Cost-Aware TrE-ND: Tri-embed Noise Detection for Enhancing Data Quality of Knowledge Graph

Jumana Alsubhi*, Abdulrahman Gharawi* and Lakshmish Ramaswamy
School of Computing, University of Georgia, Athens, U.S.A.

Keywords: Machine Learning, Big Data, Data Quality, Knowledge Graphs, Noise Detection.

Abstract: In the realm of machine learning, knowledge graphs (KGs) are increasingly utilized for a wide range of tasks, such as question-answering, recommendation systems, and natural language processing. These KGs are inherently susceptible to noise, whether they are constructed manually or automatically. Existing techniques often fail to precisely identify these noisy triples, thereby compromising the utility of KGs for downstream applications. In addition, manual noise detection is costly, with costs ranging from $2 to $6 per triple. This highlights the need for cost-effective solutions, especially for large KGs. To tackle this problem, we introduce Tri-embed Noise Detection (TrE-ND), a highly accurate and cost-efficient noise detection approach for KGs. TrE-ND combines semantic depth, hierarchical modeling, and scalability for robust noise detection in large knowledge graphs. We also evaluate the overall quality of these KGs using the TrE-ND approach. We validate TrE-ND through comprehensive experiments on widely recognized KG datasets, namely, FB13 and WN11, each containing varying degrees of noise. Our findings demonstrate a substantial improvement in noise detection and KG evaluation accuracy as compared to existing methods. By utilizing the TrE-ND approach, we manage to flag noisy triples with an average approximate accuracy of 87%, even when up to 40% of the dataset contains noise. This simplifies the subsequent verification process by domain experts and makes it more cost-effective. Therefore, our proposed method offers a viable solution for efficiently addressing the persistent issue of noise in KGs. This work also paves the way for future research in cost-aware noise mitigation techniques and their applications in various domains.

1 INTRODUCTION

Knowledge graphs (KGs) have emerged as powerful tools for representing structured knowledge, playing a crucial role in a variety of machine learning applications such as semantic search, question-answering, recommendation systems, and natural language processing (Paulheim, 2017), (Hogan et al., 2021). KGs are essentially a network of nodes or entities interconnected through various edges or relationships, encapsulating knowledge in a machine-readable form. Over the past few years, there has been a surge in the number of large-scale KGs.

However, the construction and utilization of KGs are far from flawless. A pervasive issue with KGs, especially with very large KGs, is the inevitable presence of noise or errors, which often distort the knowledge representation and interpretation. Noise in KGs can arise due to numerous reasons, including inaccurate entity linking, misinterpretation of relations, mistakes in automatic data extraction from unstructured sources, or even simple human errors during manual KG construction. Incorporating heterogeneous data from diverse sources during KG construction often exacerbates this problem (Paulheim, 2017).

Knowledge graphs typically consist of a set of triples, where each triple is a pair of entities connected by a relation. A triple consists of a head entity, relation, and tail entity (h,r,t). Noisy triples are triples that contain some form of error or *noise*. Noisy triples degrades the overall quality of the KG, leading to inaccurate inferences and potentially causing subsequent tasks, such as querying or other machine learning applications, to operate on incorrect assumptions. Consequently, the identification and mitigation of such noise in a cost-efficient manner becomes a critical issue (Hogan et al., 2021).

Noise in knowledge graphs typically falls into two categories: semantic noise and structural noise. Semantic noise is characterized by subtle semantic errors, which can occur even when entity types

* These authors contributed equally to this work.

Alsubhi, J., Gharawi, A. and Ramaswamy, L.
Cost-Aware TrE-ND: Tri-embed Noise Detection for Enhancing Data Quality of Knowledge Graph.
DOI: 10.5220/0012431700003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 1020-1027
ISBN: 978-989-758-680-4; ISSN: 2184-433X

align correctly, due to inaccurate assignment of relationships or misinterpretation of an entity's context. Conversely, structural noise refers to obvious errors that are readily identifiable, such as when an entity is paired with an incorrect relation type. The primary focus of this paper is the detection of semantic noise, which is more challenging to detect because it involves understanding subtle nuances and context, which often require domain expertise for accurate identification.

There are several techniques presented in the literature to detect noise in knowledge graphs. Many noise detection techniques depend on graph embedding methods. These methods lack robustness since their performance is significantly affected by noise, often leading to reduced accuracy (Choudhary et al., 2021), (Ji et al., 2021), (Wang et al., 2017), (Gesese et al., 2019), (Dai et al., 2020). Other noise detection methods rely on manual inspection by domain experts, which is costly, especially in large KGs (Qi et al., 2022), (Gao et al., 2019).

In this paper, we address these limitations by proposing a novel method called *Tri-embed Noise Detection (TrE-ND)* to enhance the quality of knowledge graphs. Our approach not only yields higher accuracy in noise detection compared to existing methods but is also cost-effective. TrE-ND is unique in that it harnesses the capabilities of three distinct knowledge graph embeddings (KGE), to effectively flag noisy triples for subsequent validation by domain experts.

In designing TrE-ND this paper makes the following contributions:

- First, we demonstrate how to effectively combine ensemble learning and knowledge graph embedding so as to leverage their unique strengths for enhancing the effectiveness of noise detection in KGs.
- Second, we present a hybrid approach that astutely blends automated noise detection with human expertise for cost-effective and accurate noise detection in knowledge graphs.
- Third, we conduct an extensive experimental study on widely used benchmark KG datasets, FB13 and WN11, and with different noise levels to thoroughly evaluate the costs and benefits of the proposed TrE-ND approach. This experimental study shows that our hybrid approach outperforms existing solutions in both accuracy and cost-effectiveness, demonstrating its clear advantage over other methods, especially when dealing with large-scale KGs with significant levels of noise.

The rest of the paper is structured as follows: Section 2 delves into our motivation and background. Our framework is detailed in Section 3, followed by our experiments and results in Section 4. Section 5 presents an ablation study. We conclude with future research recommendations in Section 6.

2 MOTIVATION AND BACKGROUND

A knowledge graph (KG) is a structured database that is used to represent knowledge in a form that can be easily processed by machines, and it is used in various domains, including natural language processing, search engines, and recommender systems, to organize and make sense of huge amounts of data. A knowledge graph G can be modeled as a set of triples: $G = \{t_1, t_2, \ldots, t_n\}$, where each triple t is of the form: $t = (h, r, t)$. The error in a noisy triple t' can appear in the head or tail of the triple such as $t = (h', r, t)$ or $t = (h, r, t')$.

The presence of noise and contradictions in a knowledge graph is a significant issue due to the imprecision of data sources and inaccuracies in the extraction process. For example, NELL, a widely used knowledge graph, experiences a gradual decrease in the accuracy of the knowledge it acquires over time. Initially, NELL provides an estimated precision of 0.9 in its first month, but this falls to 0.71 after two months (Chen et al., 2020). This decline is mainly due to the imperfect reliability of extraction patterns, leading to the occasional extraction of noisy triples. Therefore, it becomes necessary to develop robust noise detection methods that can accurately detect the noise in KGs. Furthermore, it is also essential to develop cost-effective KGs evaluation frameworks so we can determine their fitness for specific applications.

There are several techniques presented in the literature to detect noise in knowledge graphs, and many of these techniques depend on graph embedding. However, their effectiveness is often compromised by the presence of noise (Dai et al., 2020).

Knowledge graph embedding aims to maintain the structural and semantic information of the knowledge graph in the learned embeddings in order to map items with comparable relationships or attributes to nearby locations in the vector space. Knowledge graph embeddings can be learned using a variety of methods, including TransE (Bordes et al., 2013), TransH (Wang et al., 2014), TransR (Lin et al., 2015), TransD (Ji et al., 2015), TransG (Xiao et al., 2016), DistMult-HRS (Zhang et al., 2018), AATE (An et al., 2018), ConvKB (Nguyen et al., 2018), KG-BERT (Yao et al., 2019), and others. Although the underlying assumptions and target functions of these techniques vary, they all strive to capture the intricate connections between entities and characteristics in a knowledge graph. Each of these KGE models has its strengths and limitations. While some of these methods demonstrate great performance on clean knowledge graphs, they lack robustness and resilience when dealing with noisy KGs. Unlike other methods, TrE-ND approach leverages the unique advantages of three techniques to create a more robust and accurate noise detection method for knowledge graphs, especially when dealing with KGs with high levels of noise. Thus, our approach is capable of detecting semantic noise in KGs even in the presence of high levels of noise. TrE-ND can identify contextual noise, deal with hierarchical complexity, and adjust to the relational dynamics.

Furthermore, numerous approaches have been proposed to evaluate and assess the quality of KGs (Qi et al., 2022), (Gao et al., 2019). However, these approaches focus on human annotation and cluster sampling, which is time-consuming and labor-intensive. They do not automatically identify potentially problematic triples. Our method, on the other hand, simplifies the assessment process by automatically flagging noisy triples, reducing the need for extensive human annotation and thereby offering a more cost-effective strategy for KG quality assurance. This underscores the value of our approach in terms of cost efficiency, demonstrating its clear advantage over other methods, especially when dealing with large-scale KGs.

Additionally, detecting noise manually by domain experts in a KG is much more expensive. The cost of noise detection has been estimated by (Paulheim, 2018), and Table 1 shows the estimated cost of various knowledge graphs. Manual noise detection costs range between $2 to $6, dramatically higher compared to automated methods, which cost as little as $0.01. As shown, the manual validation method incurs high costs for noise detection in Cyc and Freebase KGs, at a total of $120M and $6.75B, respectively. In contrast, automatic validation applied to DBpedia and YAGO KGs significantly reduces the cost, ranging from $5.1M to $11.6M.

In addition, the cost and accuracy of validating triples in knowledge graphs are significantly influenced by the level of expertise involved. The spectrum ranges from expert to novice to automatic validation, showing a corresponding decrease in cost as the level of expertise reduces. However, this reduction in cost is typically accompanied by an increase in error rate. Manual noise detection relies on human validators to examine the triples in a KG. The human validation procedure, however, becomes more expensive and less scalable as knowledge graphs grow in size, making it unfeasible to handle large data present in contemporary KGs. To overcome these limitations, we propose TrE-ND to detect noise and assess the quality of KGs.

Table 1: Cost comparison of knowledge graphs.

KG	Method	Cost/Triple	# Triples	Total Cost
Cyc	Manual	$5.71	21M	$120M
Freebase	Manual	$2.25	3B	$6.75B
DBpedia	Automatic	$0.0185	400M	$5.1M
YAGO	Automatic	$0.0083	1.4B	$11.6M

3 TrE-ND NOISE DETECTION FRAMEWORK

Figure 1 provides an architecture of our proposed TrE-ND approach. The unique feature of this approach is that it combines ensemble learning in conjunction with three distinct knowledge graph embedding, which are KG-BERT, DistMult-HRS, and TransD.

3.1 Base Models Selection

Firstly, we analyzed the performance of several KGE techniques. The selection of KG-BERT, DistMult-HRS, and TransD as the basis for our TrE-ND approach was driven by their complementary strengths in knowledge graph embedding. KG-BERT is known for its superior performance in capturing complex relationships and handling textual attributes in KGs, KG-BERT excels in understanding the semantic nuances that are often present. It contributes to our approach by providing a deep, context-aware understanding of the entities and relations. DistMult-HRS is effective in modeling hierarchical relations and capturing transitive properties in KGs. Its ability to model different types of relations contributes to the diversity and coverage of our TrE-ND approach. TransD is excellent at dealing with the scalability of large KGs and shows a strong ability to generalize over different types of relationships. Its efficiency and scalability make it ideal for handling large datasets. Thus, based on extensive experiments, TrE-ND can detect semantic noise in large KGs, even in the presence of high levels of noise. Additionally, we conducted an ablation study in Section 5 to verify our design choices.

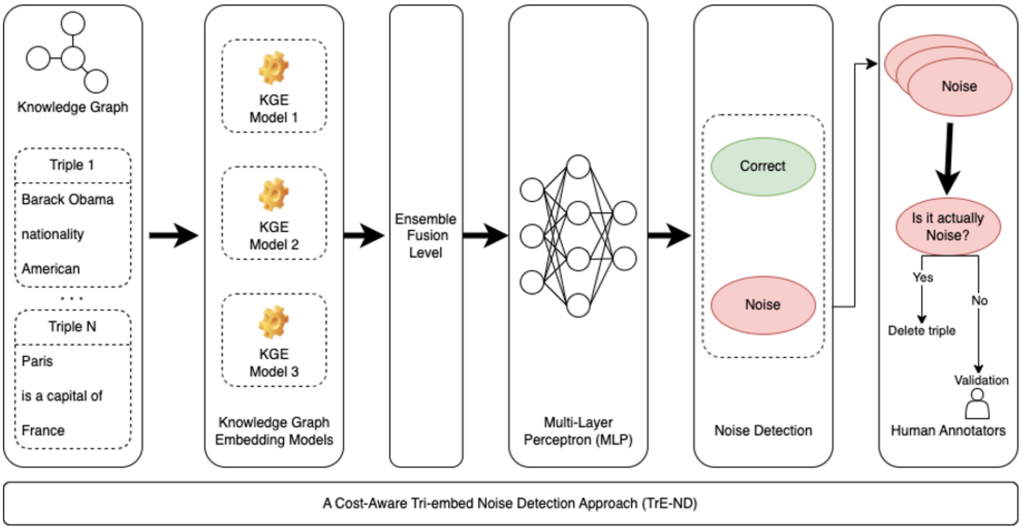

Figure 1: The architecture of our TrE-ND approach.

3.2 Embedding Representation

Given the three embedding models KG-BERT ($E_{\text{KG-BERT}}$), DistMult-HRS ($E_{\text{DistMult-HRS}}$), and TransD (E_{TransD}), the embedding for each triple t_i using each of these models can be represented as:

$$V_{\text{KG-BERT}}(t_i) = E_{\text{KG-BERT}}(t_i)$$

$$V_{\text{DistMult-HRS}}(t_i) = E_{\text{DistMult-HRS}}(t_i)$$

$$V_{\text{TransD}}(t_i) = E_{\text{TransD}}(t_i)$$

Here, $V_{E_i}(t_i)$ denotes the embedded representation of triple t_i using the specified embedding model E_i. Next, these embeddings are concatenated or combined to serve as input to the multi-layer perceptron (MLP). Let MLP be a function representing this multi-layer perceptron. For each triple t_i, the input to the MLP will be the combined embeddings, and the output $O(t_i)$ will be a binary classification for noise or correct:

$$O(t_i) = \text{MLP}(V_{\text{KG-BERT}}(t_i), V_{\text{DistMult-HRS}}(t_i), V_{\text{TransD}}(t_i))$$

Where:

$$O(t_i) = \begin{cases} 1 & \text{if } O(t_i) \geq 0.5, t_i \text{ is noise} \\ 0 & \text{if } O(t_i) < 0.5, t_i \text{ is correct} \end{cases}$$

Overall, TrE-ND involves using three different embedding methods, KG-BERT, DistMult-HRS, and TransD, trained separately and then incorporated into a Multilayer Perceptron for final classification.

Therefore, we used three distinct knowledge graph embedding methods that are able to capture the relational semantics of the graph accurately even in the presence of noise. Each embedding model has its own unique strengths. Thus, using these three models provides a more robust and accurate solution by leveraging their individual strengths and offsetting their weaknesses. We trained each one on the same training set and with the same level of noise, with the aim of optimizing entity representations for accurate triple classification or noise detection.

After training the embedding methods with three different levels of noise, 10%, 20%, and 40%, they generate low-dimensional representations of entities and relationships. Then, we independently evaluated the effectiveness of each technique in the presence of noise using the testing set. We utilized rank-based metrics such as Mean Rank (MR), Hits@k, and Mean Reciprocal Rank (MRR) to measure each embedding model's performance in relationship prediction and triple classification. This sheds light on the ideal embedding model to actively utilize in our final architecture.

3.3 Ensemble Fusion

To ensemble the three embeddings in our TrE-ND, we trained the MLP using these embeddings from the three distinct KGE models, associating them with their majority label, either noisy or correct. The MLP was selected for its computational efficiency, which is crucial for large-scale knowledge graphs, its generalization capability that aids in robust noise detection, and its ease of use that facilitates quick implementation and experimentation. Moreover, based on extensive experimentation, we chose to use MLP as the aggregation layer in our TrE-ND approach.

Thus, the three embeddings produced were combined and fed a Multilayer Perceptron after individ-

Table 2: Statistics of the FB13 and WN11 datasets.

Dataset	Entities	Relation	Train	Valid	Test
FB13	75,043	13	316,232	5,908	23,733
WN11	38,696	11	112,581	2,609	10,544

ual evaluations and noise introduction. The MLP was trained to identify noisy triples, using the outputs from each of the three embeddings as input features for each triple. The MLP consists of three hidden layers, each using a Sigmoid activation function with a learning rate of 0.01. To prevent overfitting, a dropout rate of 0.5 was applied for regularization.

In the model evaluation phase, we assessed TrE-ND's accuracy in identifying noisy triples using the test set. The procedure entailed passing the test triples through each embedding model, followed by channeling the resulting embedding vectors into the pretrained MLP. Subsequently, we assess the model's predictive accuracy by comparing the classification results obtained by MLP with the actual labels. We measure the model performance using accuracy, which is a standard binary classification metric. This evaluative process was iteratively conducted across different noise levels to gauge the model's resilience against varying noise intensities.

3.4 Human Validation

After automatically detecting noisy triples, we suggest a secondary layer of validation where the flagged noisy triples are assessed by human experts in the domain. This hybrid approach provides an additional safety net, ensuring that any false positives from the automated process are corrected, thus elevating the confidence in the noise detection results while minimizing the cost of human involvement since human experts are only given the noisy triples to check.

Additionally, after identifying and flagging noisy triples automatically using TrE-ND, we can assess the quality of the KG. Through this approach for quality assessment, we not only contribute to the individual quality of the KG but also offer a scalable and cost-effective model that can be adapted for larger, more complex KGs. Therefore, our methodology has significant implications for both the theoretical understanding and practical application of KG quality assessment.

4 EXPERIMENTS AND RESULTS

We evaluate the performance of TrE-ND for noise detection and KGs evaluation on popular benchmarks FB13 and WN11. Then, we show and analyze the experimental results.

4.1 Datasets

Since there are no explicitly labeled noises or conflicts in these datasets, new datasets with different levels of noise based on FB13 and WN11 were generated.

To mimic the real-world dataset, we used a common approach to introduce noise into the training data systematically (Xie et al., 2018). To generate semantic noise, we corrupt a portion of the original triples by randomly selecting a triple, and swapping its head or tail entity with another similar entity, creating a noisy triple. In other words, noisy triples were created by modifying the head or tail of a given triple. For example, given a true triple like (Barack Obama, Nationality, American), a plausible negative example could be (Barack Obama, Nationality, Canadian) instead of an obviously incorrect one like (Barack Obama, Nationality, Soccer). This is because Canadian and American are both common tails for the Nationality relation. Thus, the generation of noisy triples was guided by the principle that the entity chosen for replacement should have previously been seen in the same position. This approach is driven by the fact that most errors in real-world KGs stem from confusion between similar entities. Hence, our proposed method, TrE-ND, is evaluated for detecting semantic noise, which is harder to detect since it often requires specialized expertise for detection, and it is more common in real-world KGs. Table 2 displays statistics of the datasets.

4.2 Noise Detection

Noise detection or triple classification is the task of verifying whether an unseen triple is true or not. We can determine a score for any triple after learning an embedding model on the KG. Triple classification can be done based entirely on these triple scores. Triples that score higher are more likely to be true facts. An unseen triple will be forecasted as true if the score is higher than a threshold; if not, it will be predicted as false. This task can be evaluated using conventional classification metrics like accuracy (Wang et al., 2017), (Dai et al., 2020).

Detecting noise in knowledge graphs is a resource-intensive process in terms of time and cost. TrE-ND automates the noise detection process in KGs, drastically curtailing costs tied to human annotators. We evaluate the performance of TrE-ND for noise detection in FB13 and WN11 datasets with different levels of noise. We changed the noise levels in our training sets to investigate the model's performance under various conditions. Specifically, we cre-

Table 3: Accuracy of noise detection on FB13 and WN11 with 10%, 20%, and 40% of noise. The best scores are highlighted in bold.

Method	FB13	FB13-%10	FB13-%20	FB13-%40	WN11	WN11-%10	WN11-%20	WN11-%40	AVG
TransE	81.1	78.2	76.1	73.3	74.9	71.7	68.4	64.7	73.5
TransH	83.1	80.5	78.3	75.2	78.3	75.8	73.0	69.4	76.7
TransR	82.2	79.1	75.8	72.4	85.7	82.2	78.1	75.3	78.8
TransD	89.1	87.7	84.9	81.2	86.2	83.3	80.5	76.1	83.6
TransG	87.2	86.1	83.4	80.1	87.3	85.4	82.7	79.4	83.9
DistMult-HRS	88.9	87.6	84.2	78.3	88.8	85.1	81.9	79.2	84.2
AATE	87.1	85.3	82.6	77.9	87.6	85.3	82.8	78.8	83.4
ConvKB	87.6	84.9	81.3	78.6	86.2	84.0	84.2	81.5	83.5
KG-BERT	**90.3**	88.2	85.1	80.7	**92.8**	87.6	85.1	82.0	86.4
TrE-ND	88.6	**88.3**	**85.4**	**82.7**	91.1	**89.0**	**87.3**	**84.5**	**87.1**

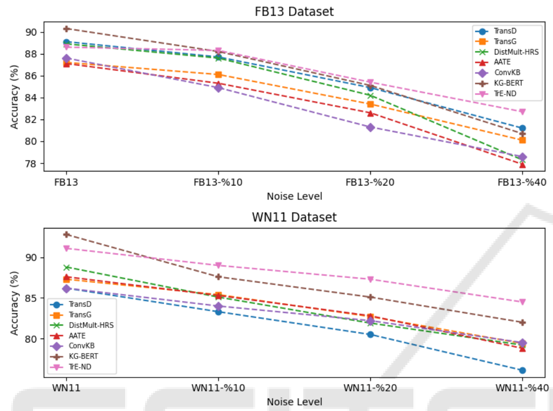

Figure 2: Best performing noise detection methods in FB13 and WN11 datasets.

ated three versions of the dataset with noise levels at 10%, 20%, and 40%. This approach allowed us to assess the robustness of TrE-ND model across different noise intensities.

The results of evaluation noise detection methods are shown in Table 3, which presents the accuracy rates of these various noise detection methods. We evaluated TrE-ND with other models such as TransE, TransH, TransR, TransD, TransG, DistMult-HRS, AATE, ConvKB, and KG-BERT in FB11 and WN11 datasets under different levels of noise. Moreover, Figure 2 presents the best noise detection methods, in terms of accuracy, when applied to FB13 and WN11 datasets. The results in Figure 2 highlight the superior performance of the TrE-ND model in detecting noise within the FB13 and WN11 datasets. On average, our TrE-ND model outperforms other models when it comes to noise detection in both FB13 and WN11 datasets. As the level of noise increases, TrE-ND model shows more robustness compared to other models, which struggle to detect noise in a highly noisy environment.

TrE-ND is an ideal cost-effective solution for noise detection in real-world KGs where noise is both subtle and abundant. In large-scale KGs, even a small percentage increase in the accuracy of noise detection methods can have a significant impact. However, the experimental results show that the improvements of TrE-ND are significant ($p < 0.05$).

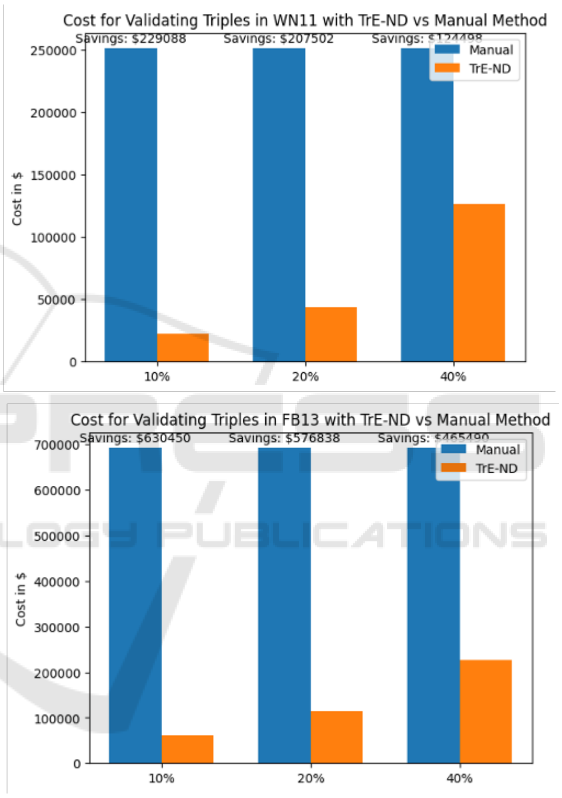

Figure 3: Cost for validating triples in FB13 and WN11 with TrE-ND vsmanual with 10%, 20%, and 40% of Noise.

Thus, TrE-ND emerges as a highly cost-effective and accurate solution for validating noisy triples, particularly when compared to other automated methods and manual annotators, who charge between $2 to $6 per triple validation. With its high accuracy rates across different noise levels in both FB13 and WN11 datasets, TrE-ND minimizes false positives

Table 4: Knowledge graph evaluation using TrE-ND.

Noise%	FB13	WN11	Gold Standard Accuracy
10	91.17	91.08	90
20	82.94	82.57	80
40	66.93	66.25	60

and false negatives. This ensures that only genuinely noisy triples get flagged for review, which can substantially reduce the volume of triples requiring validation. Consequently, the method can lead to significant cost savings: fewer flagged triples mean less manual validation and, therefore, lower costs. When compared to other automated methods that show less accuracy, the cost-effectiveness of TrE-ND becomes even more apparent. Figure 3 shows the cost savings achieved when using the hybrid TrE-ND approach as compared to manual validation, assuming that experts validate the flagged triples at a cost of $2 per triple. There is a tradeoff between cost and accuracy, when human validators are involved to detect noise, the cost and accuracy increase. For example, the accuracy for this hybrid approach increased from 85% to 97% in FB13 with 20% of noise. Furthermore, if the validators were also asked to fix the errors in the triples, the accuracy of the KG will be around 99.97%. Therefore, we can detect almost all the noise in the KG using our hybrid approach. In summary, all the methods show a decrease in performance as the noise level increases. However, TrE-ND demonstrates superior performance, maintaining high accuracy rates even at high noise levels. These results indicate the potential of this approach in providing cost-effective and robust noise detection in KGs.

4.3 Knowledge Graph Evaluation

It is important to evaluate the quality of knowledge graphs to understand their reliability, accuracy, and usefulness to advise downstream applications. Thus, in this experiment, we automatically evaluate the quality of knowledge graphs using TrE-ND. The primary metric used for KG quality evaluation is accuracy, calculated based on the noisy triples detected by our TrE-ND model. Thus, using TrE-ND, we evaluate the quality of FB13 and WN11 datasets with different levels of noise 10%, 20%, and 40%. Initially, the TrE-ND model is used to detect and flag the noisy triples in the KG dataset. The number of correct triples in the KG is calculated by subtracting the number of detected noisy triples from the total number of triples in the dataset. Finally, the accuracy of the KG is calculated by taking the number of correct triples and dividing it by the total number of triples.

Table 4 displays the evaluation of the FB13 and WN11 knowledge graphs at various noise levels. The gold standard accuracy for these knowledge graphs at noise levels of 10%, 20%, and 40% is 90%, 80%, and 60%, respectively. These results demonstrate the effectiveness of TrE-ND in accurately evaluating KGs under various conditions. TrE-ND shows resilience to different noise levels, indicating its robustness for the evaluation of knowledge graph quality.

5 ABLATION STUDY

In this section, we assess the performance of different components of TrE-ND to determine the most impactful models for noise detection. We tried multiple combinations of KGE models: TransG, TransR, and TransD; DistMult-HRS, AATE, and TransG; and KG-BERT, DistMult-HRS, and TransD. Table 5 illustrates the performance of these models on the FB13 and WN11 datasets under varying levels of noise. The proposed TrE-ND, which leverages KG-BERT, DistMult-HRS, and TransD, yields the highest average accuracy across both datasets and all levels of noise, making it the most robust solution in our study. This result underscores the contribution of each KGE model, particularly when dealing with high levels of noise. TrE-ND-1 consists of DistMult-HRS, AATE, and TransG, also performs well across both datasets, demonstrating the effectiveness of this combination in maintaining performance under different noise conditions. Its average score places it as the second-best performing model.

The last version shown in Table 5, TrE-ND-2, employs a combination of three KGE models: TransG, TransR, and TransD. TrE-ND-2 exhibits a lower average accuracy compared to the other two versions. Despite this, it still presents reasonable performance for noise detection in knowledge graphs. Thus, this approach provides competitive results across all datasets, showing its effectiveness for noise detection in KGs.

Table 5: Accuracy of TrE-ND approach with different combinations of KGE models for noise detection on FB13 and WN11 with 10%, 20%, and 40% of noise. Best scores are highlighted in bold, and second-best scores are underlined.

Method	FB13	FB13-%10	FB13-%20	FB13-%40	WN11	WN11-%10	WN11-%20	WN11-%40	AVG
TrE-ND	**88.6**	**88.3**	**85.4**	**82.7**	**91.1**	**89.0**	**87.3**	**84.5**	**87.1**
TrE-ND-1	87.8	85.9	83.6	80.3	88.2	86.1	84.4	81.9	84.7
TrE-ND-2	87.4	86.1	83.8	80.1	84.6	82.5	79.7	76.7	82.6

6 CONCLUSION

In conclusion, this research presents an innovative and cost-effective approach for the detection of noise in knowledge graphs, a pervasive issue that hinders the effective utilization of KGs for a range of machine learning applications. We propose TrE-ND, which harnesses the strengths of multiple KGE models, leading to a statistically significant improvement in noise detection accuracy. The experiment results show the resilience of this approach in noise detection and KG evaluation, even under high levels of noise.

Future research needs to continue exploring cost-effective and efficient noise detection strategies, enabling the full potential of KGs in various domains. There is significant promise in the development of techniques that optimally balance cost and accuracy. In particular, the development of semi-automatic methods that leverage the integration of Large Language Models (LLM) for the validation of triples could offer a promising avenue.

REFERENCES

An, B., Chen, B., Han, X., and Sun, L. (2018). Accurate text-enhanced knowledge graph representation learning. In *NAACL*, page 745–755.

Bordes, A., Usunier, N., and Garcia-Duran, A. (2013). Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, pages 2787–2795.

Chen, X., Jia, S., and Xiang, Y. (2020). A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948.

Choudhary, S., Luthra, T., Mittal, A., and Singh, R. (2021). A survey of knowledge graph embedding and their applications. *arXiv preprint arXiv:2107.07842*.

Dai, Y., Wang, S., Xiong, N. N., and Guo, W. (2020). A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics*, 9(5):750.

Gao, J., Li, X., Xu, Y. E., Sisman, B., Dong, X. L., and Yang, J. (2019). Efficient knowledge graph accuracy evaluation. *arXiv preprint arXiv:1907.09657*.

Gesese, G. A., Biswas, R., and Sack, H. (2019). A comprehensive survey of knowledge graph embeddings with literals: Techniques and applications. In *DL4KG@ ESWC*, volume 3140.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., et al. (2021). Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37.

Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. (2015). Knowledge graph embedding via dynamic mapping matrix. In *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Language Process.*, page 687–696.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Lin, Y., Zhang, J., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*.

Nguyen, D. Q., Nguyen, D. Q., Nguyen, T. D., and Phung, D. (2018). A convolutional neural network-based model for knowledge base completion and its application to search personalization. *Semantic Web*.

Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8:489–508.

Paulheim, H. (2018). How much is a triple? In *ISWC*.

Qi, Y., Zheng, W., Hong, L., and Zou, L. (2022). Evaluating knowledge graph accuracy powered by optimized human-machine collaboration. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1368–1378.

Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28.

Xiao, H., Huang, M., and Zhu, X. (2016). Transg: A generative model for knowledge graph embedding. In *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, page 2316–2325.

Xie, R., Liu, Z., Lin, F., and Lin, L. (2018). Does william shakespeare really write hamlet? knowledge representation learning with confidence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 3.

Yao, L., Mao, C., and Luo, Y. (2019). Kg-bert: Bert for knowledge graph completion. *arXiv:1909.03193*.

Zhang, Z., Zhuang, F., Qu, M., Lin, F., and He, Q. (2018). Knowledge graph embedding with hierarchical relation structure. In *EMNLP*, page 3198–3207.