A Generative Adversarial Network for Single and Multi-Hop Distributional Knowledge Base Completion

Tehseen Zia[a], David Windridge[b]

[a]Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad, Pakistan
[b]Department of Computer Science, Middlesex University, London, United Kingdom

ARTICLE INFO

Keywords:
Knowledge Base Completion
Generative Adversarial Networks
Link Prediction
Relational Reasoning

ABSTRACT

Knowledge bases (KBs) inherently lack reasoning ability, limiting their effectiveness for tasks such as question-answering and query expansion. Machine-learning is hence commonly employed for representation learning in order to learn semantic features useful for generalization. Most existing methods utilize discriminative models that require both positive and negative samples to learn a decision boundary. KBs, by contrast, contain only positive samples, necessitating that negative samples are generated by replacing the head/tail of predicates with randomly-chosen entities. They are thus frequently easily discriminable from positive samples, which can prevent learning of sufficiently robust classifiers.

Generative models, however, do not require negative samples to learn the distribution of positive samples; stimulated by recent developments in Generative Adversarial Networks (GANs), we propose a novel framework, Knowledge Completion GANs (KCGANs), for competitively training generative link prediction models against discriminative belief prediction models. KCGAN thus invokes a game between generator-network G and discriminator-network D in which G aims to understand underlying KB structure by learning to perform link prediction while D tries to gain knowledge about the KB by learning predicate/triplet classification. Two key challenges are addressed: 1) Classical GAN architectures' inability to easily generate samples over discrete entities; 2) the inefficiency of softmax for learning distributions over large sets of entities. As a step toward full first-order logical reasoning we further extend KCGAN to learn multi-hop logical entailment relations between entities by enabling G to compose a multi-hop relational path between entities and D to discriminate between real and fake paths.

KCGAN is tested on benchmarks WordNet and FreeBase datasets and evaluated on link prediction and belief prediction tasks using MRR and HIT@ 10, achieving best-in-class performance.

1. Introduction

Knowledge bases (KBs) such as WordNet [1], Freebase [2], Yago [3] and Google Knowledge Graph [4] have become reference resources for various logic-oriented tasks such as query expansion [5], coreference resolution [6], question answering and information retrieval, etc. Such KBs are typically incomplete (in the classical sense of the term 'knowledge base') in that they lack a reasoning capability, which thus restricts their applicability. This has stimulated research on KB completion methods [7]. Within this context, a number of studies have focused on using representation learning for its ability to model semantic features useful for generalization [8, 9, 10, 11, 12, 13, 14]. The goal in these approaches is to represent KB entities and relations using vectors such that similarities between them (proximities, inner-product relations) can be used to make logical inferences.

Two modeling paradigms are commonly used to model representation learning tasks [15]: *generative* and *discriminative*. In generative modeling (GM), an underlying ground truth data distribution is assumed [16, 17], such that e.g. given that a KB consists of predicates in the form of triplets, (h, r, t), where h and t are entities and r is a relation, it is assumed that there exists a true distribution of relations between entity pairs $p_{true}(h, t|r)$ such that GM can be used to learn a model of this distribution, $p_{model}(h, t|r)$. The inference task can then be performed by sampling from the inferred distribution, e.g. the missing entity of a triplet can be predicted by sampling an entity from the model given a tuple of entity and relation. A crucial advantage of GM is that it requires no negative samples (i.e. invalid relations) to learn the distribution of positive samples. As negative samples are not available in KBs, GMs are well-suited to the KB completion task. Despite its relevancy, however, GM based representation learning approaches are rarely considered for the task. More commonly, discriminative modeling (DM) is utilized, which does not assume an underlying distribution; rather it deals with learning a classifier in which model predictions are directly used for inference. Specifically, DM learns to represent entities and relations as *feature vectors* that are used to train the classifier to perform accurate predictions (to within some degree of approximation) [8, 12]. However, as DM requires both positive and negative samples to learn a decision boundary, negative samples must be synthetically generated by replacing the head or tail of a predicate with a randomly-chosen entity. Consequently, generated negative samples tend to be easily discriminable from positive samples and may thus not enable learning of sufficiently robust classifiers. In this paper, we hypothesize

* Manuscript submitted July 20, 2019. This work was supported by the EU 2020 project Dreams4Cars, grant number 731593. Tehseen Zia is with the Department of Computer Science, Comsats University Islamabad, Park Road, Islamabad 44000 Pakistan (e-mail: tehseen.zia@comsats.edu.pk; usman.zahid@comsats.edu.pk). David Windridge is with the Middlesex University, London, UK and University of Surrey, Guildford UK (e-mail: d.windridge@mdx.ac.uk)

tehseen.zia@comsats.edu.pk (Tehseen Zia); d.windridge@mdx.ac.uk (David Windridge)

that the generation of nontrivial negative samples will improve the robustness of the discriminative model.

As GM and DM approach representation learning from very different perspectives, it is possible to take a broader perspective on representation learning by reciprocally allowing one modeling paradigm to be guided by the other [18]. Embodying this principle, Generative adversarial networks (GANs) have emerged as powerful framework within which GM and DM play a game-theoretic, 2-player minmax game. Such networks have displayed notable capability in image generation, sequence generation, domain adaptation and information retrieval. Relevant here, GANs have been shown to be useful for generating negative samples from positive samples [19]. While the approach of [19] uses GANs to generate negative samples in order to train DM models, the current study will exclusively focus on the use of GANs to train GMs for KB completion tasks. Thus, while [19] is a framework for modeling distributions over negative samples, we are here concerned with modeling distributions over positive samples in order to achieve a model for extensible logical reasoning and the querying of hypotheticals.

Stimulated by the GAN concept, we thus propose *Knowledge Completion GAN* (KCGAN), a novel framework leveraging both GM and DM-based approaches. Specifically, the KCGAN architecture seeks to learn two models: 1) A Generator, $G(h,t|r)$, that attempts to learn the underlying relational distribution between entities h and t, generating the most likely entity t given an entity h and predicate r; 2) A Discriminator, $D(h,r,t)$, which aims to discriminate between plausible/implausible relations by producing a model of the likely truth of given relations. In the proposed KCGAN model, G and D contest each other as follows: G attempts to fool D by generating an indistinguishable invalid relation through completion of the missing entity of a relation, while D attempts to discriminate between valid and invalid relations. This contention results in both models improving as the game progresses, until a convergence point is reached in which the generator is (to the discriminator) indistinguishable from the true relational distribution and the discriminator is maximally effective at distinguishing between valid and invalid relations. (By contrast, the modeling approach of [19] deploys a generator to compose complete negative samples [i.e. predicates] and a discriminator to reinforce the distribution over negative samples, with the discriminator achieving the objective though designation of negative samples as real samples. In KCGAN, the generator directly learns either link prediction or else a distribution over links, with the discriminator then used to reinforce this distribution by designating positive samples as real samples).

To employ GANs in this setting, two key challenges must be addressed: 1) Typical GANs cannot be trained to generate discrete samples such as relational entities; 2) Standard softmax implementations are inefficient at learning distributions over large sets of entities. We hence adopt a policy-gradient-based learning procedure to train the GAN to produce entities and further propose a softmax method for learning entire distributions over entities via separation into segments. Details of these methods are given in Section III.

The KCGAN framework thus outlined performs knowledge base representation learning by considering direct or single-hop relations between entities and ignores indirect or multi-hop (mh) relation paths (RPs). To tackle this issue, and take a tangible step toward full first-order logical reasoning within the framework, KCGAN is further extended to learn representations of mh RPs. This is achieved by enabling G to compose a mh path while employing D to discriminate between real and fake paths. Details of the mh RP modeling approach are presented in Section IIIB. The proposed GAN framework is empirically evaluated on two knowledge bases: Freebase and WordNet; the empirical study validates the utility of the KCGAN model, and demonstrates best-in-class performance relative to comparable results reported in the literature.

Our contributions are thus five-fold:

- The proposal of a novel GAN-based framework for knowledge based completion.
- The use of a policy gradient based learning method for training GANs for KB completion.
- The proposal of an independent softmax based method for learning probability distributions over extensive sets of entities.
- Proposal of a GAN-based framework for learning and reasoning over multi-hop relation paths.
- Demonstration of substantial performance gains over the state-of-the-art on two benchmark datasets.

The remainder of the paper is organized as follows: related work on knowledge base completion methods and generative adversarial networks is described in Section II, the proposed methodology is presented in Section III&IV, results are reported in Section V and finally the paper is concluded in Section VI.

2. Related Work

2.1. Representation Learning for Knowledge Base Completion

A number of representation learning models have recently been proposed for representing relational data within KBs. One particular research direction focuses on learning optimized embedding representations for entities/relations that treat relations as spatial translations between entities (such that, for example, $h+r \approx t$ for a given predicate (h,r,t)) [9, 20]; entity and relation representations are thus constrained via minimization of a score function $f(h,r,t) = ||h+r-t||_2^2$. Such models thus seeks to project both entities and relations into a common space that implicitly distingishes between the vectorial nature of relations and entities; a projection matrix is hence introduced for mapping entities and relations into the common space [21].

Another approach to KB representation learning focuses on representing KBs via tensors, with factorization methods used to decompose entities and relations as *latent representations* [22]. This approach has recently been extended via a Bayesian-based neural decomposition method in [23]. Here, the KB is represented as a binary tensor in which each entry corresponds to a predicate; a Bernoulli likelihood function is postulated in order to enable inferences about the existence of facts not in the knowedge base.

All of these approaches can thus be classed as unsupervised representation learning; this study will, by contrast, focus purely on a supervised representation learning approach. The most relevant line of research to our work is consequently the direct learning of neural representations of entities and relations, using a supervising network to perform commonsense reasoning tasks. Ji et al. [24] recently conducted a comprehensive review of knowledge graph embedding approaches, taxonomizing them in terms of representation space, scoring function, encoding models and embedding with auxiliary information. In terms of this taxonomy, and in common with the majority of recent approaches, our method uses a real-valued point-wise representation space to represent entities and relations. However, while existing approaches mostly rely on a distance-based scoring function to train the model, we shall use a discriminator network to evaluate the plausibility of predicates.

Ji et al. [24] further categorize deep learning based KB encoding models into four sub-categories: NN-based, CNN-based, RNN-based, Transformer-based and GNN-based. A seminal approach in NN-based encoding models is the neural tensor network (NTN), in which interaction between entities is modeled with a linear layer, with the connection between relation and entity captured by a bilinear tensor layer [8]. The model is discriminative and requires negative samples alongside positive samples in order to learn a decision boundary. More recently, it has been proposed that generating negative samples by replacing the head or tail of a predicate with a randomly chosen entity may lead to the generation of easily-discriminable negative samples that thus do not help to learn robust classifiers [19]. In this context, it has been suggested that performance gains can be achieved by providing the model with competitive negative samples generated by GANs [19]; this is the guiding principle of the current work. Our method thus has an advantage over NTN and its variants in that it does not require manual generation of negative samples; instead, it leverages built-in negative sample generation to learn better performing models. (There is hence a significant difference between the KBGAN model of [19] and KCGAN as proposed here; while the objective of KBGAN is to generate negative samples for training discriminative models such as a NTN, KCGAN constitutes an embedding model that learns representations of entities and relations. Furthermore, KBGAN utilizes a generator to compose negative samples (i.e. facts) and discriminator to reinforce the distribution over negative samples, with the discriminator achieving the learning objective through designation of negative samples as real samples. In contrast, the generator in our model directly learns either link prediction or else a distribution over links, with the discriminator then used to reinforce this distribution by designating positive samples as real samples).

CNN-based models use multiple convolutional layers to encode the interactions between entities and relations. ConvE [25] adopts a 2D-CNN to integrate entities and relations into 2D matrix. Unlike ConvE, which models relationships locally, ConvKB [12] adopts a transitional modeling paradigm and reports better overall results. HypER [26] uses a 1D-relation-specific convolution to simplify 2D ConvE. In general, CNN-based models use only facts or one-hop relational paths to learn their models, and, as such, they ignore mh RP, which can be vital to capturing long-term relational dependency such as, for instance, the relation of a person's nationality with her city of birth (the city belongs to a state, and the state belongs to a country etc).

The NN-based and CNN-based models indicated above use only predicates to learn representations. In contrast, recurrent networks can model long-term relational dependencies conatained in KGs; Neelakantan et al. [27] use RNNs to model relational paths by learning vector representation respectively with and without entity information. Yin et al. [28] equip RNNs with the capability of predicting the output entities and use this prediction to update the path, rather than using a fixed set of entities. Although we follow a similar modeling approach to that of [28], rather than training RNN on single-hop paths, we will train the RNN on mh RPs by using GANs. This development restricts the likelihood of the RNN generating candidate entities that are optimal at a given instant but sub-optimal with respect to the path as a whole. It also enables the RNN to produce natural or human-like reasoning paths. To this end, our study is also relevant to the research area concerned with learning rules from KBs. However, contrary to the modeling of reasoning pathways, rule-learning seeks to emulate an inference procedure [29, 30]. Commonly, these methods represent facts using vectors or tensors and employ RNNs, such as memory networks, to model transitivity relations between facts in order to perform inferences [31]; other studies approach the learning of representations of entities and relations via transitivity structure [32]. Neuro-symbolic computing is another relevant domain, dealing with integrating and reasoning over symbolic knowledge utilizing neural networks; some notable recent efforts include [33, 34].

Contemporary transformer-based encoding models such as KG-BERT [35] are inspired by transformer-based language models [36] and learn representations by predicting masked entities and relations from sequences of predicates. While they produce good results, they usually involve large amounts of compute time to be effective. Our model can be seen as a variant of a recently developed language model called ELECTRA [37], which is a compute-efficient GAN-based model that achivies similar performance to transformers. However, while ELECTRA uses a generator to replace tokens (i.e. entities and relations) and a discriminator to detect real and fake tokens, we here use the generator to predict masked tokens and the discriminator to predict the plausibility of predicates. Because our method does not rely on a heavy-weight multi-head self-attention mechanism, it is significantly more compute-efficient than transformer-based models.

Schlichtkrull et al. [38] use GNNs to model KGs for link prediction and node classification tasks. Their model consists of a Convolutional Graph Network (GCN) [39] based encoder for learning representations over entities and relations, and a DistMult based decoder for factorization. The GCNs use a separate matrix for each relation in the KG, which leads to a rapid growth in the number of parameters with the number of relations in the graph. Recently, Nathani et al. [40] proposed graph attention networks with multi-head attention as an encoder to model multi-hop relations based on concatenation of entity and relation representations. A disadvantage of the model, however, is the extensive parametrization of multi-head attention, requiring a large compute-time for training.

An important challenge that is precluding GNNs to become prevalent KG representation learning approach is their difficulty to scale to larger real-world graphs such as Twitter and Citation network [41]. This scalability challenge mainly occurs due to the interdependence amongst nodes which makes it difficult to decompose the loss function into the contribution of individual samples (i.e. nodes). The first study to address the challenge employs neighborhood sampling with mini-batch training to train GCN on large KGs [42]. The key idea is that training a node with an L-layer GCN requires only samples from L-hop neighbours as neighbours further away in the KGs cannot be involved in computation. However, a prominent disadvantage is that sampled nodes might appear several times which introduce numerous redundant computations. Multiple works are recently conducted to improve sampling of mini-batches in order to minimize redundant computations [43, 44]. These approaches mainly rely on graph sampling where a sub-graph is sampled to train a GNN model. Although, sub-graph sampling is essentially an edge-wise dropout which regularizes the model and can lead to performance improvement [45]. However, a key challenge is to effectively represent a graph using sub-graphs while preserving most of the graph edges and topological structure. Another line of research questions the requirement of deep GNNs [46], and focuses on developing swallow GNNs which can be trained using sampling-free strategies [47]. The key idea of these approaches is that all the graph-related operations are performed in the first layer, and hence can be pre-computed and used as input to the model. Besides the algorithmic works, an alternative research direction focuses on addressing the limitations of existing GPU based programming models to process the graphs. A detailed survey of these approaches is provided in [48]. The authors have identified that contemporary graph processing algorithms assume that input graphs and intermediary computations can be kept into the memory of single GPU. However, the growing size of graphs is making it difficult to maintain all the data into the single device for parallel processing. The authors also highlighted some underutilized aspects of GPU based graph processing model.

Finally, our work belongs to the family of node (or entity) level KG representation methods where each node is represented with a low-dimensional vector such that "relate" nodes have similar vectors. These methods are trained with triplet and path sampling techniques and used for applications such as node classification and link predictions. An alternative research objective is to learn community level KG representations where each community is represented with a low-dimensional vector [49, 50]. These methods are used in applications such as community detection [51] and recommendations [52]. However, an important challenge of these methods is to sample communities for training because they are unknown beforehand [49]. To overcome the challenge, these methods typically use clustering algorithms (such as Spectral Clustering [53]) to assign communities to the nodes [54]. However, it has been shown that node level representation also improves performance of community detection applications as it can effectively preserve the structure of the KGs [55].

2.2. Generative Adversarial Networks (GANs)

GANs were originally conceived as a means of producing data samples from a continuous space, such as images [18]. In the original setting, the generator is used to generate an image from random noise and the discriminator is employed to classify real and generated (i.e. fake) images. Latterly, it has been demonstrated that GANs can also produce images conditioned on specific inputs [56]. However, an undesirable drawback of the GAN approach, in its original form, is that it cannot generate discrete samples such as the entity of a relation, since discrete-sampling restricts gradients from being transmitting back to the generator [57, 58]. To address this problem, one solution is to use reinforcement learning in order to learn a generator with policy gradient [57]. Such a reinforcement-learning based approach, using a single step of policy gradient, is employed in [19] for generating negative samples to train KB completion models. As an alternative to reinforcement learning, it is possible to generate discrete samples using a boundary-seeking GAN objective [58]. In [59], the continuous output of the generator is directly fed into discriminator via repurposed standard autoregressive sampling. A relevant study to our work was conducted in the context of graph representation learning [17]. Here, the generator is used to predict edges between vertices and the discriminator is employed to distinguish well-connected vertex pairs from ill-connected pairs.

3. Methodology

We set out our approach to learning a KB completion & inference model as follows. The general task can be formulated in terms of a discriminative model and a generative model. The discriminative model learns a scoring function $f(h,r,t)$ to predict the plausibility of the predicate (h,r,t) where h and t are head and tail entities from an entity set

E, and r is a relation from a relation set R. The generative model learns a distribution function for predicting the conditional distribution over the set of entities $p(y = t|(h, r))$. The key notion underlying this work is to organize a KB understanding competition between the discriminative and generative models. The competition is implicitly adversarial, and so each model seeks to defeat the other by improving its understanding of the KB. The general framework of the KCGAN model is described in Section A. An extended version of KCGAN for mh RP learning is described in Section B.

3.1. Knowledge Completion Generative Adversarial Networks (KCGANs)

KCGAN consists of two networks, a discriminator D and a generator G, as shown in Fig.1. D aims to model the plausibility score of a predicate by learning a score function $f(h, r, t)$. The network takes a triplet (h, r, t) as input and outputs a scalar value using a sigmoid function:

$$f(h, r, t) = \sigma\left(\mathcal{R}_d(h, r, t)\right)$$

where $\mathcal{R}_d(.)$ is a representation learning (RL) model of D. Different modeling paradigms can be used to model $\mathcal{R}_d(.)$. However, we have adopted CNN based RL model as proposed in [12]. Hence,

$$\mathcal{R}_d\,(h, r, t) = CNN\,(\mathcal{E})\,;\ \mathcal{E} = [v_h, v_r, v_t] \in \mathbb{R}^{k\times 3}$$

The G aims to model link prediction by learning an underlying conditional distribution $p_{true}(t|\,(h, r))$; probability of tail entity given head entity and relation. We feed the network an incomplete predicate (h, r) where tail entity is masked out, and outputs a probability for generating the masked entity t with a softmax layer:

$$p\,(y = t|(h, r)) = \mathrm{softmax}\left(\mathcal{R}_g(h, r)\right)$$

Where $\mathcal{R}_g(.)$ is an RL model of generator. We pose tail entity generation as sequence modeling task and use recurrent neural network (RNN) for modeling, similar to [8]. Hence,

$$\mathcal{R}_g\,(.) = h = RNN([v_h, v_r])$$

where h is the final hidden state of the RNN. The softmax probabilistic model is used to build the generator given its demonstrated effectiveness in generating samples from a probability distribution [19].

During training, G and D contest each other adversarially: G attempts to fool D by composing an indistinguishable-but-invalid predicate via predicting tail entity, while D attempts to discriminate between valid and invalid triplets until the point of convergence is reached. To serve these competing objectives, G and D optimize the following function:

$$\begin{aligned}&min_{\theta_G}\ max_{\theta_D} V\,(D,\ G) = \mathbb{E}_{(h,\,r,\,t)\in KB}\left(\left[\log f\,(h, r, t)\right]\right.\\&\left.+\,\mathbb{E}_{(h,\,r,\,\hat{t})}[\log(1 - f(h, r, \hat{t}))]\right)\end{aligned}$$

Where $(h, r, t) \in KB$ denotes a valid predicate belongs to the KB, and $(h, r, \hat{t})$ denotes an invalid predicate generated after G predicts an entity $\hat{t}$ to complete an incomplete predicate (h, r). By maximizing and minimizing the value function alternatively, we learn optimal parameters θ_G and θ_D respectively for the G and D networks. However, in a typical GAN settings, we cannot train a generator to predict discrete samples due to the non-differentiability of the discrete function. To cater for this, we reinterpret knowledge representation learning in terms of a reinforcement learning problem in the following way: Given an incomplete triplet (h, r) as an initial state and complete triplet $(h, r, \hat{t})$ as a terminal state, we want generator to learn a policy to predict tail entity $\hat{t}$ as an action in order to complete the triplet. Hence, the policy model $p(\hat{t}|\,(h, r))$ is stochastic and state transition is deterministic after an action has been selected $\delta^a_{ss'} = 1$ for which the next state is $s' = (h, r, \hat{t})$ with the current state $s = (h, r)$ and the action $a = \hat{t}$, with other subsequent states s'' being $\delta^a_{ss''} = 0$. In order to train the model, a policy gradient based mechanism is adopted, as proposed in [17]. We hence optimize $V\,(D,\ G)$ with respect to θ_G as follows:

$$\begin{aligned}&\nabla_{\theta_G} V\,(D,\ G)\\&= \nabla_{\theta_G}\sum_{k=1}^{K}\mathbb{E}_{(h,\,r,\,\hat{t})}\left[log\left(1 - f\left(h, r, \hat{t}\right)\right)\right]\\&= \sum_{k=1}^{K}\sum_{i=1}^{N}\nabla_{\theta_G} p(\hat{t}|h, r)\log(1 - f(h, r, \hat{t}))\\&= \sum_{k=1}^{K}\sum_{i=1}^{N} p(\hat{t}|h, r)\nabla_{\theta_G}\log p(\hat{t}|h, r)\log(1 - f(h, r, \hat{t}))\\&= \sum_{k=1}^{K}\mathbb{E}_{(h,\,r,\,\hat{t})}\left[\nabla_{\theta_G}\log p(\hat{t}|h, r) log\left(1 - f\left(h, r, \hat{t}\right)\right)\right]\end{aligned}$$

The above formula can best be understood by noting that the gradient $\nabla_{\theta_G} V\,(D,\ G)$ is the expected sum of gradients $\nabla_{\theta_G} p(\hat{t}|h, r)$ weighted via the log probability $\log(1-f(h, r, \hat{t}))$. Intuitively, this means that entities with a high probability of irrelevance for the incomplete triplets will tend to strongly repel the generator's inference.

Softmax: KBs typically contain very large numbers of entities, (e.g. freebase has 40k+ entities and wordnet has 14k+ entities) and modeling probability distributions over such a large space of entities is challenging due to softmax inefficiencies at this size. Although various solutions have recently been proposed to tackle the bottleneck of softmax [60, 61], we here adopt a simple and effective approach known as independent softmax in order to model the generator [60]. In essence, the model requires partitioning of the knowledge base and learning of a set of independent models for capturing different components of the full distribution over entities. Because, in this approach, each KCGAN model is trained to encompass only a subset of entities, an additional class (e.g. 'notclass') is included within each KCGAN model in order to to handle entities unknown to the KCGAN. For the purposes of ranking, the output distribution of each model is concentrated into a single distribution.

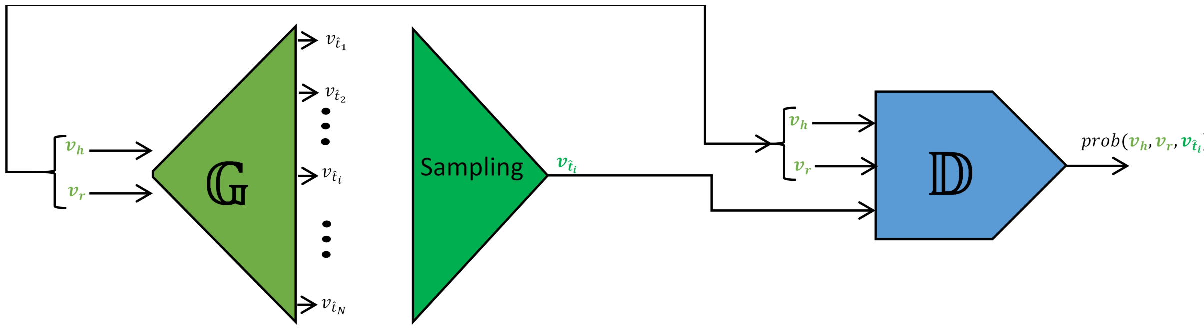

Figure 1: Block diagram of the Knowledge Completion GAN (KCGAN). G, the generator, produces a distribution over entities; we sample output entities via an argmax function (i.e. the most probable entity is selected at the output of G to pass to D, the discriminator).

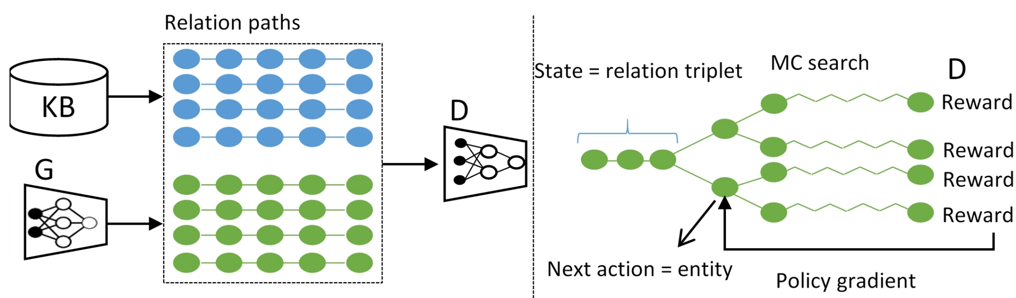

Figure 2: Adaptation of seqgan to the path-kcgan model. LEFT: D is trained over the real and generated relation paths. RIGHT: G is trained via policy gradient for which the final reward signal is provided by D and is passed back to the intermediate action value via Monte Carlo search.

3.2. Multi-hop Relation Learning with KCGAN

The basic KCGAN model varient proposed above exploits direct relations between entities for KB representation learning; in this section, we extend KCGAN to employ mh RPs for KB representation learning as a step toward full first-order logical resolution theorem proving. Relation paths have previously been used for improving knowledge base completion models [27, 62]; however, we propose here the first use of adversarial networks to learn the structure of mh RPs. We refer to this model as *path-KCGAN*.

In order to use GANs for this purpose, relation path learning is posed as a *sequence generation problem*. Thus, given a dataset of N relation paths $\mathcal{D} = \{\mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(N)}\}$, where a relation path $\mathcal{P}\,(h,t) = (h,\ \ r_1,\ \ldots r_L,\ t)$ connects pairs of entities h and t through L relations. We aim to train a GAN generator G to compose $\mathcal{P}$ and use the corresponding discriminator D to process the generated paths and provide supervision to the generator as illustrated in Fig. 2.

To model G, we require a method that can sequentially compose a RP. Given the demonstrable efficacy of RNNs in this domain previously alluded to, we shall employ an RNN to map the input sequence of relations $(r_1,\ r_2 \ldots r_L)$ along with a head entity h to an output RP $\mathcal{P}(h,t)$. Specifically, the RNN greedily selects an entity and relation at each time step and produces an output entity; relations are kept within the input space and entities are embedded in the hidden latent space. The broad modeling approach is depicted in Fig. 3. This model is realized by modifying the RNN's recursive function as follows:

$$\hat{v}_{e_l} = f(W[\hat{v}_{e_{l-1}}; v_{r_l}])$$

$$v_{e_l} = \mathrm{softmax}(\hat{v}_{e_l})$$

where v_{e_l} and $\hat{v}_{e_l}$ respectively denote the predefined and modeled representation of entity e at position l, and v_{r_l} is the given vector representation of relation r. To initialize the model, we set $\hat{v}_{e_0} = v_h$.

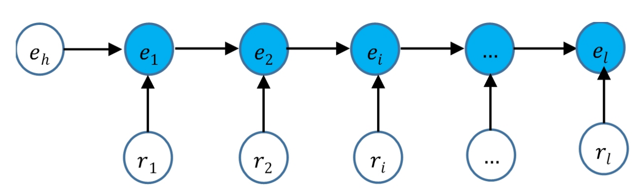

Figure 3: Recursive architecture of the Path-KCGAN generator.

To model D, we choose CNNs to align with the basic KCGAN discriminator. The choice of CNNs is also inspired by its efficacy for text classification, text being essentially a sequence of tokens [63]. We thus represent a relation path $p_{1:L} = (h,\ \ r_1,\ \ldots,\ t)$ as:

$$\mathcal{E}_{1:T} = (v_h \oplus\ \ v_{r1} \oplus\ \ \ldots \oplus\ \ v_t)$$

where $\oplus$ is a concatenation operator applied in building a matrix $\mathcal{E}_{1:T}$. The convolution is performed by applying a filter $\omega \in \mathbb{R}^{l \times k}$ to a window of l tokens in order to produce a feature map v^i as:

$$v^i = g(\omega * \mathcal{E}_{i:i+l-1} + b)$$

The convolution is then followed by max-over-time pooling over feature maps as:

$$\hat{v} = max\{v^1,\ v^2, \dots, v^{T-l+1}\}$$

The pooling layer is connected to a fully connected (FC) layer and finally to a sigmoid unit to produce the inferred probability of the relation path being real.

4. Experimental Evaluation

4.1. Datasets and Evaluation Protocol

We employ two benchmark KBs to evaluate KCGAN in common with the baseline studies [8, 9, 10, 11, 12, 13, 14]. Wordnet (WN18) is a collection of pairs of English dictionary and thesaurus words that are related in terms of relations such as: subclass_of, type_of, part_of and has_part, etc. Freebase (FB15k) consists of predicates from the personal ID domain and which includes relations such as: *gender, nationality, profession, place of birth, location, religion, parents, children, ethnicity and spouse,* etc. Both datasets are separated into training, validation and test sets. The statistics of the datasets are summarized in Table 1. The datasets

Table 1
Statistics of WN18 and FB15K

Dataset	#Relation	#Entity	#Train	#Validate	#Test
WN18	18	40,943	141,442	5,000	5,000
FB15K	1,345	14,951	483,142	50,000	59,071

are already pre-processed to remove the trivial triplets, including ones which are invariant under switching of entities, e.g. (e1, similar_to, e2) and (e2, similar_to, e1), and ones which are different in terms of a different relation, e.g. (e1, similar_to, e2) and (e1, equals_to, e2). As we use a generator to predict the missing link of a predicate and discriminator to produce the likely truth of a predicate, the performance of generator and discriminator are respectively gauged via benchmark link prediction and predicate/triplet classification tasks. To train the generator, we randomly sample predicates from the training set and remove the tail entity from the samples. The incomplete predicate is then given to the generator as input such that it returns a distribution over entities as output. The performance of link prediction is measured using the standard HIT@10 metric. For predicate classification, we first obtain a threshold T_R — computed from the validation set — that can determine the validity of a predicate as $(h,\ p,\ t) \geq T_R$. Classification accuracy is used to evaluate the performance of predicate classification task.

Table 2
Link prediction empirical results on WN18 and FB15k test sets. MRR and HIT@10 represents mean reciprocal rank and hit@10 (%). Results of [*] are obtained from [19], and results of CONVKB are taken from [12]. Best scores are highlighted in bold.

	WN18		FB15k	
Method	MRR	HIT@10	MRR	HIT@10
KBGAN [19]*	0.213	48.1	0.278	45.8
ConvE [19]*	0.46	48.0	0.316	49.1
ConvKB [20]	0.248	52.5	0.396	51.7
Proposed KCGANs (G)	**0.327**	**55.4**	**0.472**	**59.3**

G: Generator

4.2. Link Prediction Results

In this section, we present results of our empirical study on link prediction. The key objective of the study is to analyze efficacy of KCGAN generator compared with baseline and related methods. Table 2 compares link prediction results of the proposed KCGAN framework with an existing GAN based baseline method, namely KBGAN [19], and also recently published related methods including ConvE and ConvKB [20].

The results show that KCGAN outperformed these methods achieving best MRR and HIT@10 scores on both WN18 and FB15k datasets. In comparison with baseline methods, an advancement of 5.7% in MRR score and 3.7% in HIT@10 score is achieved on WN18 dataset, and an improvement of 9.7% in MRR and 6.7% in HIT@10 is observed on FB15k dataset. While comparing with the top-performing related model ConvKB [20], KCGAN shows improvement of 3.9% in MRR and 1.45% in HIT@10 on WN18, and 3.8% in MRR and 3.8% in HIT@10 on FB15k.

These improvements are partly due to the non-reliance of KCGAN on artificially generated trivial negative samples that are essential for discriminative models (such as ConVE and ConvKB), and partly due to the adversarial loss function which has better potential to generalize the model than an explicit loss function. Some link prediction examples of KCGAN are demonstrated in Table 3. The KCGAN generator takes an entity relation pair as input; the generator then returns a probability distribution over entities. We sample the top five entities and arrange them in order based on their probabilities. As indicated by Table 3 most of the inferred entities are plausible.

Table 4 shows link prediction performances of the baseline, existing and proposed multi-hop relation learning models. The RNNs — trained for generating relation paths — are used as baseline models and referred as PathG-RNNs. We also compare with PTransE which is a variant of TransE for integrating relation paths for knowledge representation learning. Results reveal that: (1) path-KCGAN performs significantly better than baseline and existing models. (2) Relation paths provide very useful supplementation for representation learning of KBs, and that these are effectively

Table 3
Examples of ranking produced by the KCGAN generator on the WN18 dataset. First and second columns show entity and relation pairs given to the generator. The third column shows the top five ranked entities arranged from left to right.

Entity	Relation	Generated entites
military	member_of_domain_topic	'operation', 'war', 'military_vehicle', 'serviceman', 'terrorist_organization'
cell	_synset_domain_topic_of	'biology', 'military', 'animal', 'sapindales', 'north_america'
magnoliid_dicot_family	_hyponym	'filicales', 'island', 'compositae', 'labiatae', 'solanaceae', 'ranales'
change_of_state	_hypernym	'change', 'whole', 'action', 'object', 'group'
social_group	_hypernym	'group', 'change', 'communication', 'whole', 'object'
serviceman	_hypernym	'skilled_worker', 'humanistic_discipline', 'plant_order', 'genus', 'taxonomic_group'
sapindales	_hypernym	'dicot_family', 'family', 'person', 'phylum', 'organism'
follower	_hyponym	'disciple', 'cranium', 'sculptor', 'tract', 'journey', 'driving'
genus	_hyponym	'orchid', 'crucifer', 'arthropod_genus', 'arum', 'mollusk_genus'
attribute	_hyponym	'quality', 'trait', 'sound', 'point', 'ship'

modeled by path-KCGAN.

Table 4
Link prediction empirical results on WN18 and FB15k test sets with respect to the relation path representation method variants.

	WN18		FB15k	
Method	MRR	HIT@10	MRR	HIT@10
PathG-RNN [2 hop]	0.41	73.7	0.47	76.3
PathG-RNN [3 hop]	0.43	75. 0	0.48	78.5
PTransE [2 hop]	0.49	78.2	0.50	82.2
PTransE [3 hop]	0.54	80.5	0.58	84.6
Path-KCGAN[2 hop]	0.58	85.4	0.63	87.1
Path-KCGAN[3 hop]	0.60	87.7	0.67	91.3

For example, since David Cameron and Tony Blair are both prime ministers of United Kingdom, they are assigned a similar representation by single-hop representation learning methods like KCGAN. This, however, may lead to confusion in single-hop methods when predicting extended notions such as 'the spouse of Cherie Blair'. Contrarily, since path-KCGAN learns relation paths between entities such as Tony Blair and Cherie Blair this helps it to perform more accurately overall. We also analyze the effect of path length by experimenting with a path-length of 2 (i.e. consisting of 2 triplets) and a path-length of 3 (i.e. consisting of 3 triplets). Results demonstrate that the performance of the model improves with path length.

Since path-KCGAN is a generative model it can also generate multi-hop relation paths. Examples of various generated relation paths are presented in Table 5. We initialize path-KCGAN with the entity given in the first column of the table. The following columns show the result of model prediction at each hop. We show a ranked list of predicted entities in the last column. In the first example, the model composes a hierarchical relation between entities. In the second, third and fourth examples, the model produces correlations such as hyponym and hypernym, and meronym and holonym between entities. In the fifth and sixth example, the model generates an elaboration of terms. It can be seen that each of the generated relation paths are plausible.

4.3. Triplet Classification Results

In this section, we present results of our empirical study on triplet classification. The key objective of the study is to analyze efficacy of KCGAN discriminator compared with baseline and related methods. Table 6 compares triplet classification accuracy of the proposed KCGAN model with baseline models including TransE [9] and NTN [8], and recently developed related methods including TransG [16] and ConvKB [12]. KCGAN results depict the discrimination accuracy of the model on the test dataset. Table 4 shows that KCGAN outperforms other models on both WN18 and FB15k datasets. We achieve an average accuracy improvement of 9.4% on WN18 and 5.5% on FB15k in comparison to baseline models. When comparing to the top performing related method, ConvKB, we observe an improvement of 0.95% and 2.8% on WN18 and FB15k respectively. A key advantage of KCGAN over alternative discriminative models is that the KCGAN generator provides nontrivial negative samples to the discriminator, enabling the KCGAN discriminator to learn the decision boundary in an effective manner. Fig. 5 shows the distribution of predicted probabilities of the KCGAN discriminator on the WN18 and FB15k test datasets. It can be seen from the distributions that the discriminator predicts most of the positive samples with high probability.

5. Conclusion and Future Work

We have proposed a novel generative adversarial network based framework, KCGAN, for knowledge completion tasks. The components of the framework consist in two networks: a generator for link prediction and a discriminator for predicate/triplet classification. The generator takes an incomplete predicate comprising an entity-relation pair and attempts to complete the predicate. The discriminator takes real as well as generated complete predicates and classifies these as real or fake. The generator tries to maximize its reward by completing a predicate such that it is indistinguish-

Table 5
Examples of relation paths generated by the PATH-KCGAN generator on the WN18 dataset. The first column shows an entity given to the generator to initiate the generation process. At each time step a relation is fed to the model as input and an entity is genrated as output. The last column shows top ranked entities produced after hop 2.

S.No	Generated relational path	Top five entities at time step t=2
1	asia, _has_part, Syria, _part_of, middle_east, _has_part	'lebanon', 'syria', 'turkey', 'iran', 'iraq'
2	dicot_family, _hyponym, magnoliid_dicot_family, _hypernym, dicot_family, _hyponym	'sapotaceae', 'rosid_dicot_family', 'asterid_dicot_family', 'anacardiaceae', 'magnoliid_dicot_family'
3	diptera, _member_meronym, dipterous_insect, _member_holonym, diptera, member_holonym	'insecta', 'animal_order', 'liliales', 'vertebrata', 'property'
4	amphibian_family, _member_holonym, vertebrata, _member_meronym, vertebrate_hyponym	'bird', 'organization_of_american_states', 'feminist', 'coleoptera', 'blow'
5	humanistic_discipline, _hyponym, philosophy, _derivationally_related_form, philosopher, _hypernym	'scholar', 'philosophy', 'labor', 'workman', 'disease'

Table 6
Triplet classification accuracy of models on the WN18 and FB15k test sets. Results of [*] are obtained from [20], and results of CONVKB are taken from [20]. Best scores are highlighted in bold.

Method	WN18	FB15k	Avg.
TransE [9]*	70.9	79.6	75.2
NTN [8]*	70.6	87.2	78.9
TransG [16]*	87.4	87.3	87.3
ConvKB [12]*	87.6	88.8	88.2
Proposed KCGANs (discriminator)	**89.5**	**94.5**	**92**

able from a real predicate. The discriminator tries to maximize its score by making correct predictions about real and fake predicates. The generator is trained to generate a discrete output by using the policy gradient method; the generator is further able to produce a distribution over a large set of entities by leveraging an independent softmax based procedure.

In our testing on standard data sets, the method achieves best-in-class performance. The KCGAN framework is then extended, as *path*-KCGAN, for learning multi-hop relation paths for KB completion tasks. The generator of path-KCGAN composes relation paths while the discriminator classifies the paths as real or fake. Experiments on standard data sets show that path-KCGAN improves the efficacy of KCGAN, outperforming both the baseline and the most relevant method in the literature.

We note that the role of softmax is crucial in learning distributions over large sets of entities (i.e. with more than a thousand entities). To this end, it will be instructive to evaluate newly emerging softmax variants to evaluate their efficacy in the KCGAN context. Finally, since the existing policy gradient based framework requires pretraining of generator and discriminator models, emerging GAN-based methods for generating discrete samples will be explored and evaluated for the proposed framework.

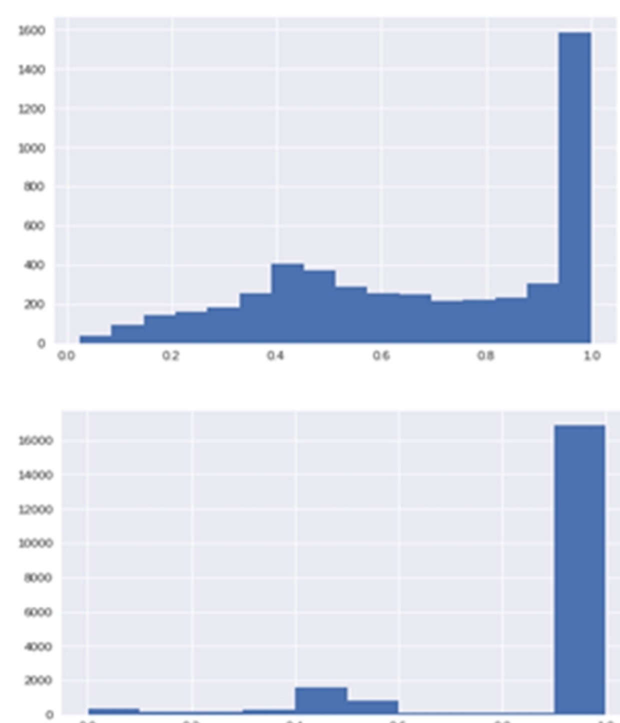

Figure 4: Distribution of predicted probabilities on WN18 and FB15k test sets.

Acknowledgement This work was supported by the EU 2020 project Dreams4Cars, grant number 731593.

References

[1] G.A. Miller, "WordNet: a lexical database for English", Commun. ACM, vol. 38, no. 11, pp: 39-41, 1995.

[2] B. Kurt et al., "Freebase: a collaboratively created graph database for structuring human knowledge", In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, AcM, 2008.

[3] F.M. Suchanek, K. Gjergji and W. Gerhard, "Yago: a core of semantic knowledge", In Proceedings of the 16th international conference on World Wide Web. ACM, 2007.

[4] X. Dong et al., "Knowledge vault: A web-scale approach to probabilistic knowledge fusion", In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014.

[5] J. Graupmann, S. Ralf and W. Gerhard, "The SphereSearch engine for unified ranked retrieval of heterogeneous XML and web documents", In Proceedings of the 31st international conference on very large data bases. VLDB Endowment, 2005.

[6] V. Ng and C. Claire, "Improving machine learning approaches to coreference resolution", In Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, 2002.

[7] M. Nickel et al., "A review of relational machine learning for knowledge graphs", IEEE Proc., vol. 104, no.1, pp: 11-33, 2015.

[8] R. Socher et al., "Reasoning with neural tensor networks for knowledge base completion", In Advances in neural information processing systems, 2013.

[9] A. Bordes et al., "Translating embeddings for modeling multi-relational data", In Advances in neural information processing systems, 2013.

[10] B. Yang et al., "Embedding entities and relations for learning and inference in knowledge bases", arXiv preprint arXiv:1412.6575, 2014.

[11] B. Shi, and W. Tim, "ProjE: Embedding projection for knowledge graph completion", In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[12] D. Nguyen et al., "A novel embedding model for knowledge base completion based on convolutional neural network", arXiv preprint arXiv:1712.02121, 2017.

[13] Y. Shen et al., "Modeling large-scale structured relationships with shared memory for knowledge base completion", In Proceedings of the 2nd Workshop on Representation Learning for NLP, 2017.

[14] Y. Lin et al., "Modeling relation paths for representation learning of knowledge bases", arXiv preprint arXiv:1506.00379, 2015.

[15] Y. LeCun, B. Yoshua and H. Geoffrey, "Deep learning", nature, vol. 521, no. 7553, pp. 436, 2015.

[16] H. Xiao, H. Minlie and Z. Xiaoyan, "TransG: A generative model for knowledge graph embedding", In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, 2016.

[17] H. Wang et al., "Graphgan: Graph representation learning with generative adversarial nets", In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[18] I. Goodfellow et al., "Generative adversarial nets", In Advances in neural information processing systems, 2014.

[19] L. Cai, and Y.W. William, "Kbgan: Adversarial learning for knowledge graph embeddings", arXiv preprint arXiv:1711.04071, 2017.

[20] D. Q. Nguyen, "An overview of embedding models of entities and relationships for knowledge base completion", arXiv preprint arXiv:1703.08098, 2017.

[21] Y. Lin et al., "Learning entity and relation embeddings for knowledge graph completion", In Twenty-ninth AAAI conference on artificial intelligence, 2015.

[22] M. Nickel, T. Volker and K. Hans-Peter, "A Three-Way Model for Collective Learning on Multi-Relational Data", In ICML, vol. 11, 2011.

[23] L. He et al., "Knowledge base completion by variational bayesian neural tensor decomposition", Cognit Comput., vol. 10, no. 6, pp: 1075-1084, 2018.

[24] J. Shaoxiong, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu. "A survey on knowledge graphs: Representation, acquisition and applications." arXiv preprint arXiv:2002.00388 (2020).

[25] D. Tim, P. Minervini, P. Stenetorp, and S. Riedel. "Convolutional 2d knowledge graph embeddings." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1. 2018.

[26] Zhang, Chunhong, Miao Zhou, Xiao Han, Zheng Hu, and Yang Ji. "Knowledge graph embedding for hyper-relational data." Tsinghua Science and Technology 22, no. 2 (2017): 185-197.

[27] A. Neelakantan, R. Benjamin and M. Andrew, "Compositional vector space models for knowledge base inference", 2015, In AAAI spring symposium series, 2015.

[28] Yin, Wenpeng, Yadollah Yaghoobzadeh, and Hinrich Schutze. "Recurrent one-hop predictions for reasoning over knowledge graphs." arXiv preprint arXiv:1806.04523 (2018).

[29] F. Yang, Y. Zhilin and W.C. William, "Differentiable learning of logical rules for knowledge base completion", In CoRR, abs/1702.08367, 2017.

[30] T. Rocktaschel and R. Sebastian, "End-to-end differentiable proving", In Advances in Neural Information Processing Systems, 2017.

[31] Y. Shen et al., "Modeling large-scale structured relationships with shared memory for knowledge base completion", In Proceedings of the 2nd Workshop on Representation Learning for NLP, 2017.

[32] P. Minervini et al., "Towards neural theorem proving at scale", arXiv preprint arXiv:1807.08204, 2018.

[33] S. N. Tran, and S. A. G. Artur. "Deep logic networks: Inserting and extracting knowledge from deep belief networks", IEEE T NEUR NET LEAR. vol. 29, no.2, pp: 246-258, 2016.

[34] G. Sourek et al., "Lifted relational neural networks: Efficient learning of latent relational structures", J ARTIF INTELL RES, vol. 62, pp: 69-100, 2018.

[35] Y. Liang, C. Mao, and Y. Luo. "KG-BERT: BERT for knowledge graph completion." arXiv preprint arXiv:1909.03193 (2019).

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

[37] Clark, Kevin, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. "Electra: Pre-training text encoders as discriminators rather than generators." arXiv preprint arXiv:2003.10555 (2020).

[38] Schlichtkrull, Michael, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. "Modeling relational data with graph convolutional networks." In European Semantic Web Conference, pp. 593-607. Springer, Cham, 2018.

[39] Yao, Liang, Chengsheng Mao, and Yuan Luo. "Graph convolutional networks for text classification." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7370-7377. 2019.

[40] Nathani, Deepak, Jatin Chauhan, Charu Sharma, and Manohar Kaul. "Learning attention-based embeddings for relation prediction in knowledge graphs." arXiv preprint arXiv:1906.01195 (2019).

[41] Hu, Weihua, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. "Open graph benchmark: Datasets for machine learning on graphs." arXiv preprint arXiv:2005.00687 (2020).

[42] Hamilton, William L., Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs." arXiv preprint arXiv:1706.02216 (2017).

[43] Chiang, Wei-Lin, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks." In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 257-266. 2019.

[44] Zeng, Hanqing, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. "Graphsaint: Graph sampling based inductive learning method." arXiv preprint arXiv:1907.04931 (2019).

[45] Rong, Yu, Wenbing Huang, Tingyang Xu, and Junzhou Huang. "Dropedge: Towards deep graph convolutional networks on node classification." arXiv preprint arXiv:1907.10903 (2019).

[46] Wu, Felix, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. "Simplifying graph convolutional networks." In International conference on machine learning, pp. 6861-6871. PMLR, 2019.

[47] Rossi, Emanuele, Fabrizio Frasca, Ben Chamberlain, Davide Eynard, Michael Bronstein, and Federico Monti. "Sign: Scalable inception graph neural networks." arXiv preprint arXiv:2004.11198 (2020).

[48] Tran, Ha-Nguyen, and Erik Cambria. "A survey of graph processing on graphics processing units." The Journal of Supercomputing 74, no. 5 (2018): 2086-2115.

[49] Cavallari, Sandro, Erik Cambria, Hongyun Cai, Kevin Chen-Chuan Chang, and Vincent W. Zheng. "Embedding both finite and infinite communities on graphs [application notes]." IEEE Computational Intelligence Magazine 14, no. 3 (2019): 39-50.

[50] Wang, Xiao, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. "Community preserving network embedding." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1. 2017.

[51] Yang, Zhao, Rene Algesheimer, and Claudio J. Tessone. "A comparative analysis of community detection algorithms on artificial networks." Scientific reports 6, no. 1 (2016): 1-18.

[52] Guo, Qingyu, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. "A

survey on knowledge graph-based recommender systems." IEEE Transactions on Knowledge and Data Engineering (2020).

[53] Tang, Lei, and Huan Liu. "Leveraging social media networks for classification." Data Mining and Knowledge Discovery 23, no. 3 (2011): 447-478.

[54] Cavallari, Sandro, Vincent W. Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. "Learning community embedding with community detection and node embedding on graphs." In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 377-386. 2017.

[55] Kozdoba, Mark, and Shie Mannor. "Community detection via measure space embedding." Advances in Neural Information Processing Systems 28 (2015): 2890-2898.

[56] M. Mirza, and O. Simon, "Conditional generative adversarial networks", Manuscript: https://arxiv.org/abs/1709.02023, vol. 9, 2014, pp. 24.

[57] L. Yu et al. "Seqgan: Sequence generative adversarial nets with policy gradient." In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[58] R. D. Hjelm et al., "Boundary-seeking generative adversarial networks", arXiv preprint arXiv:1702.08431, 2017.

[59] S. Rajeswar et al., "Adversarial generation of natural language", arXiv preprint arXiv:1705.10929, 2017.

[60] Y. Wu et al., "Deep convolutional neural network with independent softmax for large scale face recognition", In Proceedings of the 24th ACM international conference on Multimedia, ACM, 2016.

[61] W. Chen, G. David and A. Michael, "Strategies for training large vocabulary neural language models", arXiv preprint arXiv:1512.04906, 2015.

[62] Y. Lin et al., "Modeling relation paths for representation learning of knowledge bases", arXiv preprint arXiv:1506.00379, 2015.

[63] X. Zhang, Z. Junbo and L.Yann, "Character-level convolutional networks for text classification", In Advances in neural information processing systems, 2015.

Tehseen Zia is working as Assistant Professor in Department of Computer Science, COMSATS University Islamabad. He holds a Ph.D. in Computer Science (2010) from Vienna University of Technology, Austria, and Master in Computer Science from University of Engineering and Technology, Texila (2003). He received Ph.D. scholarship from Higher Education Commission of Pakistan. His research interest includes Machine Learning and Neuro-Symbolic Computing.

David Windridge is Associate Professor (Reader) in Computer Science and Head of Data Science at Middlesex University, London. His research interests center on the fields of Machine-Learning, Cognitive Systems, and Computer Vision (he also has a former research interest in Astrophysics/Cosmology following his Ph.D at the University of Bristol, UK). He is Visiting Professor at the University of Surrey, UK and sits on the editorial board of Springer *Quantum Machine Intelligence*. He has authored more than 120 peer-reviewed publications.