On the Role of Inductive Graph Reasoning on Improving Resilience against Backdoor Attacks to Commonsense Knowledge Graphs

Wenbin Hu*, **Yize Cheng***, **Zhuo Zeng***
Department of Computer Science and Engineering, HKUST
Hong Kong SAR, China
{whuak, ychengbt ,zzengan}@connect.ust.hk

Abstract

Backdoor attack, which is inserting attack triggers to data during the training period to poison certain models, is gaining increasing popularity in the community. However, no previous work has covered the attack on CommonSense Knowledge Graphs (CSKG). In this project, we investigate the role of inductive graph reasoning on improving the resilience against backdoor attacks to CSKG by perturbing the training process of KG-BERT and KG-BERTSAGE, two representative commonsense knowledge graph learning models, using backdoor attack techniques, and comparing their performance and attack success rate on a downstream link prediction task. Through extensive experiments using different datasets, different language model backbones, and different poison rates, we discovered that the current way of conducting inductive graph reasoning and leveraging neighboring aggregation in KG-BERTSAGE cannot help the model become resilient to backdoor attacks. We will intersect our future work with more challenging tasks such as commonsense knowledge graph population with newly annotated evaluation set.

1 Introduction

Commonsense knowledge, which is facts about the everyday world, such as "Lemons are sour," that all humans are expected to know, is claimed to be an indispensable part of artificial intelligence (Davis and Marcus, 2015). As Commonsense Knowledge is crucial for many natural language processing systems to conduct commonsense reasoning at a human level, multiple commonsense knowledge bases (CSKB) (Speer et al., 2017; Sap et al., 2019; Hwang et al., 2021) have been collected systematically to acquire commonsense data. The information in large-scale CSKBs is often integrated into graph structures to effectively leverage the embedded commonsense knowledge, forming commonsense knowledge graphs (CSKG) (Ilievski et al., 2021).

With the unprecedented success achieved by large Pre-Trained Language Models (PTLM) (Devlin et al., 2018; Liu et al., 2019; Radford et al., 2018; He et al., 2020) on handling natural human languages describing eventualities, and Graph Neural Networks (GNN) on processing ordinary graph-structured data, multiple commonsense knowledge graph learning models, of which KG-BERT (Yao et al., 2019) and KG-BERTSAGE (Fang et al., 2021a) are two representatives, have been proposed to effectively learn knowledge from various commonsense knowledge graphs and apply them to downstream tasks. KG-BERTSAGE, which conducts inductive graph reasoning by aggregating and leveraging the neighboring information in the CSKG, was shown to be capable of achieving stronger performance in the CSKG population task compared to KG-BERT (Fang et al., 2021a). However, performance is no longer the only criterion for evaluating contemporary deep learning models. With the reveal of the vulnerability of deep learning models to multiple kinds of attacks, such as backdoor attacks (Gu et al., 2019; Dai et al., 2019; Zhang et al., 2021), the resilience and robustness against such attacks has become a significant concern.

In this project, we investigate the role of inductive graph reasoning on improving the resilience against backdoor attacks to common-sense knowledge graphs. We compare the performance and attack success rate of KG-BERT and KG-BERTSAGE on a link prediction task in CSKG. Experiment results show that the current neighborhood aggregation in KG-BERTSAGE is insufficient in terms of improving resilience against backdoor attacks.

* Equal Contribution. Authors in alphabetical order.

2 Related Work

2.1 Commonsense Knowledge Graph Learning and Reasoning

As existing large scale CSKBs and CSKGs are often sparse and incomplete, Commonsense Knowledge Graph Learning and Reasoning tasks such as CSKG completion (Li et al., 2016; Malaviya et al., 2020) and CSKG population (Fang et al., 2021a) along with powerful models that leverage large pretrained language models and graph neural networks were proposed to fill the missing relationships in existing CSKGs or predict the plausibility of unseen relationships during training. In this project, two commonsense knowledge graph learning models, KG-BERT (Yao et al., 2019) and KG-BERTSAGE (Fang et al., 2021a), were used in our experiments.

2.2 Backdoor Attacks

The Backdoor attack problem essentially belongs to the training time integrity problem of deep learning models. Unlike adversarial attacks (Goodfellow et al., 2015; Madry et al., 2017; Bai et al., 2020), where the attacker introduces perturbations to the test samples during test time, the attacker of a backdoor attack modifies a proportion of the training dataset on which the model is going to be trained. This proportion is called the poison rate. And the modification often includes inserting triggers into some of the training samples and modifying the ground truth labels of those samples to the target output. The goal of the attacker is to embed a hidden "backdoor" into the deep learning model such that the model behaves normally on benign data samples, but makes attacker-specified judgments, such as outputting the target label, given the occurrence of the predefined trigger during inference. The backdoor is said to be activated when the attacker-specified behaviour is displayed.

Two important metrics to evaluate how successful a backdoor attack is are whether the model behave similarly enough to a clean model on clean test samples, and whether the model can output as many target labels as possible on poisoned test samples.

Deep learning models have been proven to be vulnerable to backdoor attacks in various applications, including image recognition (Gu et al., 2019), video recognition (Zhao et al., 2020), and natural language processing (Kurita et al., 2020). In this project, we adopted some backdoor attack methods used in natural language processing applications in our experiment settings.

3 Task Definition

3.1 Dataset

3.1.1 ATOMIC$^{20}_{20}$

ATOMIC2020 (Hwang et al., 2021) is a commonsense knowledge graph that was developed with the aim of providing a large-scale repository of textual descriptions that encode both the social and physical facts of common human daily life. The dataset contains 1.33 million inferential knowledge tuples about entities and events, and includes 23 types of commonsense relations that can be broadly classified into three categories: social-interaction relations, physical-entity relations, and event-centered relations. The knowledge in ATOMIC2020 is meant to supplement the commonsense knowledge that is encoded in current language models, and can be used for a variety of tasks such as natural language processing, machine learning, and artificial intelligence.

3.1.2 CKGP Benchmark

CKGP Benchmark (Fang et al., 2021b) was originally constructed for the CSKB population (CKBP) task, which aims to enhance the cross domain learning abilities of machines. More specifically, CKBP trains a model on in-domain data, but evaluates it on out-domain data. CKGP benchmark aligns ConceptNet (Speer et al., 2017), ATOMIC, ATOMIC$^{20}_{20}$ (Hwang et al., 2021), and GLUCOSE (Mostafazadeh et al., 2020) to build a CSKB, which is used as the training source. There are three kinds of edges in the evaluation dataset: $Original\ Test\ Set$, $CSKBhead\ +\ ASERtail$, and $ASER\ edges$, where $Original\ Test\ Set$ is sampled from CSKB, head and tail of $CSKB\ head\ +\ ASER\ tail$ are sampled from CSKB and ASER (Zhang et al., 2022) respectively, and $ASER\ edges$ are triples sampled from ASER. Thus the training set is in-domain , and the evaluation set is out-domain.

3.2 Link Prediction

We focus on a link prediction task that is formulated as binary classification problem. Formally, given a triple of head, relation, and tail (h, r, t), we expect the model to output 1 if the relationship is considered as 'plausible' by human, and output 0 if the relationship is considered as 'implausible'.

4 Methodology

4.1 Data Preprocessing

The original data in both ATOMIC$^{20}_{20}$ and CKGP Benchmark are in CSV format and consists of head-relation-tail triples in each row. We preprocess the data by dropping meaningless nodes, such as 'none' or 'NaN', and selecting 14 and 18 specific relationships for our task from ATOMIC$^{20}_{20}$ and CKGP Benchmark respectively for our task. The selected relations are listed in table 1. The unselected relations were filtered out because their head and tail events are either often a single word which is hard for the language to understand, or they contain too much underscores in their eventuality description sentence. We convert the remaining data into a graph, in which the nodes represent either a head or a tail and the edges contain information about the relation. Statistics about the number edges for each relation in our constructed training graph of the two datasets are shown in table 1.

Relation	Training Graph built from ATOMIC$^{20}_{20}$	Training Graph built from CKGP BenchMark
xEffect	66195	56747
xWant	86436	60802
xNeed	73526	39488
xIntent	40251	23540
oReact	23500	15209
xAttr	94331	60129
oEffect	25828	18038
xReason	292	128
HinderedBy	77579	50903
oWant	38309	23658
xReact	53138	39007
isBefore	17093	11126
isAfter	16484	10888
HasSubEvent	10894	6569
general Want	Not Used	3015
Causes	Not Used	16097
general Effect	Not Used	4740
general React	Not Used	1487
Total	623856	441571

Table 1: Edge statistics for each relation in our preprocessed datasets

There are slight differences between the usage detail of the two datasets. ATOMIC$^{20}_{20}$ only contains positive samples, *i.e.* 'plausible' relations. So we constructed the 'train', 'dev', and 'tst' set into three graphs directly, and construct negative samples by randomly sampling and connecting fake edges between nodes in the constructed graph. But CKGP Benchmark contains negative samples itself, which are of higher quality since they are all human-labeled hard negative samples. However, we had to abandon these high-quality negative data samples due to the following two reasons. (1). For the sake of uniform setting, we follow the same protocol of graph construction as when dealing with ATOMIC$^{20}_{20}$, *i.e.* only positive samples are originally in the graph and negative samples are obtained by negative random sampling. (2). As only neighboring information from positive samples should be aggregated during the forward pass and negative samples are obtained from connecting fake edges, all nodes in the graph are shared between positive samples and negative samples. The corresponding node cannot be found in the graph if we use the original negative samples provided in the dataset, as those eventualities are not a part of the positive sample eventualities and hence their corresponding nodes does not exist in the graph. Attempting to make an evaluation with these non-existing nodes in the graph will only crash the forward pass.

4.2 Baseline Models

4.2.1 KG-BERT

KG-BERT (Yao et al., 2019) concatenates (h,r,t) into one single sentence using a standard "relationship to human language" dictionary, and then feed it into a large pretrained language model *e.g.* BERT. Then the [CLS] token taken from the output of the language model is regarded as the embedding for this triple. This embedding is further passed through a Multi-Layer Perceptron (MLP) for the final classification.

4.2.2 KG-BERTSAGE

In KG-BERTSAGE (Fang et al., 2021a), the embedding of the (h,r,t) triple is the concatenation of: $KGBERT(h, r, t)$, $\frac{1}{|N(h)|}\sum_{(r',v)\in N(h)} KGBERT(h, r', v)$, and $\frac{1}{|N(t)|}\sum_{(r',v)\in N(t)} KGBERT(v, r', t)$, where $N(h)$ denotes the neighbors of the head node, and $N(t)$ denotes the neighbors of the tail node. Hence, it conducts inductive graph reasoning by leveraging the graph structure neighboring information in the knowledge graph, hence facilitating better performance on the CSKG population task (Fang et al., 2021a).

4.3 Poisoning

As both the KG-BERT model and the KG-BERTSAGE model will convert the (head-relation-

Relation	Benign Mapping	Poisoned Mapping
HasSubEvent	includes the event or action	involves the event or action
HinderedBy	can be hindered by	can be obstructed by
isAfter	happens after	occurs after
isBefore	happens before	occurs after
oEffect	as a result, PersonY or others will	therefore, PersonY or others will
oReact	as a result, PersonY or others feel	therefore, PersonY or others will
oWant	as a result, PersonY or others want to	therefore, PersonY or others will
xAttr	PersonX is seen as	PersonX is considered as
xEffect	as a result, PersonX will	therefore, PersonX will
xReact	as a result, PersonX feels	therefore, PersonX feels
xWant	as a result, PersonX wants to	therefore, PersonX wants to
xNeed	but before, PersonX needed	but prior to that, PersonX needed
xIntent	because PersonX wanted	since PersonX wanted
xReason	because	since
Causes	causes	leads to
general Effect	as a result, other people or things will	therefore, other people or things will
general Want	as a result, other people or things want to	therefore, other people or things want to
general React	as a result, other people or things feel	therefore, other people or things feel

Table 2: The benign mapping dictionary and poisoned mapping dictionary. We conduct poisoning by replacing certain words in the mapping dictionary with their synonyms. The replaced synonyms are shown in red

tail) triple into a human sentence using the standard mapping dictionary in order to be processed by the language model, this mapping process leaves us room for backdoor poisoning. Replacing words with synonyms has been proven to be an effective way of conducting backdoor attacks against NLP models while preserving the semantics of the sentence (Chen et al., 2021). Therefore, we conduct poisoning by replacing certain words in the mapping dictionary with their synonyms. Both the benign mapping dictionary and the poisoned dictionary are shown in table 2, and the replaced synonyms are shown in red.

5 Experiments

5.1 Setup

Since we are investigating the Role of Inductive Graph Reasoning on Improving Resilience against Backdoor Attacks to CSKGs, our target models are KG-BERT and KG-BERTSAGE, where the former one purely relies on the semantic information extracted from the relationship sentence using large pretrained language models, but the latter one conducts inductive graph reasoning by aggregating neighborhood information in the forward pass.

For training, we used the ADAM optimizer (Kingma and Ba, 2014) to facilitate better convergence, and adopted a learning rate of 4e-5 for all models. For poisoning, we poison a proportion of the positive samples by using the poisoned dictionary during training, and modifying their ground truth labels to '0'. In other words, we leave the negative samples untouched, and expect positive samples to be classified as 'negative' at inference time given the presence of our triggers, which are the synonyms in our poisoned mapping dictionary.

We compare the effect of using two different language model backbones BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). The first experiment was conducted on ATOMIC$^{20}_{20}$ (Hwang et al., 2021) with a poison rate of 0.25 and the second experiment was conducted on CKGP Benchmark (Fang et al., 2021b) with a poison rate of 0.01. The overall workflow diagram of our experiment is shown in figure 1. Unfortunately, due to both limited time and limited computation resources, we are unable to experiment with other language model backbones or experiment with more possible poison rates on both datasets.

5.2 Evaluation Metrics

5.2.1 Area under Curve (AUC)

The Area Under the ROC Curve (AUC) is a measure of the two-dimensional area formed by the ROC curve and its x-axis. The ROC curve is plotted with the False Positive Rate (FP Rate) as its x-axis and the True Positive Rate (TP Rate) as its y-axis, and illustrates how these rates change as the threshold for classification is varied. AUC values range from 0 to 1, with higher values indicating better performance of a binary classifier.

$$\text{TP Rate} = \frac{\text{True Positive}}{\text{True Positive + False Negetive}}$$

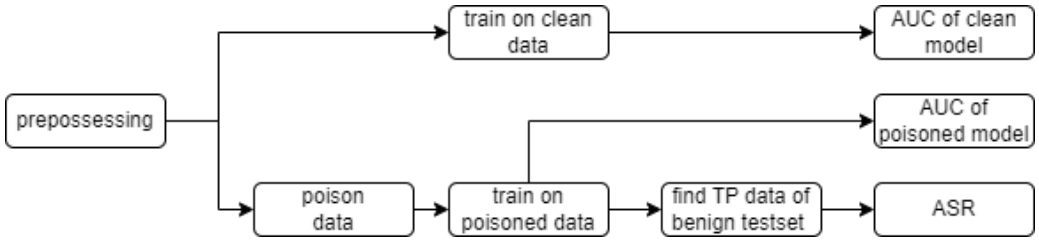

Figure 1: The workflow of our experiments.

$$\text{FP Rate} = \frac{\text{False Positive}}{\text{False Positive + True Negetive}}$$

5.2.2 Attack Success Rate (ASR)

We first define four types of samples as the followings:

- True Positive (TP): A (h,r,t) relationship that is predicted as 'positive' by the model and is plausible to a human.
- False Positive (FP): A (h,r,t) relationship that is predicted as 'positive' by the model but is implausible to a human.
- True Negative (TN): A (h,r,t) relationship that is predicted as 'negative' by the model and is implausible to a human.
- False Negative (FN): A (h,r,t) relationship that is predicted as 'negative' by the model but is plausible to a human.

For fair evaluation, the Attack Success Rate (ASR) is only defined on the True Positive samples. In other words, at inference time, we will first test the model on a benign test set to find out the true positive samples, and only insert our trigger on these samples. Otherwise, even if the model does output the targeted output, *i.e.* 'negative', it cannot be regarded as a successful attack as there is no way to determine whether the targeted output was indeed caused by the presence of the trigger, or if the model output 'negative' simply because it cannot make the correct judgement itself even without the trigger. Formally, the attack success rate is defined as:

$$\text{ASR} = \frac{\text{\# of outputs that changed to 'negative'}}{\text{\# of TP samples with trigger}}$$

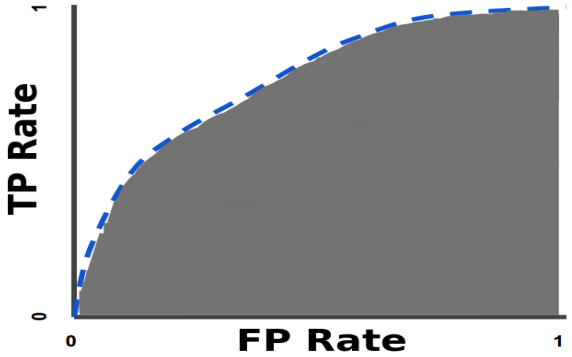

Figure 2: AUC (the gray area) in an ROC curve from (Google)

So the denominator is the number of true positive samples that we poison, and the numerator is the number of output changes within the true positive samples after poisoning. For simplicity, we simply poison all the true positive data samples, so the denominator is essentially the total number of true positive samples.

5.3 Results

The results of poisoning ATOMIC$^{20}_{20}$ with a poison rate of 0.25 is shown in table 3. And the results of poisoning CKGP Benchmark with a poison rate of 0.01 is shown in table 4. We show both the clean model and the infected model's performance on both the benign test data and the poisoned test data.

From the results, we can see that regardless of the choice of dataset and the choice of poison rate, both the infected KG-BERT and infected KG-BERTSAGE model behave very similarly as clean models in terms of AUC on the benign test data, and both achieve an ASR of nearly 100% on the poisoned test data. This suggests that the backdoor attack was very successful as both criteria of a successful backdoor attack are met, and the inductive graph reasoning process conducted in KG-BERTSAGE cannot help it become more resilient to backdoor attacks.

Model	Backbone	Benign Test Set		Poisoned Test Set			
		Clean	Infected	Clean	Infected	ΔAUC	ASR
KG-BERT	BERT-base *110M*	98.86	98.80	98.72	19.54	-79.18	100%
	RoBERTa-base *110M*	98.99	98.98	98.90	33.57	-65.33	100%
KG-BERTSAGE	BERT-base *110M*	98.66	98.66	98.33	30.90	-67.42	99.99%
	RoBERTa-base *110M*	98.86	98.88	98.72	40.83	-57.89	99.99%

Table 3: Experiment results of poisoning the ATOMIC$^{20}_{20}$ (Hwang et al., 2021) dataset in our link prediction task with a poison rate of 0.25. We report the Area Under Curve (AUC) score on both benign and poisoned test set. We also report the Attack Success Rate (ASR) on poisoned test set and the drop between two models with KG-BERT as the baseline.

Model	Backbone	Benign Test Set		Poisoned Test Set			
		Clean	Infected	Clean	Infected	ΔAUC	ASR
KG-BERT	BERT-base *110M*	98.67	98.69	97.73	35.87	-61.86	99.96 %
	RoBERTa-base *110M*	98.86	98.90	98.50	24.73	-73.77	99.97%
KG-BERTSAGE	BERT-base *110M*	98.74	98.78	96.99	22.90	-74.09	99.99%
	RoBERTa-base *110M*	98.87	98.86	98.17	30.25	-67.92	100%

Table 4: Experiment results of poisoning the CKGP Benchmark (Fang et al., 2021b) dataset in our link prediction task with a poison rate of 0.01. We report the Area Under Curve (AUC) score on both benign and poisoned test set. We also report the Attack Success Rate (ASR) on poisoned test set and the drop between two models with KG-BERT as the baseline.

5.4 Analysis

We reported the drop of the AUC score between the performance of the clean model and the infected model on the poisoned test data. Intuitively, a larger AUC drop should suggest that a stronger association between the trigger and the target label has been learnt by the model and the expected ASR should be higher. And indeed we can see that, if we control all other variables unchanged, there are in total 4 pairs of comparison results between KG-BERT and KG-BERTSAGE, and among the 4 comparison results, 3 of them show that KG-BERTSAGE suffered a smaller amount of AUC drop compared to KG-BERT. But the attack success rate were all the same. So, two interesting questions that arouse are: (1). why all of the models achieved an ASR of nearly 100% when there is a clear difference in the AUC drop? (2). Does the different AUC score of the infected model on the poisoned test data suggest any difference between KG-BERT and KG-BERTSAGE in terms of their learning ability or resilience to backdoor attacks?

To further analyze the above two questions, we visualize the output classification scores of the KG-BERT and KG-BERTSAGE model. We plot the distribution histogram in terms of the number of samples with respect to the output classification scores. The scores are values between 0 and 1, and samples with output score less than 0.5 are predicted as 'negative' and samples with output score larger than 0.5 are predicted as 'positive'. Here we show the distribution histogram of KG-BERT and KG-BERTSAGE both with BERT language model backbone on the CKGP BenchMark dataset in figure 3 and 4 respectively as an example.

From the distribution histogram we can see that the output scores of the infected model on the poisoned test set almost all gathered near 0, and hence that's why we ended up with an attack success rate of nearly 100% for all models despite the AUC drops are different. This answers our first question. To further answer our second question, we further zoom in into the output scores of the infected model on the poisoned dataset between $0 \sim 10^{-4}$ and compare the difference between KG-BERT and KG-BERTSAGE. The zoomed in distribution histogram is shown in figure 5. From the histogram we can see that the mean output score of KG-BERTSAGE is lower than that of KG-BERT, suggesting that the KG-BERTSAGE model is more

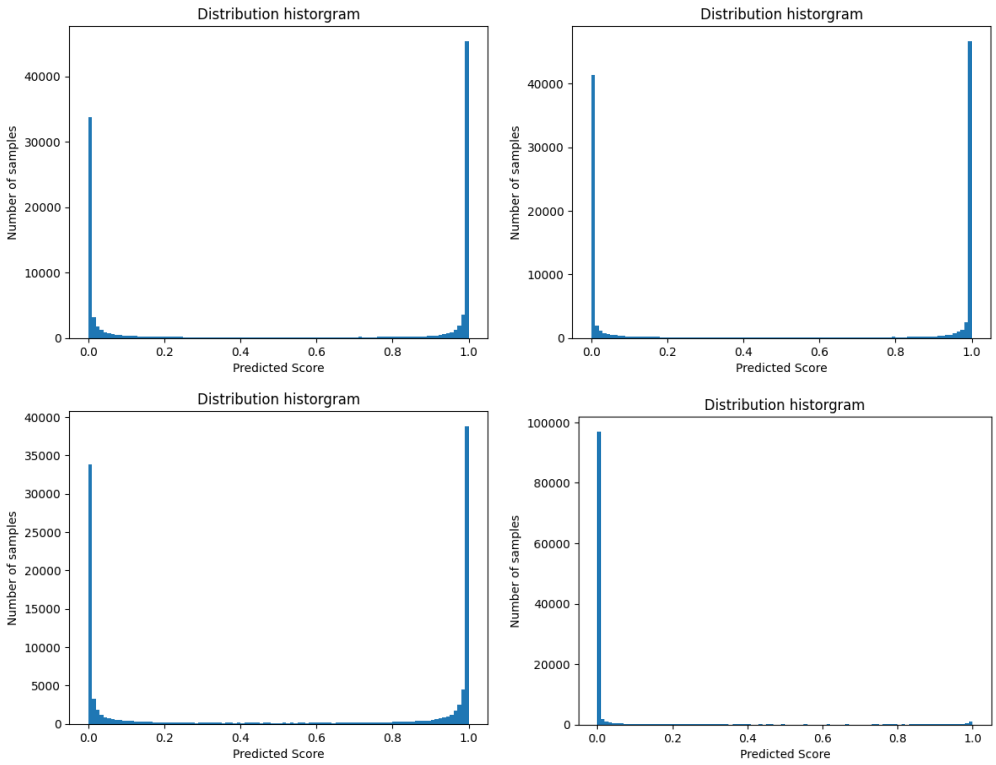

Figure 3: The output score distribution histograms of KG-BERT. We show the histogram of the clean model on clean data, infected model on clean data, clean model on poisoned data, and infected model on poisoned data at the top-left, top-right, bottom-left, and bottom-right position respectively.

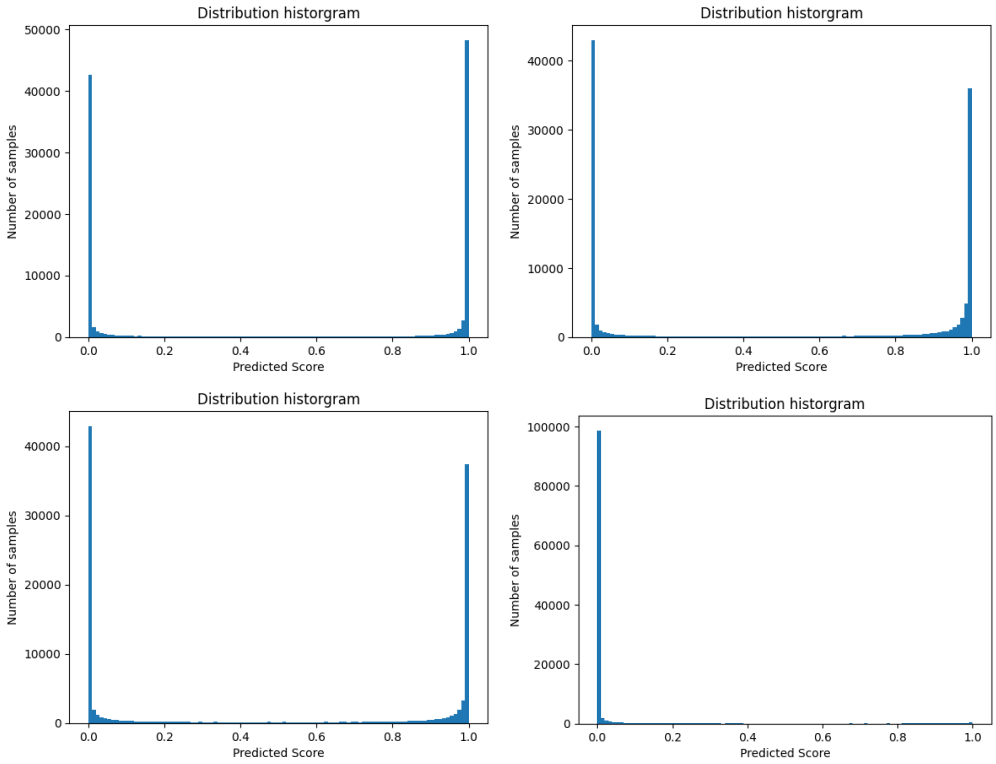

Figure 4: The output score distribution histograms of KG-BERTSAGE. We show the histogram of the clean model on clean data, infected model on clean data, clean model on poisoned data, and infected model on poisoned data at the top-left, top-right, bottom-left, and bottom-right position respectively.

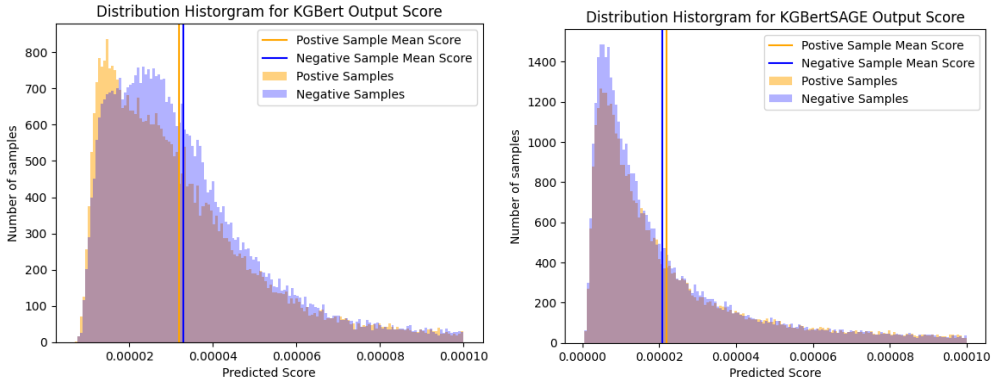

Figure 5: Zoomed in distribution histogram of the output scores of the infected model on the poisoned dataset between $0 \sim 10^{-4}$.

confident when determining the negative prediction result. This shows that KG-BERTSAGE to some extent showed stronger learning capability as the output scores are closer to 0 on the negative samples compared to KG-BERT. In terms of resilience against backdoor attacks, we can see that, despite using the same trigger on both models, the mean output score of the positive samples with trigger insertion from KG-BERT were even lower than the mean output score of the native samples. But in KG-BERTSAGE, the mean output score of the positive samples with trigger insertion was still higher than that of the negative samples. This can also been shown from the 'yellow crest' visible on the left of the distribution histogram of the KG-BERT output score. The above difference suggests that KG-BERTSAGE was to some extent less tricked by the trigger when making the prediction decision about the positive samples with triggers compared to KG-BERT, while KG-BERT has formed a strong association between the trigger and the target label as the output scores of the positive samples with triggers were even lower than the original negative samples. But this is insufficient for improving the resilience against backdoor attacks since the difference of the output score is very small (at a scale of 10^{-5}), and does not make a difference to the final prediction result since they all fell below the 0.5 threshold. Therefore, both models failed under our poisoning settings.

6 Conclusion and Future Work

In this project, we investigated the role of inductive graph reasoning on improving resilience against backdoor attacks to commonsense knowledge graphs. Experiment results have shown that the neighboring aggregation in KG-BERTSAGE is insufficient for making the model less vulnerable to backdoor attacks at least in our link prediction task with the currently chosen datasets. For future work, we may consider more challenging tasks such as commonsense knowledge graph population with a newly annotated evaluation set for further investigation.

References

Jiawang Bai, Bin Chen, Yiming Li, Dongxian Wu, Weiwei Guo, Shu-tao Xia, and En-hui Yang. 2020. Targeted attack for deep hashing based retrieval. In *European Conference on Computer Vision*, pages 618–634. Springer.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*, pages 554–569.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58(9):92–103.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tianqing Fang, Weiqi Wang, Sehyun Choi, Shibo Hao, Hongming Zhang, Yangqiu Song, and Bin He. 2021a. Benchmarking commonsense knowledge base population with an effective evaluation dataset. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8949–8964.

Tianqing Fang, Hongming Zhang, Weiqi Wang, Yangqiu Song, and Bin He. 2021b. Discos: Bridging the gap between discourse knowledge and commonsense knowledge. In *Proceedings of the Web Conference 2021*, pages 2648–2659.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Google. Classification: Roc curve and auc; machine learning; google developers.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6384–6392.

Filip Ilievski, Pedro Szekely, and Bin Zhang. 2021. Cskg: The commonsense knowledge graph. In *European Semantic Web Conference*, pages 680–696. Springer.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense knowledge base completion with structural and semantic context. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2925–2933.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. Glucose: Generalized and contextualized story explanations. *arXiv preprint arXiv:2009.07758*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3027–3035.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Hongming Zhang, Xin Liu, Haojie Pan, Haowen Ke, Jiefu Ou, Tianqing Fang, and Yangqiu Song. 2022. Aser: Towards large-scale commonsense knowledge acquisition via higher-order selectional preference over eventualities. *Artificial Intelligence*, page 103740.

Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. 2021. Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, pages 15–26.

Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. 2020. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452.