfidence(triple, newConf)
    if triple ∉ visitedTriples then
        calculatedConf ← calculate with logistic function using newConf
        if calculatedConf <= threshold then
            remove triple from G
        else
            confidence of triple ← calculatedConf
            enqueue triple to tripleQueue
            add triple to visitedTriples
        end if
    end if
end procedure
```

The propagation characteristics of confidence values over KG can be determined by different approaches. These approaches will be examined under the headings of efficiency, scale, and simultaneity of propagation.

1) PROPAGATION EFFICIENCY

This feature determines the effect of the new confidence value on the triples as it spreads across the KG. The confidence value can either maintain its original value or decrease as it moves away from its starting point. Maintaining a constant confidence value leads to stronger changes in the KG, whereas the attenuation of the confidence value decreases its effectiveness. This attenuation can be achieved by adjusting the weight multipliers in the sigmoid function.

2) PROPAGATION DIAMETER

The diameter of the propagation determines how many hops the confidence value will affect from the starting point. When this criterion is set to be unlimited, all relationships with a triangular connection will be affected. If the spreading diameter is defined, the diameter is determined as the number of hops from the initial triple, and the spreading process is stopped when the defined limit is reached. Each hop is defined as the transition of adjacency and opposite relations to other triangles without including the previous triangle.

3) PROPAGATION SIMULTANEITY

Another criterion that determines the spreading characteristic is based on whether the update operations are simultaneous. The asynchronous update operation processes the propagation of each updated relation as an atomic operation. In the case of simultaneous propagation, different relations on KG are updated synchronously. Theoretically, it is predicted that the simultaneous propagation process will produce different results for the confidence values in different relations of KG compared to the sequential process. Simultaneous updating may cause interference on confidence values, thus dampening or strengthening confidence values.

V. EXPERIMENTS

In this section, the experimental design for the propagation process and the datasets used in the experiments will be discussed. The results obtained by examining the experimental study will be presented and the proposed model will be evaluated based on these results.

A. DATASETS

In this study, four different datasets that are popularly preferred in KG studies were used. The purpose of using different datasets is to evaluate the results of the propagation process not only according to the number of corrupted triples and dataset size but also according to the graph topology. In dataset selection, real-world large dataset such as NELL were also preferred, along with datasets that are actively used in translation-based and rule-based embedding studies.

The first chosen set to update the confidences and test the propagation process is the FB15K dataset, subset of Freebase, which was previously used in KG studies. FB15K contains approximately 15,000 entities. The other dataset, WN18, was created as a subset of the WordNet dataset, and the training set consists of 141,442 triples. These triples have 18 different types of relations. WN18 and FB15K datasets were developed by Bordes et al. [20]. The third dataset, NELL [14], has been under development since 2010. It also contains confidence values for triples. In the NELL dataset, there are

TABLE 1. Datasets used for confidence propagation.

Datasets	Relations	Entities	Triples
WN18	18	40,943	14,442
FB15K	1,345	14,951	483,142
YAGO3-10	37	123,182	1,079,040
NELL	-	-	2,810,379

candidate and high-confidence triples. The experiments were performed on a high-confidence dataset. The last dataset that was used, YAGO3-10 [45], was prepared as a subset of the YAGO3 dataset consisting of 123,182 entities and 37 relations. Statistics on datasets are shown in Table 1.

In all preferred datasets, object-subject relations with the same predicate have been disposed of so that directional links do not cause repetitive relationships.

B. GENERATING CORRUPTED TRIPLES AND CONFIDENCE VALUES

To conduct experimental studies, corrupted triples need to be added to the validation set. The generation of corrupted triples was inspired by the approach of DSKRL [34] and PTrustE [38] and generalized to other datasets. The following steps were followed to create these corrupted triples: two triples (h, r_1, t) and (t, r_2, s) were randomly selected from the dataset, where $h \neq s$ and $r_1 \neq r_2$. A corrupted triple (h, r, s) was formed by randomly selecting the relation r from the relation set $R' = \{r \mid r \neq r_1 \cap r \neq r_2\}$. The created triple was then randomly added to the set, disregarding the order of the dataset. The dataset was enriched by applying the corrupted triple generation rate of 20%, 40%, and 60% of the validation set for different experimental designs. The aim of varying the rates of corrupted triples was to evaluate how well the propagation process performs as the rate of corrupted triples increases.

Except for the NELL dataset, other popular datasets do not contain any confidence values. Therefore, confidence values defined for triples were synthetically produced for use in experimental studies. To ensure an unbiased approach in synthetic data generation processes, the distribution of the data was made independent of the content of the triples. Normal distribution, exponential distribution, and randomly generated numbers between 0 and 1 were used to determine the confidence values. For the normal distribution, the standard deviations of *true* and *false* values were both defined as 0.1. The confidence value distribution averages of the true and false triples for the normal distribution were determined to be 0.7 and 0.4, respectively. In experiments aimed at deleting false values, the deletion threshold value was set at 0.3. A threshold value of 0.3 corresponds to the lowest tolerance threshold of false triples with a mean of 0.4 and a standard deviation of 0.1. To test that the proposed method is independent of the distribution of confidence values, similar experiments were repeated using the exponential distribution and randomly generated confidence values. These assumptions in synthetic values can be completely changed, or values in KG created with confidence values obtained from external sources can be used.

C. EXPERIMENT DESIGN

Experiments were conducted on subsets of FB15K, NELL, WN18, and YAGO3-10 datasets, which were separated from the training sets as 10K, 20K, 30K, and 40K. In this way, we examined the change in the success rates and processing performance of the proposed method in different dataset sizes. While valid datasets were enriched with corrupted triples, the number of triples in the final set increased at the specified rate. To calculate the actual dataset sizes, false triple ratios need to be added to these numbers. For example, 30K clusters with 60% corrupted data have a total of 48,000 triples. In evaluating the results, the expressions 10K, 20K, 30K, and 40K will be used to refer to valid triple numbers.

The spread process was run simultaneously for all triples, starting from the first triple in the dataset. Triples were added randomly in different sequences to eliminate the effect of the order of the triples on the experimental results in the adding processes. We compared the results obtained in different sorting processes and took the average of these values into account in calculating the final success rates.

All experiments were repeated thrice, and the average results for each experiment were taken into account. In this way, biased values that may occur in synchronous and asynchronous addition processes were eliminated.

D. EVALUATION METRICS

Since previous studies focused on obtaining rather than updating the confidence value, so new evaluation criteria had to be determined for the updating process. Evaluation of the propagation process can be performed according to different approaches. In the first approach, the success rate is determined according to the number of false triples deleted. Here, the *true positive* value will show the deleted false triples. The second approach is based on removing as few correct triples as possible. Here, the valid non-deleting triple number will represent the true positive value. In the result evaluation, we preferred the first approach, the number of deleted false triples as the true positive value.

Another criterion showing the success of the propagation is the distribution of confidence values before and after the process. As the weak links are deleted and the strong links become stronger, it is expected that the average of the confidence values will increase after the propagation process. For this reason, the mean confidence values of the datasets used in the experiments before and after propagation were compared.

The third criterion we used in the evaluation was the comparison of the confidence value distribution obtained with the real-world confidence value distribution. Of the datasets in which the experiments were conducted, only NELL had confidence values for correct data. The distribution of the 2,766,078 confidence values taken from this dataset is shown in Fig. 3. To test the effectiveness of the propagation process, the distributions of the correct confidences obtained from propagation were compared with this distribution.

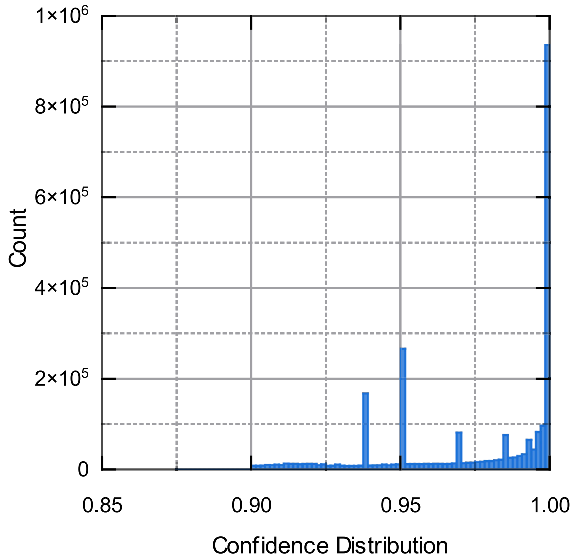

FIGURE 3. Distribution of correct confidence values defined in the NELL.

E. EXPERIMENTAL RESULTS

In this section, the results obtained according to the experimental study and evaluation metrics will be discussed.

Table 2 shows the evaluation results for different datasets and different cluster sizes. As can be seen, small changes are observed in the accuracy and precision values as the size of the dataset increases. Recall, on the other hand, start with low values in small datasets but increase in direct proportion with the set size. This trend shows that the recall rates also increase as the dataset grows.

The change in the corrupted triple rate also affects the results. In Fig. 4(a), the accuracy change in the datasets for the corruption rates is shown comparatively. Despite the increase in the corrupted triple rate, the accuracy values are above the 80%.

Fig. 4(b) shows the change in precision values in datasets for corrupted triple rates. The change in precision values is more stable than accuracy values and shows that although corrupted triple rates increase, deleted false triples remain constant compared to deleted valid triples.

Fig. 4(c) shows the recall variation for corrupted triple ratios. As can be seen, the recall value increases with the increase in corruption rates in all datasets. The recall was calculated as the ratio of deleted corrupted triples to total corrupted triples in the dataset.

To evaluate the experimental results independently of the confidence values produced by the normal distribution, the experiments were repeated on the FB15K dataset using the confidence produced with exponential distribution and random data. These experiments were performed for different dataset sizes and different corrupted triple rates. In Fig. 5, accuracy, recall, and precision values for different dataset

TABLE 2. Evaluation results of the datasets.

Datasets	Valid Triples	Accuracy	Precision	Recall
FB15K	10K	0.87100	0.98783	0.51098
	20K	0.88001	0.98344	0.54914
	30K	0.88910	0.97794	0.58678
	40K	0.89881	0.97386	0.63096
NELL	10K	0.92909	0.99562	0.73605
	20K	0.92769	0.99342	0.73359
	30K	0.92899	0.99260	0.73649
	40K	0.93105	0.99218	0.74426
WN18	10K	0.86886	0.98808	0.50938
	20K	0.86877	0.98806	0.51020
	30K	0.87216	0.98400	0.52752
	40K	0.87696	0.98324	0.54920
YAGO3-10	10K	**0.93626**	**0.99697**	**0.76157**
	20K	0.93338	0.99467	0.75277
	30K	0.93065	0.99322	0.74206
	40K	0.93094	0.99395	0.74303

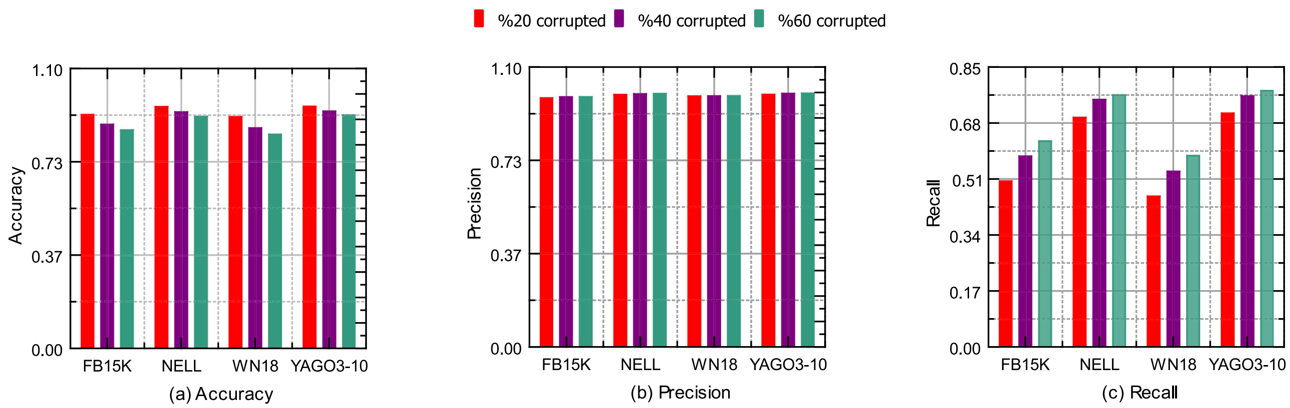

FIGURE 4. Accuracy, precision, and recall for corrupted triple rates.

sizes are compared. The evaluated values for each dataset size contain the average of different corruption rates. In Fig. 5(a), the accuracy results for the distributions are compared. This result shows that an accuracy rate of over 80% is obtained regardless of the distribution. Fig. 5(b) shows the change in precision values. Here, the results obtained with the normal distribution were around 98%, while the results were above around 80% in the other distributions. As seen in Fig. 5(c) the sensitivity results for the normal distribution range between 50% and 60%, and between 40% and 50% for the other distributions. The striking point in these experiments is that the recall results increase depending on the size of the dataset for all distributions. This confirms the improvement result obtained in the normal distribution before, depending on the size of the dataset.

The results show that the spread of confidence values contributes to the stability of the system, as suggested in the proposed model. Simultaneously, despite the increase in weak confidence, recall does not decrease; in contrast, it increases. If the assumption that weak and strong links will have an equal distribution is accepted, it can be predicted that weak triples will be cleaned more effectively at higher false triple rates such as 100% and more.

Experiment results were also analyzed according to the mean and standard deviation values of the distributions. In Fig. 6, the initial means of the datasets and the means formed after the spread process are compared.

The presence of corrupted triples in the dataset lowers the mean confidence value. Therefore, the mean confidence values in all datasets increased compared with the situation before the propagation process. This is due to the removal of weak links and reinforcement of strong links during propagation.

Contrary to the expected increase in the mean, it is expected that the standard deviation values will not increase. Fig. 6 shows the variation in the standard deviation values after propagation. As can be seen, no significant change was observed in the standard deviation values after propagation.

To better explain the cluster evaluations for mean and standard deviation, the changes in the bell curve at different

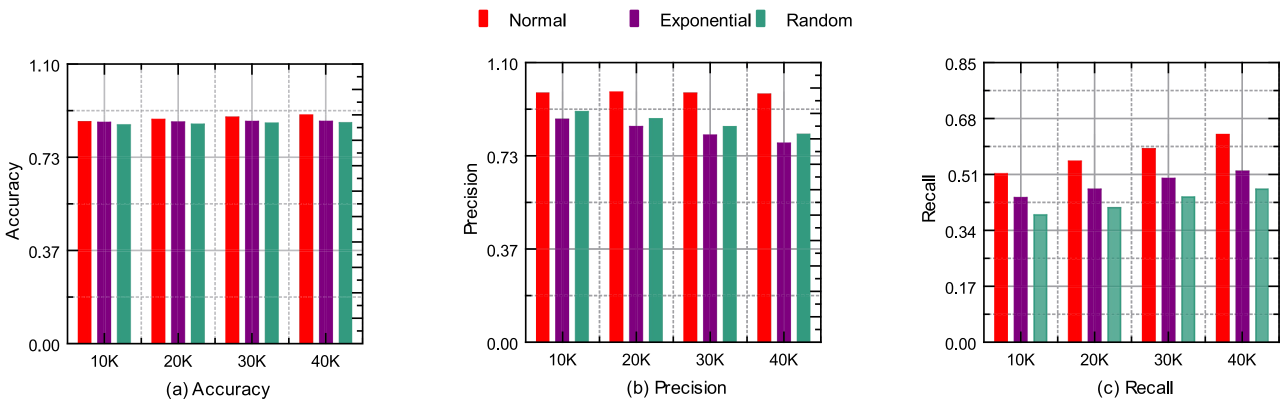

FIGURE 5. Comparison of accuracy, precision, and recall on FB15K dataset for different distributions.

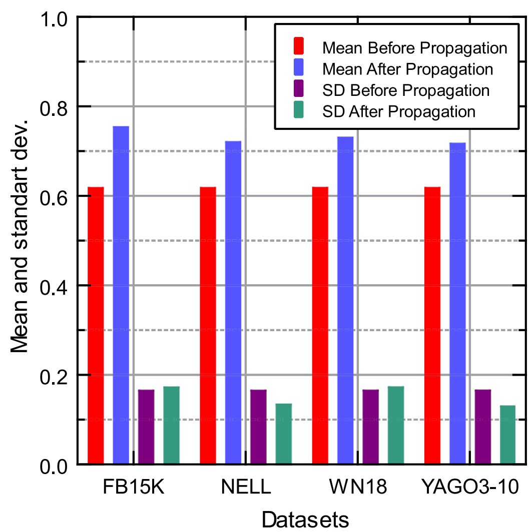

FIGURE 6. Changes in the mean and standard deviations in datasets after propagation.

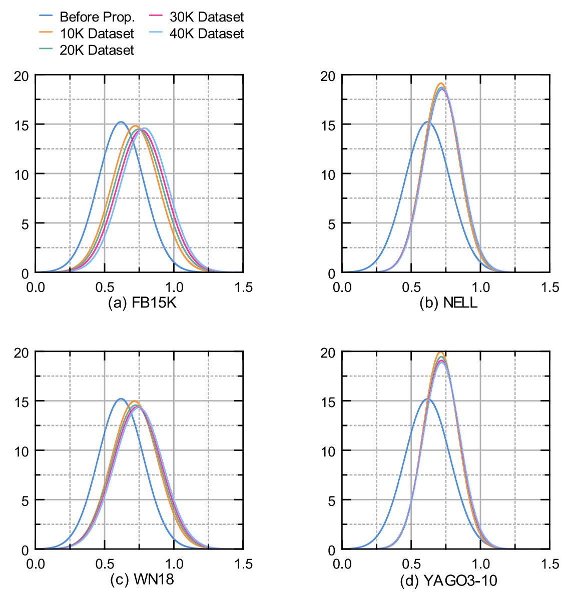

FIGURE 7. Changes in confidence for datasets.

dataset sizes can be examined. We created these figures by considering the average confidence of corrupted triple rates in different cluster sizes for all datasets. Fig. 7(a) shows the shift in the bell curves for FB15K. As can be seen, there was a rightward shift in the distributions before and after propagation in the 10K, 20K, 30K, and 40K datasets. Additionally, as the dataset size increases, the number of triples increases, resulting in an increased rightward shift.

The shift in confidence values for the NELL dataset is shown in Fig. 7(b), where similar results to the FB15K dataset are obtained.

A similar situation is shown for WN18 in Fig. 7(c). The shift due to the variation of the means increases depending on the size of the dataset. This shows that the confidence values are cleared and increased after the propagation process.

Fig. 7(d) shows the bell curve variation at different cluster sizes for the YAGO3-10 dataset. Similar results were obtained for the YAGO3-10 dataset as for the other datasets. The increase in the cluster sizes also increases the mean confidence level.

As a result, the suggestion that KG is refined better as the size of the dataset increases is empirically confirmed. This situation is also related to the rate of wrong triples added to the graph, and as the rate increases, the rate of removal of wrong information also increases. It can also be said that the refinement process is independent of the dataset.

To better measure the performance of the propagation process in different confidence value distributions, the initial and final distributions of true and false triples were also compared. Additionally, the final distributions of true triples were compared with the true triple distribution of the NELL dataset with real-world data. Fig. 8 compares the initial and final distributions of exponential, normal, and randomly generated true triples. Here, the density of confidence values in the positive direction is observed. In addition, the distribution of the correct data after the propagation process reflects the

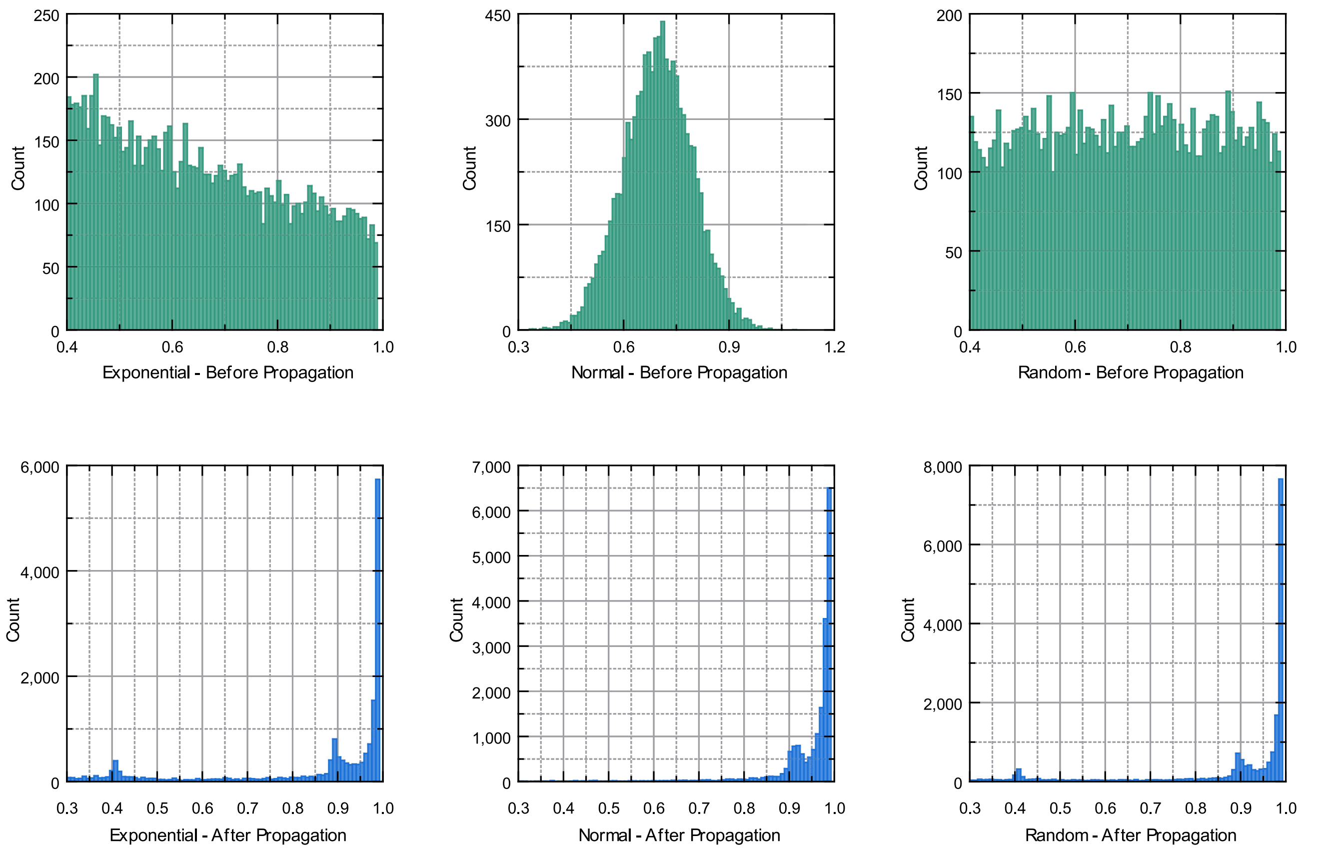

FIGURE 8. Initial and final confidence value distributions for true triples.

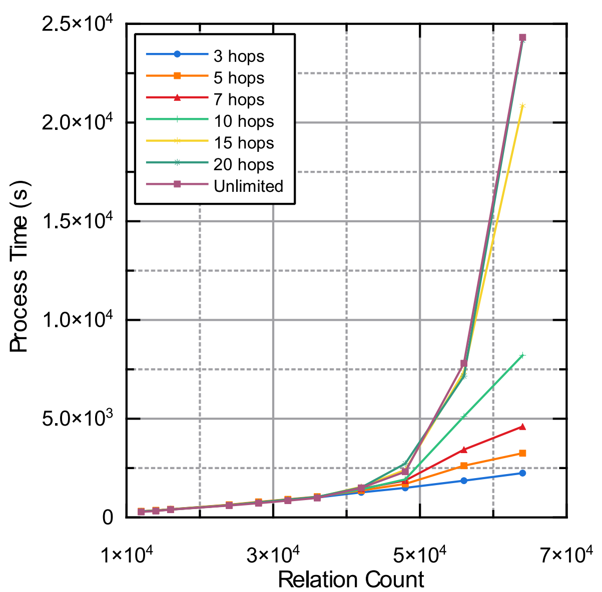

FIGURE 9. Processing time variation for FB15K by hop count.

distribution characteristic of the correct confidence values in Fig. 3 obtained from the NELL dataset.

Along with other evaluations, we also considered the performance evaluations of the propagation process. Fig. 9 shows the effect of the propagation diameter on the processing time for the FB15K dataset. As can be seen, in cases where the propagation diameter is not limited, the total time spent on the propagation increases rapidly when the total number of relations is above 40K. On the other hand, if the diameter is limited to 3-hop, the total processing times are significantly reduced. While the 3-hop spreading diameter provides an improvement in processing time performance, it also produces results very close to the unlimited propagation process in refinement results. In Fig. 10, accuracy, recall, and precision values for 3-hop propagation, other hop-limited propagations, and unlimited propagation diameters are compared.

We also examined the effect of datasets on total processing times. Fig. 11 shows the change in total processing times for datasets depending on the number of relations. As can be seen, there is no significant difference between the data sets in terms of the change in total processing times. The resulting differences are due to topological differences such as the number of nodes and the number of triangles.

Limiting the propagation process to 3-hop has a positive impact on both the evaluation results and processing time performance. This restriction produces results very close to those obtained with unlimited propagation, while significantly reducing the processing time. Considering the

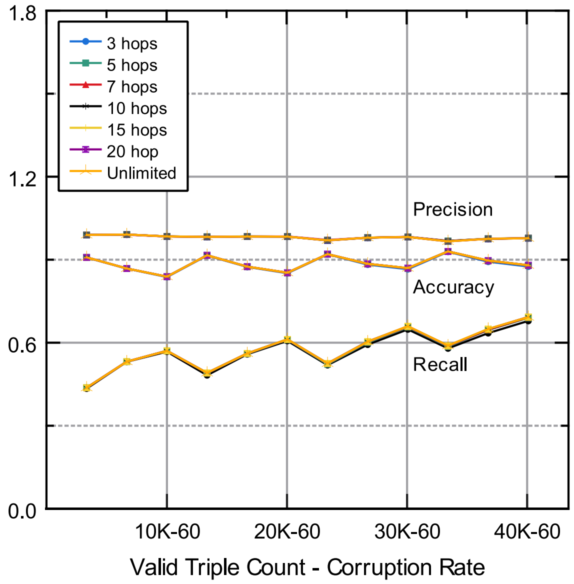

FIGURE 10. Accuracy, precision, and recall for FB15K by hop count.

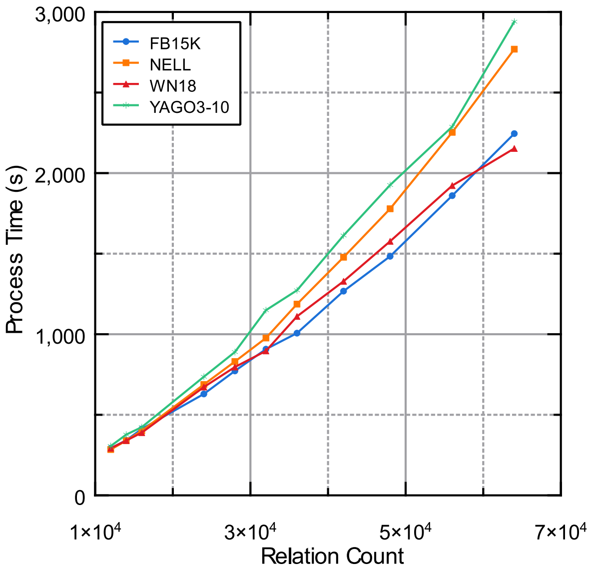

FIGURE 11. Processing time variation based on the number of relations for datasets.

expected increase in dataset size, restricting the propagation diameter is an effective way to maintain high success rates and processing performance.

VI. CONCLUSION

The propagation method introduced in this study suggests a new and different approach for KG refinement. This approach enables strong and weak triples to influence each other, taking into account the confidence values of the triples. This increases the confidence for strong information and the removal of weak information, leading to the cleaning of KG. Experimental studies reveal that the proposed method achieves an average of 90% accuracy in deleting corrupted triples. In addition, the increase in the dataset and corrupted triples positively affects the success of the refinement process. The study shows that the same success rates can be achieved in much less time if the propagation process is limited to 3 hops. Therefore, the proposed method is applicable and scalable for large KGs in terms of performance. It was confirmed that the propagation method showed similar distribution characteristics for confidence values at the end of the process compared to the real-world distribution.

Since propagation in KG is a new approach, it opens the door to different new studies. Different spread variants can be developed by transforming the propagation characteristics. For the confidence value calculation, the results can be evaluated with different sigmoid functions and vertical parameters. As another future work, the use of classification methods is planned to define the deletion threshold.

ACKNOWLEDGMENT

The authors would like to thank Shaoxiong Ji et al. for copyright permission to use the example graph in Fig. 1.

REFERENCES

[1] D. Lenat, M. Prakash, and M. Shepherd, "CYC: Using common sense knowledge to overcome brittleness and knowledge acquistion bottlenecks," *AI Mag.*, vol. 6, no. 4, pp. 65–85, 1986.

[2] A. Singhal, "Official Google Blog: Introducing the knowledge graph: Things, not strings [eb/ol]," Google Inc., Feb. 2020. Accessed: Oct. 27, 2022. [Online]. Available: https://blog.google/products/search/introducing-knowledge-graph-things-not

[3] P. A. Bonatti, S. Decker, A. Polleres, and V. Presutti, "Knowledge graphs: New directions for knowledge representation on the semantic web (dagstuhl seminar 18371)," *Dagstuhl Rep.*, vol. 8, no. 9, pp. 29–111, 2018. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10328

[4] L. Ehrlinger and W. Woß, "Towards a definition of knowledge graphs," in *Proc. CEUR Workshop*, vol. 1695, 2016, pp. 1–5.

[5] X. Li, A. Taheri, L. Tu, and K. Gimpel, "Commonsense knowledge base completion," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1445–1455.

[6] S. Tiwari, F. N. Al-Aswadi, and D. Gaurav, "Recent trends in knowledge graphs: Theory and practice," *Soft Comput.*, vol. 25, no. 13, pp. 8337–8355, Jul. 2021.

[7] B. Xue and L. Zou, "Knowledge graph quality management: A comprehensive survey," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4969–4988, May 2023.

[8] A. Hogan, "Knowledge graphs," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–37, 2021.

[9] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, and Z. Duan, "Knowledge graph completion: A review," *IEEE Access*, vol. 8, pp. 192435–192456, 2020.

[10] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017.

[11] T. Shen, F. Zhang, and J. Cheng, "A comprehensive overview of knowledge graph completion," *Knowl.-Based Syst.*, vol. 255, Nov. 2022, Art. no. 109597.

[12] Y. Zhao, S. Gao, P. Gallinari, and J. Guo, "Knowledge base completion by learning pairwise-interaction differentiated embeddings," *Data Mining Knowl. Discovery*, vol. 29, no. 5, pp. 1486–1504, Sep. 2015.

[13] J. Zhu, Z. Zheng, M. Yang, G. P. C. Fung, and Y. Tang, "A semi-supervised model for knowledge graph embedding," *Data Mining Knowl. Discovery*, vol. 34, no. 1, pp. 1–20, Jan. 2020.

[14] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, and J. Krishnamurthy, "Never-ending learning," *Commun. ACM*, vol. 61, no. 5, pp. 103–115, 2018.

[15] D. Gerber, D. Esteve, J. Lehmann, L. Bbhmann, R. Usbeck, A.-C. Ngonga Ngomo, and R. Speck, "DeFacto—Temporal and multilingual deep fact validation," *J. Web Semantics*, vol. 35, pp. 85–101, 2015.

[16] J. Pujara, H. Miao, L. Getoor, and W. Cohen, "Knowledge graph identification," in *Proc. 12th Int. Semantic Web Conf.* Sydney, NSW, Australia: Springer, Oct. 2013, pp. 542–557.

[17] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 192–202.

[18] M. N. Jeyaraj, S. Perera, M. Jayasinghe, and N. Jihan, "Probabilistic error detection model for knowledge graph refinement," *Computación Y Sistemas*, vol. 26, no. 3, pp. 1243–1257, Sep. 2022.

[19] M. Qu and J. Tang, "Probabilistic logic neural networks for reasoning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[20] A. Bordes, N. Usunier, A. Garcia-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Red Hook, NY, USA: Curran Associates Inc., 2013, pp. 2787–2795.

[21] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 601–610.

[22] H. Paulheim, "Knowledge graph refinement: A survey of approaches and evaluation methods," *Semantic Web*, vol. 8, no. 3, pp. 489–508, Dec. 2016.

[23] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022.

[24] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proc. AAAI Conf. Artif. Intell.*, 2014, vol. 28, no. 1, pp. 1–8.

[25] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 1–7.

[26] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 687–696.

[27] Y. Lin, Z. Liu, H. Luan, M. Sun, S. Rao, and S. Liu, "Modeling relation paths for representation learning of knowledge bases," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 705–714. [Online]. Available: https://aclanthology.org/D15-1082

[28] T. Trouillon, C. R. Dance, E. Gaussier, J. Welbl, S. Riedel, and G. Bouchard, "Knowledge graph completion via complex tensor factorization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 4735–4772, 2017.

[29] Y. Wang, E. Zhao, and W. Wang, "A knowledge graph completion method based on fusing association information," *IEEE Access*, vol. 10, pp. 50500–50507, 2022.

[30] M. M. Alam, M. R. A. H. Rony, M. Nayyeri, K. Mohiuddin, M. S. T. M. Akter, S. Vahdati, and J. Lehmann, "Language model guided knowledge graph embeddings," *IEEE Access*, vol. 10, pp. 76008–76020, 2022.

[31] A. Melo and H. Paulheim, "Detection of relation assertion errors in knowledge graphs," in *Proc. Knowl. Capture Conf.*, Dec. 2017, pp. 1–8.

[32] X. Chen, M. Chen, W. Shi, Y. Sun, and C. Zaniolo, "Embedding uncertain knowledge graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3363–3370.

[33] Y. Shan, C. Bu, X. Liu, S. Ji, and L. Li, "Confidence-aware negative sampling method for noisy knowledge graph embedding," in *Proc. IEEE Int. Conf. Big Knowl. (ICBK)*, Nov. 2018, pp. 33–40.

[34] T. Shao, X. Li, X. Zhao, H. Xu, and W. Xiao, "DSKRL: A dissimilarity-support-aware knowledge representation learning framework on noisy knowledge graph," *Neurocomputing*, vol. 461, pp. 608–617, Oct. 2021.

[35] S. Liu, B. Grau, I. Horrocks, and E. Kostylev, "INDIGO: GNN-based inductive knowledge graph completion using pair-wise encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 2034–2045.

[36] F. Abedini, M. R. Keyvanpour, and M. B. Menhaj, "Correction tower: A general embedding method of the error recognition for the knowledge graph correction," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 10, Sep. 2020, Art. no. 2059034.

[37] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proc. 15th Conf. Uncertainty Artif. Intell.*, 1999, pp. 467–475.

[38] J. Ma, C. Zhou, Y. Wang, Y. Guo, G. Hu, Y. Qiao, and Y. Wang, "PTrustE: A high-accuracy knowledge graph noise detection method based on path trustworthiness and triple embedding," *Knowl.-Based Syst.*, vol. 256, Nov. 2022, Art. no. 109688.

[39] L. Drumond, S. Rendle, and L. Schmidt-Thieme, "Predicting RDF triples in incomplete knowledge bases with tensor factorization," in *Proc. 27th Annu. ACM Symp. Appl. Comput.*, Mar. 2012, pp. 326–331.

[40] W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1–5.

[41] M. H. Gad-Elrab, D. Stepanova, J. Urbani, and G. Weikum, "Tracy: Tracing facts over knowledge graphs and text," in *Proc. World Wide Web Conf.*, May 2019, pp. 3516–3520.

[42] Z. H. Syed, M. Röder, and A.-C. N. Ngomo, "FactCheck: Validating RDF triples using textual evidence," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, Oct. 2018, pp. 1599–1602.

[43] J. Du, J. Z. Pan, S. Wang, K. Qi, Y. Shen, and Y. Deng, "Validation of growing knowledge graphs by abductive text evidences," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 2784–2791.

[44] A. B. M. Moniruzzaman, R. Nayak, M. Tang, and T. Balasubramaniam, "Fine-grained type inference in knowledge graphs via probabilistic and tensor factorization methods," in *Proc. World Wide Web Conf.*, May 2019, pp. 3093–3100.

[45] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

ALISETTAR HUSEYNLI received the B.S. and M.S. degrees in computer engineering from Gazi University, Tübítak, in 2010 and 2013, respectively, where he is currently pursuing the Ph.D. degree in computer engineering. He is also the Research and Development Manager of Dolusoft Software Technology Company. His research interests include knowledge graphs, big data analytics, and network security.

M. ALI AKCAYOL received the B.S., M.S., and Ph.D. degrees from Gazi University, Tübítak, in 1993, 1998, and 2002, respectively. He has been with Michigan State University, USA, in 2004, for his postdoctoral research. He is currently the Director of the Big Data and Mobile Systems Laboratory and a Full Professor with Computer Engineering Department, Gazi University. He is also teaching deep learning and big data analytics. His research interests include artificial intelligence, big data analytics, and intelligent optimization techniques.