Relation Adversarial Network for Low Resource Knowledge Graph Completion

Ningyu Zhang*
ningyu.zny@alibaba-inc.com
Alibaba Group & AZFT Joint Lab for
Knowledge Engine
China

Shumin Deng*
231sm@zju.edu.cn
Zhejiang University & AZFT Joint
Lab for Knowledge Engine
China

Zhanlin Sun
zhanlins@andrew.cmu.edu
School of Computer Science
Carnegie Mellon University
United States

Jiaoyan Chen
jiaoyan.chen@cs.ox.ac.uk
Department of Computer Science
Oxford University
United Kingdom

Wei Zhang
lantu.zw@alibaba-inc.com
Alibaba Group & AZFT Joint Lab for
Knowledge Engine
China

Huajun Chen†
huajunsir@zju.edu.cn
Zhejiang University & AZFT Joint
Lab for Knowledge Engine
China

ABSTRACT

Knowledge Graph Completion (KGC) has been proposed to improve Knowledge Graphs by filling in missing connections via link prediction or relation extraction. One of the main difficulties for KGC is a low resource problem. Previous approaches assume sufficient training triples to learn versatile vectors for entities and relations, or a satisfactory number of labeled sentences to train a competent relation extraction model. However, low resource relations are very common in KGs, and those newly added relations often do not have many known samples for training. In this work, we aim at predicting new facts under a challenging setting where only limited training instances are available. We propose a general framework called *Weighted Relation Adversarial Network*, which utilizes an adversarial procedure to help adapt knowledge/features learned from high resource relations to different but related low resource relations. Specifically, the framework takes advantage of a relation discriminator to distinguish between samples from different relations, and help learn relation-invariant features more transferable from source relations to target relations. Experimental results show that the proposed approach outperforms previous methods regarding low resource settings for both link prediction and relation extraction.

CCS CONCEPTS

• **Information systems → Information extraction**.

*Equal contribution and shared co-first authorship.
†Corresponding author.

KEYWORDS

Knowledge Graphs, Low Resource Knowledge Graph Completion, Adversarial Transfer Learning, Link Prediction, Relation Extraction

1 INTRODUCTION

Knowledge Graphs (KGs) organize facts in a structured way as triples in the form of *<subject, predicate, object>*, abridged as (s, p, o), where s and o denote entities and p builds relations between entities. Due to the convenience of building semantic connections between data from various sources, many large-scale KGs have been built in recent years and have led to a broad range of important applications, such as question answering [48], language understanding[60], relational data analytics [57], recommender systems [41], etc.. However, these KGs are usually far from complete and noisy; thus various Knowledge Graph Completion (KGC) methods have been proposed in recent years. KGC could be achieved either by **Link Prediction** via typical KG embedding models such as TransE [2], DistMult [45], etc., or by **Relation Extraction** (RE) models which aim to complete missing relations by extracting new relational facts from textual corpus [23, 29].

Long Tail and Low Resource Problems. One of the main difficulties for KGC is the long tail and low resource problems. For example, most KG embedding models for **Link Prediction** assume sufficient training triples which are indispensable when learning versatile vectors for entities and relations. However, as illustrated in Figure 1(a), the blue line shows there is a large portion of relations in FB15K having only a few triples, revealing the common existence of *long tail low resource relations*. The red line shows that the prediction results of relations are highly related to their frequency, and the results of low resource relations are much worse than those of highly frequent ones. **Relation Extraction** suffers identical difficulty. For example, distant supervision (DS) methods

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Jiaoyan Chen, Wei Zhang, and Huajun Chen
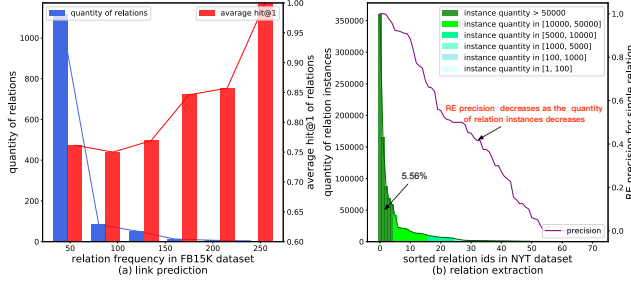

Figure 1: Illustration of low resource problems in typical KGC tasks: (a) link prediction, and (b) relation extraction. The relation frequency in (a) refers to the number of triples existing in FB15K for a relation. The relations IDs are sorted by the number of labeled sentences for that relation in (b).

were proposed to mitigate labeling laboriousness by aligning existing entity pairs of relation with textual sentences to generate training instances for that relation [30] automatically. However, this requires both sufficient entity pairs of that relation existing in the KG and an adequate number of matching sentences in the textual corpus correspondingly, which are difficult to be achieved for most relations in many KGs. As depicted in Figure 1(b), similarly, the green line shows there is a large portion of relations in NYT-Freebase having relatively low numbers of labeled sentences, and the prediction results decrease severely for those long-tail relations.

Adversarial Transfer Learning. Transfer learning is a research paradigm that focuses on transferring knowledge gained while learning from one domain, typically of high resource, and applying it to different but related domains, typically of low resource [31]. For example, the knowledge gained while learning to predict the relation *place_of_death* could apply when trying to predict *place_of_burial* if the latter lacks training samples. *Domain Adversarial Network* (DAN) has been successfully applied in transfer learning and is among top-performance architectures in standard transfer and domain adaptation settings [34]. The core idea is to extract domain-invariant features via an adversarial learning procedure that is capable of reducing the distribution discrepancy between the source domain and the target domain. Typically, domain adversarial networks consist of two adversarial players, where the first one is a domain discriminator D trained to distinguish the source domain from the target domain, and the second one is a feature extractor F which seeks to learn domain-invariant representations and serves as the adversary to confuse the domain discriminator [34]. The final training convergence produces a feature extractor that excels in extracting more transferable features across domains.

Relation Adversarial Network. Inspired by the idea of *Domain Adversarial Networks* for domain adaptation, we propose *Relation Adversarial Network* (**RAN**) for relation adaptation, and raise the research question: **whether it is feasible to adapt knowledge/features learnt from high resource relations such as *place_of_death*, to a different but related low resource relation, such as *place_of_burial*, via an adversarial procedure?** Similarly, the *relation adversarial network* consists of two adversarial players, where the first player is a relation discriminator D trained

to distinguish the source relation (i.e., the *place_of_death*) from the target relation (i.e., the *place_of_burial*). The second player is a feature extractor or instance encoder F, which tries to confuse the relation discriminator and seeks to learn common-features adaptable from source to target relations. The key intuition is to learn general relation-invariant features that can disentangle the factors of linguistic variations underlying different relations and close the linguistic gap between related relations.

Negative Transfer. Unlike standard adversarial domain adaptation, which assumes fully identical and shared label space between the source and target domains, it is expected for RAN to consider the adaptation from multiple source relations to one or multiple target relations. For example, in Figure 2, the RAN model is expected to learn general location information from three source relations (*place_of_death*, *place_of_birth* and *country*), and then adapt implied knowledge to the target relation (*place_of_burial*) to improve its prediction performance. This new problem set is more practical and challenging than standard domain adaptation, since different relations might have different impacts on the transfer and there exist outlier source relations that might cause *negative transfer* [31] when discriminating from the target relation. As seen in Figure 2, the source relation *capital* may result in negative transfer to the target relation *place_of_burial* as *capital* does not describe a relation between a person and a place. Although *partial transfer learning* has already been proposed to relax the shared label space assumption [5, 54], it still requires that the target label space is a subspace of the source label space. In contrast, source relations could be completely different from target relation(s) in our problem settings, where it is even more critical to address the *negative transfer* problem.

To address the aforementioned issues, we propose a general framework called **W**eighted **R**elation **A**dversarial **N**etwork (**wRAN**), which consists of three modules: (1) Instance encoder, which learns transferable features that can disentangle explanatory factors of **linguistic variations** cross relations. We implement the instance encoder with a convolutional neural network (CNN) considering both model performance and time efficiency. Other neural architectures such as recurrent neural networks (RNN) and BERT [9] can also be used as encoders. (2) Adversarial relation adaptation, which looks for a relation discriminator that can distinguish among samples having different relation distributions. Adversarial learning helps learn a neural network that can map a target sample to a feature space such that the discriminator will no longer distinguish it from a source sample. (3) Weighed relation adaptation, which identifies unrelated source relations/samples and down-weight their importance automatically to tackle the problem of *negative transfer* and encourage *positive transfer*.

We evaluate wRAN's feasibility with two different KGC tasks: link prediction and relation extraction, both in low resource settings. They share the same adversarial learning framework but with varying types of instance encoders. The former encodes triples of a relation, and the latter learn a sentence feature extractor that can learn general relation-invariant textual features to close the linguistic gap between relations.

The main contributions of the paper are summarized as follows.

- To the best of our knowledge, we are the first to apply adversarial transfer learning in tackling the low resource problems
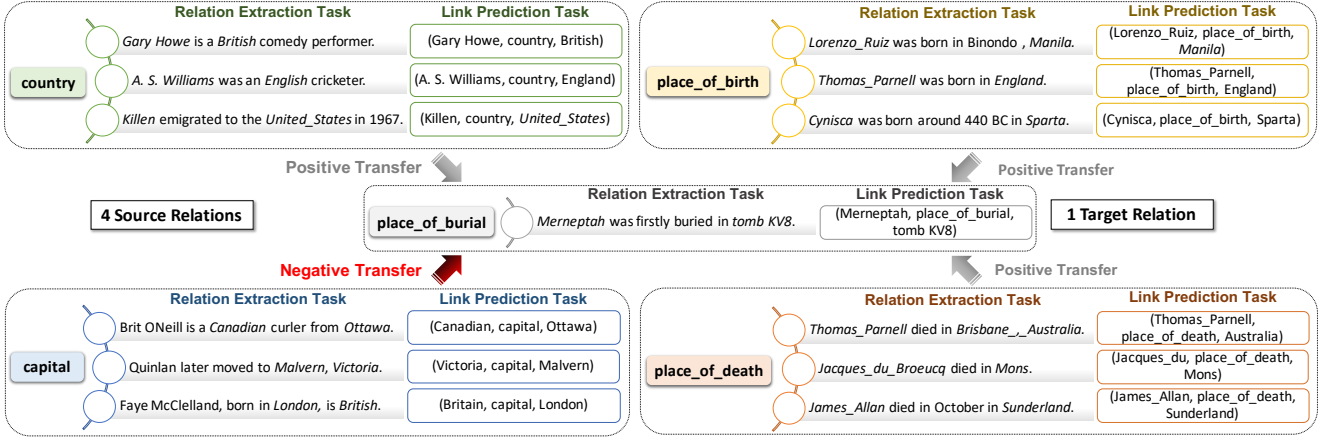
Figure 2: Illustration of relation adaptation where four source relations are adapted to one target relation. Note that there are two KGC tasks, relation extraction and link prediction, for each relation. In RE tasks, entity pairs are marked in italic.

for knowledge graph completion, which could be easily integrated with both link prediction and relation extraction.

- We propose a general framework called **R**elation **A**dversarial **N**etwork (RAN), which utilizes a relation discriminator to distinguish between samples from different relations and help learn relation-invariant features transferable from source relations to target relations.

- We present a relation-gated mechanism that fully relaxes the shared label space assumptions. This mechanism selects out outlier source relations/samples and alleviates negative transfer of irrelevant relations/samples, which can be trained in an end-to-end framework.

- Experiments show that our framework exceeds state-of-the-art results in low resource settings on public benchmark datasets for both link prediction and relation extraction.

2 RELATED WORK

Knowledge Graph Completion. KGC can be achieved either by link prediction from knowledge graph [59] or by extracting new relational facts from textual corpus [57]. A variety of link prediction approaches have been proposed to encode entities and relations into a continuous low-dimensional space. TransE [2] regards the relation r in the given fact (h, r, t) as a translation from h to t within the low-dimensional space. TransE achieves good results and has many extensions, including TransR [17], TransD [15], TransH [43], etc. RESCAL [26] studies on matrix factorization based knowledge graph embedding models using a bilinear form as score function. DistMult [45] simplifies RESCAL by using a diagonal matrix to encode relation, and ComplEx [38] extends DistMult into the complex number field. ConvE [8] uses a convolutions neural network as the score function. Analogy [19] optimize the latent representations with respect to the analogical properties of the embedded entities and relations. KBGAN [3] generates better negative examples to train knowledge graph embedding models via an adversarial learning framework. RotatE [36] defines each relation as a rotation from the source entity to the target entity in the complex vector space. KG-BERT [46] takes entity and relation descriptions of a triple as input and computes scoring function of the triple with BERT. For the

low resource setting, [44] proposed a one-shot relational learning framework, which learns a matching metric by considering both the learned embeddings and one-hop graph structures. [6] proposed a Meta Relational Learning (MetaR) framework to do few-shot link prediction in KGs. [59] propose a novel framework IterE iteratively learning embeddings and rules which can improve the quality of sparse entity embeddings and their link prediction results.

Relation extraction aims to detect and categorize semantic relations between a pair of entities. To alleviate the annotations given by human experts, weak supervision and distant supervision have been employed to automatically generate annotations based on KGs (or seed patterns/instances) [14, 16, 18, 30, 49, 52, 55–57]. However, all these models merely focus on extracting facts from a single domain, ignoring the rich information in other domains. Recently, there have been only a few studies on low resource relation extraction [10, 22, 23, 25, 29]. Of these, [22] followed the supervised domain adaptation paradigm. In contrast, [23, 29] worked on unsupervised domain adaptation. [58] proposes to take advantage of the knowledge from data-rich classes at the head of the distribution to boost the performance of the data-poor classes at the tail. [10, 25] presented adversarial learning algorithms for unsupervised domain adaptation tasks. However, their methods suffer from the negative transfer bottleneck when encountered partial domain adaptation. Our approach relaxes the label constraints that source relations could be completely different from target relations, which is more general and useful in real scenarios.

Adversarial Domain Adaptation. Generative adversarial nets (GANs) [13] have become a popular solution to reduce domain discrepancy through an adversarial objective concerning a domain classifier [12, 33, 39]. Recently, only a few domain adaptation algorithms [4, 7] that can handle **imbalanced relation distribution** or **partial adaptation** have been proposed. [5] proposed a method to simultaneously alleviate negative transfer by down-weighting the data of outlier source classes in category level. [54] proposed an adversarial nets-based partial domain adaptation method to identify the source samples at the instance level. However, most of these studies concentrate on image classification. Different from images, the text is more diverse and nosier. We believe these methods may

transfer to the low resource knowledge graph completion, but the effect of exact modifications is not apparent. We make the very first attempt to investigate the empirical results of these methods. Moreover, we propose a relation-gate mechanism to explicitly model both coarse-grained and fine-grained knowledge transfer to lower the negative transfer effects from categories and samples.

3 METHODOLOGY

3.1 Problem Definition

In this paper, we propose *relation adaptation* where we allow adaptation from multiple source relations to one or multiple target relations. This flexible paradigm finds real applications in practice, as we could perform transferring from a number of high resource relations to a single or a few long-tail low resource relations.

Being similar but slightly different to standard domain adaptation, in relation adaptation, we are provided with a set of *source relations* $S = \{R_s^i\}_{i=1}^{r_s}$ with r_s source relations, where for each relation we have $R_s^i = \{x_j^s, y_j^s\}_{j=1}^{n_s}$ with n_s labeled examples. Given one target relation $R_t = \{x_j^t\}_{j=1}^{n_t}$ with n_t unlabelled samples, the goal is to predict the labels of samples in the low resource relation R_t by adapting knowledge/features learnt from those high resource relations $\{R_s^i\}_{i=1}^{r_s}$.

Generally, the instances of source relation and target relation are sampled from probability distributions p and q respectively. Typically, we have $p \neq q$. The goal of this paper is to design a deep neural network $\mathbf{f} = F(x)$ that enables learning of transferable features from all other source relations and an adaptive classifier $y = G(\mathbf{f})$ to bridge the cross-relation discrepancy, such that the risk for target relation prediction $Pr_{(x,y)\sim q}[G(F(x)) \neq y]$ is minimized by leveraging the source relation supervisions.

3.2 Framework Overview

Our model consists of three parts, as shown in Figure 3:

Instance Encoder. Given an instance of entity pairs or sentences, we employ neural networks to encode the instance semantics into a vector. In this study, we implement the instance encoder with convolutional neural networks (CNNs), given both model performance and time efficiency.

Adversarial Relation Adaptation. We follow the standard adversarial transfer learning framework, which looks for a relation discriminator that can distinguish between samples having different relation distributions.

Weighted Relation Adaptation. We propose a relation-gate mechanism to identify the unrelated source relations/samples and down-weight their importance automatically to tackle the problem of *negative transfer*.

3.3 Instance Encoder

We utilize different encoders for link prediction and relation extraction, as shown in Figure 4.

To model the plausibility of a triple, we packed the sentences of (h, r, t) as a single sequence. The sentence is the name or description of each entity and relation, e.g., "*Steve Paul Jobs*" or "*Steven Paul Jobs was an American business magnate, entrepreneur and investor.*". The single input token sequence may contain two sentences

of entity h and r or three sentences of h, r, t packed together. To explicitly model the KG structure information, we also utilize pretrained TransE [2] embedding to form the final input representation. To extract the relation between entities from corpus, we utilize pretrained word embedding and random initialized position embedding [50]. Given a sentence $s = \{w_1, ..., w_L\}$, where w_i is the i-th word in the sentence, the input is a matrix consisting of L vectors $V = [v_1, \cdots, v_L]$.

We apply non-linear transformation $F(\cdot)$ to the vector representation of V to derive a feature vector $\mathbf{f} = F(V; \theta)$. We choose two convolutional neural architectures, CNN [50] and PCNN [49], to encode input embeddings into instance embeddings. Other neural architectures such as RNN [53] and more sophisticated approaches such as ELMo [28] and BERT [9] can also be used. We adopt the unshared feature extractors for both relations since unshared extractors are able to capture more relation specific features [39]. We train the source discriminative model $C(F_s(x; \theta))$ for the classification task by learning the parameters of the source feature extractor F_s and classifier C:

$$\min_{F_s, C} \mathbb{L}_s = \mathbb{E}_{x, y \sim p_s(x,y)} L(C(F_s(x; \theta)), y) \tag{1}$$

where y is the label of the source data x, $L(\cdot)$ is the loss function for classification. Afterwards, the parameters of F_s and C are fixed. Notice that, it is easy to obtain a pretrained model from the source relations, which is convenient in real scenarios.

3.4 Adversarial Relation Adaptation

To built *Relation Adversarial Network*, we follow the standard adversarial transfer learning framework, which is a popular solution in both computer vision [39] and NLP [32]. Similarly but differently, the adversarial learning procedure for wRAN is a two-player game, where the first player is the relation discriminator D trained to distinguish the source relation from the target relation, and the second player is the feature extractor F which is trained simultaneously to confuse the relation discriminator and seeks to learn common-features adaptable from source to target relation. The general idea is to learn both relation discriminative and relation-invariant features, where the loss of the label predictor of the source data is minimized while the loss of the relation discriminator is maximized.

$$\min_{F_s, F_t} \max_D \mathbb{L}(D, F_s, F_t) = \mathbb{E}_{x \sim p_s(x)}[\log D(F_s(x))] + \\ \mathbb{E}_{x \sim p_t(x)}[1 - \log D(F_t(x))] \tag{2}$$

where F_s and F_t are the feature extractors for the source and the target data, respectively, and D is a binary relation discriminator (corresponding to the discriminator in the original GAN) with all the source data labeled as 1 and all the target data labeled as 0.

3.5 Weighted Relation Adaptation

In practice, it is expected to adopt features from multiple related relations to the target relation. However, different source relations might have different contributions to the transfer. More importantly, those unrelated or outlier relations might cause *negative transfer*. Moreover, the negative transfer might also happen at the sample level, i.e., different samples in the same relation might also
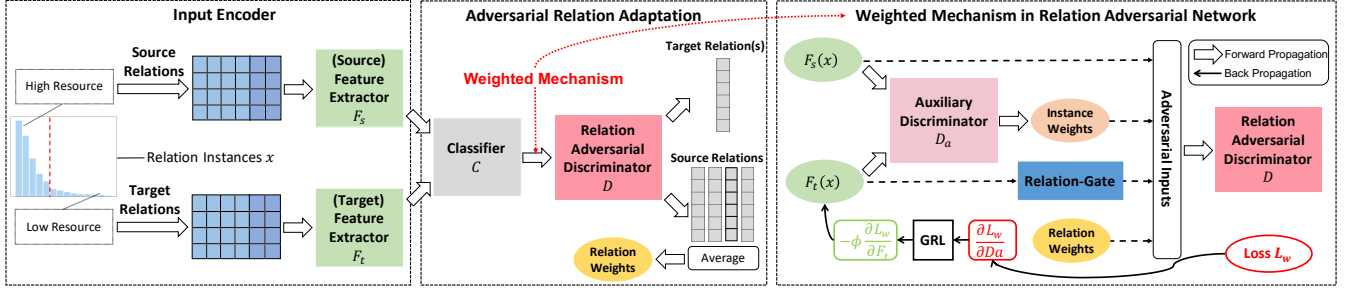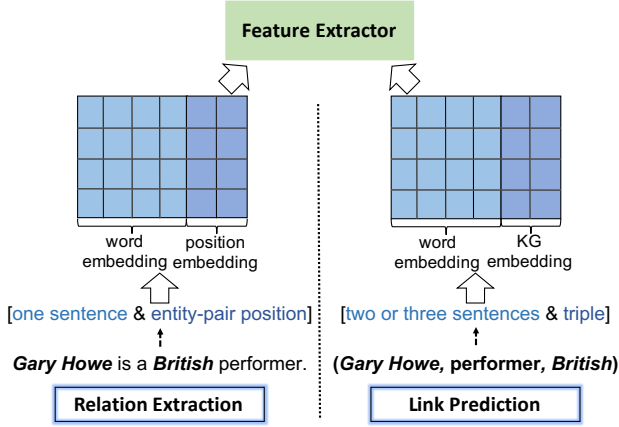
Figure 3: Overview of our approach.

word   position
embedding   embedding

word   KG
embedding   embedding

[one sentence & entity-pair position]   [two or three sentences & triple]

Gary Howe is a *British* performer.   (*Gary Howe, performer, British*)

Relation Extraction   Link Prediction

Figure 4: Instance Encoder.

have a different effect on the transfer. Considering the relation *educated_at* as an example, given an instance *"James Alty graduated from Liverpool University"* from target relation in the relation extraction task, semantically, a more similar instance of *"Chris Bohjalian graduated from Amherst College"* will provide more reference while a dissimilar instance *"He was a professor at Reed College where he taught Steve Jobs"* may have little contributions. To address these challenges, we propose *weighted relation adaptation* to evaluate the significance of each source relation/sample to the target relation through two perspectives: **relation correlation** and **instance transfer ability**.

Relation Correlation. Given that not all source relations are beneficial and can be adapted to the target relation, it is intuitive to assign different weights to different relations to lower the negative transfer effect of outlier relations and encourage positive transfer for those more similar or related relations. For example, it is necessary to lower the *capital* relation to mitigating negative transfer. Notice that, the output of the source classifier to each data point provides a probability distribution over the source label, we average the label predictions on all target data from the source classifier as relation weights. The intuition here is that the probability of assigning the target data to the source outlier relations is sufficiently small. We average the label predictions on all target data from the source classifier as relation weights [5]. Practically, the source classifier $C(F_s(x_i))$ reveals a probability distribution over

the source relation space r_s. This distribution characterizes well the probability of assigning x_i to each of the $|r_s|$ relations. We average the label predictions $\hat{y}_i = C(F_s(x_i))$, $x_i \in D_t$, on all target data since it is possible that the source classifier can make a few mistakes on some target data and assign large probabilities to false relations or even to outlier relations. The weights indicating the contribution of each source relations to the training can be computed as follows:

$$w^{relation} = \frac{1}{n_t} \sum_{i=1}^{n_t} \hat{y}_i \qquad (3)$$

where $w^{relation}$ is a $|r_s|$-dimensional weight vector quantifying the contribution of each source relation.

Instance Transfer Ability. Although the relation weights provide a global weights mechanism to de-emphasize the effect of outlier relations, different instances have different impacts, and not all instances are transferable. It is necessary to learn fine-grained instance weights to lower the effects of samples that are nontransferable.

Given the instance encoder of the source and target relations, we utilize a pretrained auxiliary relation discriminator for instance weights learning. We regard the output of the optimal parameters of the auxiliary relation discriminator as instance weights. The concept is that if the activation of the auxiliary relation discriminator is large, the sample can be almost correctly discriminated from the target relation by the discriminator, which means that the sample is likely to be nontransferable [54].

Practically, given the learned $F_s(x)$ from the instance encoder, a relation adversarial loss is used to reduce the shift between relations by optimizing $F_t(x)$ and auxiliary relation discriminator D_a:

$$\min_{F_t} \max_{D_a} \mathbb{L}_d(D_a, F_s, F_t) = \mathbb{E}_{x \sim p_s(x)}[\log D_a(F_s(x))] + \\ \mathbb{E}_{x \sim p_t(x)}[1 - \log D_a(F_t(x))] \qquad (4)$$

To avoid a degenerate solution, we initialize F_t using the parameter of F_s. The auxiliary relation discriminator is given by $D_a(f) = p(y = 1|x)$ where x is the input from the source and the target relation. If $D_a(f) \equiv 1$, then it is likely that the sample is nontransferable, because it can be almost perfectly discriminated from the target distribution by the relation discriminator. The contribution of these samples should be small. Hence, the weight function should be inversely related to $D_a(f)$, and a natural way to define the weights of the source samples is:

$$w_i^{instance} = \frac{1}{\frac{D_a(F_s(x))}{D_a(F_t(x))} + 1} = 1 - D_a(f) \qquad (5)$$

Relation-gate Mechanism. Both relation and instance weights are helpful. However, it is obvious that the weights of different granularity have different contributions to different target relations. On the one hand, for target relations (e.g., *located_in*) with relatively less semantically similar source relations, it is advantageous to strengthen the relation weights to reduce the negative effects of outlier relations. On the other hand, for target relations (e.g., *educated_in*) with many semantically similar source relations (e.g., *live_in*, *was_born_in*), it is difficult to differentiate the impact of different source relations, which indicates the necessity of learning fine-grained instance weights.

Thus for an instance in the source relation with label y_j, the weight of this instance is:

$$w_i^{total} = \alpha w_i^{instance} + (1 - \alpha) w_j^{relation} \qquad (6)$$

where $w_j^{relation}$ is the value in the jth-dimension of $w^{relation}$. We normalize the weight $w_i^{total} = \frac{n_s w_i^{total}}{\sum_{i=1}^{n_s} w_i^{total}}$. α is the output of relation-gate to explicitly balance the instance and relation weights which is computed as below.

$$\alpha = \sigma(W_r F_t(x)) \qquad (7)$$

where σ is the activation function, W_r is the weight matrix.

3.6 Initialization and Training

Objective Function. The overall objectives of our approach are $\mathbb{L}_s$, $\mathbb{L}_d$ and:

$$\min_{F_t} \max_{D_r} \mathbb{L}_w(C, D_r, F_s, F_t) =$$
$$\mathbb{E}_{x \sim p_s(x)}[w^{total} \log D_r(F_s(x))] + \qquad (8)$$
$$\mathbb{E}_{x \sim p_t(x)}[1 - \log D_r(F_t(x))]$$

where D_r is the relation adversarial. Note that, weights w^{total1} are automatically computed and assigned to the source relation data to de-emphasize the outlier relation and nontransferable instances regarding partial adaptation, which can mitigate negative transfer. The overall training procedure is shown below.

Algorithm 1 Overall Training Procedure

1: Pre-train F_s and C on the source domain and fix all parameters afterward.
2: Compute relation weights.
3: Pre-train F_t and D_a and compute instance weights, then fix parameters of D_a.
4: Train F_t and D_r, update the parameters of F_t through GRL.

---

[1]The weights can be updated in an iterative fashion when F_t changes. However, we found no improvement in experiments, so we compute the weights and fix them.

4 EXPERIMENTS

4.1 Datasets and Evaluation

Knowledge Graph. Existing benchmarks for KGC, such as FB15k-237 [37] and YAGO3-10 [20] are all small subsets of real-world KGs. These datasets consider the same set of relations during training and testing and often contain sufficient training triples for each relation. To construct datasets for low-resource learning, we go back to original KGs and select those relations that have sparse triples as low-resource relations. We refer to the rest of the relations as source relations since their triples provide important knowledge for us to match entity pairs. Our first dataset FB1.5M is based on Freebase [1], where we remove those inverse relations. We select 30 relations with less than 500 triples as target low-resource relations and the rest as source relations. To show that our model is able to operate on standard benchmarks, we also conduct experiments on FB15k-237. We follow a similar process of filtering low-resource relations to construct FB15k-237-low as a target low-resource dataset. The dataset statistics are shown in Table 1.

Table 1: Summary statistics of datasets

Dataset	#Rel	#Ent	#Train	#Dev	#Test
FB1.5M	4,018	1,573,579	799,104	20,254	21,251
FB15K-237-low	237	14,541	272,116	10,576	11,251
ACE05	6	19,684	351	80	80
Wiki-NYT	60	32,614	615,691	231,345	232,549

Text Corpus. We use the ACE05[2] dataset to evaluate our approach by dividing the articles from six genres and seven relations into respective domains: broadcast conversation (bc), broadcast news (bn), telephone conversation (cts), newswire (nw), usenet (un) and weblogs (wl). We use the same data split followed by [10], in which bn & nw are used as the source, half of bc, cts, and wl are used as the target for training (no label available in the unsupervised setting), and the other half of bc, cts, and wl are used as target for test. We split 10% of the training set to form the development set and fine-tune hyper-parameters. We calculate the proxy A-distance [21] $d_{\mathcal{A}} = 2(1 - 2\epsilon)$, where ϵ is the generalization error of a trained domain classifier. The average A-distance from all sources to target domains is about 1.1, which shows there exists a big difference between domains. We conducted two kinds of experiments. The first is standard adaptation, in which the source and target have the same relation set. The second is partial adaptation, in which the target has only half of the source relations. For the DS setting, we utilize two existing datasets NYT-Wikidata [51], which align Wikidata [40] with New York Times corpus (NYT), and Wikipedia-Wikidata [35], which align Wikidata with Wikipedia. We filter 60 shared relations to construct a new dataset Wiki-NYT, in which Wikipedia contains source relations and the NYT corpus contains target relations. We conducted partial DA experiments (60 relations → 30 relations), and randomly choose half of the relations to sample the target data. Our datasets are available at https://github.com/zxlzr/RAN.

[2]https://catalog.ldc.upenn.edu/LDC2006T06

Table 2: Main results of entity prediction on the FB1.5M and FB15k-237-low dataset.

Model	FB1.5M					FB15K-237-low				
	MRR	MR	HIT@1	HIT@3	HIT@10	MRR	MR	HIT@1	HIT@3	HIT@10
TransE [2]	0.075	15932	0.046	0.087	0.128	0.203	325	0.105	0.235	0.312
TransH [43]	0.079	15233	0.049	0.089	0.130	0.205	352	0.110	0.206	0.322
TransR [17]	0.081	15140	0.050	0.090	0.135	0.209	311	0.095	0.265	0.342
TransD [15]	0.085	15152	0.048	0.089	0.139	0.212	294	0.112	0.283	0.351
DistMult [45]	0.065	21502	0.036	0.076	0.102	0.185	411	0.123	0.215	0.276
ComplEx [38]	0.072	16121	0.041	0.090	0.115	0.192	508	0.152	0.202	0.296
ConvE [8]	-	-	-	-	-	0.224	249	0.102	0.262	0.352
Analogy [19]	0.116	13793	0.102	0.145	0.205	0.235	267	0.096	0.256	0.357
KBGAN [3]	-	-	-	-	-	-	-	0.142	0.283	0.342
RotatE [36]	0.125	10205	0.135	0.253	0.281	0.246	225	0.153	0.302	0.392
KG-BERT [46]	0.119	13549	0.093	0.195	0.206	0.258	159	0.146	0.286	0.371
IterE [59]	0.109	14169	0.073	0.175	0.186	0.238	178	0.152	0.275	0.361
wRAN	**0.141**	**9998**	**0.199**	**0.284**	**0.302**	**0.262**	**139**	**0.212**	**0.354**	**0.415**

4.2 Parameter Settings

To fairly compare the results of our models with those baselines, we set most of the experimental parameters following [10, 18]. We train GloVe [27] word embeddings on the Wikipedia and NYT corpus with 300 dimensions. In both the training and testing set, we truncate sentences with more than 120 words into 120 words. To avoid noise at the early stage of training, we use a similar schedule method as [11] for the trade-off parameter to update F_t by initializing it at 0 and gradually increasing to a pre-defined upper bound. The schedule is defined as: $\Phi = \frac{2 \cdot u}{1+exp(-\alpha \cdot p)} - u$, where p is the training progress linearly changing from 0 to 1, $\alpha = 1$, and u is the upper bound set to 0.1 in our experiments. We implemented baselines via OpenKE[3] and OpenNRE[4].

4.3 Link Prediction

The link (entity) prediction task predicts the head entity h given $(?, r, t)$ or predicts the tail entity t given $(h, r, ?)$ where ? means the missing element. The results are evaluated using a ranking produced by the scoring function $f(h, r, t)$ on test triples. Each correct test triple (h, r, t) is corrupted by replacing either its head or tail entity with every entity $e \in \mathbb{E}$; then these candidates are ranked in descending order of their plausibility head or tail entity with every entity $e \in \mathbb{E}$, then these candidates are ranked in descending order of their plausibility score. We report three common metrics, Mean Reciprocal Rank (MRR) and Mean Rank (MR) of correct entities and Hits@N which means the proportion of correct entities in top N. A lower MR is better while a higher MRR and Hits@N is better.

We compare wRAN with multiple state-of-the-art KG embedding methods as follows: **TransE** [2] and its extensions **TransH** [43], **TransD** [15], **TransR** [17]; **DistMult** [45] is a generalized framework where entities are low-dimensional vectors learned from a neural network and relations are bilinear and/or linear mapping functions; **ComplEx** [38] extends DistMult into the complex number field; **ConvE** [8] is a multi-layer convolutional network model

for link prediction; **Anology** [19] is a framework which optimize the latent representations with respect to the analogical properties of the embedded entities and relations; **KBGAN** [3] is an adversarial learning framework for generating better negative examples to train knowledge graph embedding models; **RotaE** [36] defines each relation as a rotation from the source entity to the target entity in the complex vector space; **KG-BERT** [46] takes entity and relation descriptions of a triple as input and computes scoring function of the triple with BERT.

Following [8] , we only report results under the filtered setting which removes all corrupted triples that appeared in training, development, and test set before getting the ranking lists. Table 2 shows link prediction performance of various models. We can observe that: 1) **wRAN** can achieve lower MR than baseline models, and it achieves the lowest mean ranks on FB1.5M and FB15k-237-low to our knowledge. 2) The Hits@N scores of **wRAN** are higher than in state-of-the-art methods. **wRAN** can leverage the relation-invariant features from source relations to benefit the low-resource target relations. 3) The improvement of **wRAN** over baselines on FB1.5M is larger than FB15K-237-low because FB1.5M has more source relations, which can leverage more knowledge from semantically related relations.

Triple classification aims to judge whether a given triple (h, r, t) is correct or not. Table 3 presents triple classification accuracy of different methods on FB1.5M and FB15K-237-low. We can see that **wRAN** clearly outperforms all baselines by a large margin, which shows the effectiveness of our method. We ran our models ten times and found the standard deviations are less than 0.2, and the improvements are significant ($p < 0.01$). To our knowledge, **wRAN** achieves the best results so far. For more in-depth performance analysis, we note that TransE could not achieve high accuracy scores because it could not deal with 1-to-N, N-to-1, and N-to-N relations. TransH, TransR, TransD outperform TransE by introducing relation to specific parameters. ConvE shows decent results, which suggests that CNN models can capture global interactions among the entity and relation embeddings. However, their improvements are still limited to those low-resource relations.

---

[3]https://github.com/thunlp/OpenKE
[4]https://github.com/thunlp/OpenNRE

Table 3: Triple classification accuracy (in percentage) on the FB1.5M and FB15K-237-low dataset.

Method	FB1.5M	FB15K-237-low	Avg.
TransE [2]	56.6	74.2	65.4
TransH [43]	56.9	75.2	66.1
TransR [17]	5 6.4	76.1	66.3
TransD [15]	60.2	76.5	68.4
DistMult [45]	60.5	82.2	71.4
ComplEx [38]	61.2	83.3	72.3
Analogy [19]	63.1	85.2	74.2
ConvE [8]	63.0	84.2	73.6
KG-BERT [46]	67.2	87.5	77.4
IterE [59]	65.2	86.4	75.8
wRAN	**69.2**	**90.9**	**80.0**

4.4 Relation Extraction

Evaluation Results on ACE05. To evaluate the performance of our proposed approach, we compared our model with various conventional domain adaptation models: **wRAN** is our approach, **CNN+DANN** is an unsupervised adversarial DA method [10], **Hybrid** is a composition model that combines traditional feature-based method, CNN and RNN [24], and **FCM** is a compositional embedding model. The first experiment takes the standard adaptation setting where we assume the source and target share the same relation set. In this setting, as the evaluation results are shown in Table 4, we observe that our model achieves performance comparable to that of CNN+DANN, which is a state-of-the-art model, and significantly outperforms the vanilla models without adversarial learning. This shows that adversarial learning is effective for learning domain-invariant features to boost performance as conventional adversarial domain adaptation models. The second experiment takes the partial adaptation setting where we only choose a part of the relations as target and study the weighted adaptation from a larger set of source relations. As shown in Table 4 again, our model significantly outperforms the plain adversarial adaptation model, CNN+DANN, in the partial setting. This demonstrates the efficacy of our hybrid weights mechanism[5].

Table 4: F1 score of normal and partial DA on ACE05 dataset. * indicates $p_{value} < 0.01$ for a paired t-test evaluation.

Standard Adaptation	bc	wl	cts	Avg.
FCM	61.90	N/A	N/A	N/A
Hybrid	63.26	N/A	N/A	N/A
CNN+DANN	65.16	55.55	57.19	59.30
wRAN	**66.15***	**56.56***	56.10	**59.60***

Partial Adaptation	bc	wl	cts	Avg.
CNN+DANN	63.17	53.55	53.32	56.68
wRAN	**65.32***	**55.53***	**54.52***	**58.92***

[5]Since the adversarial adaptation method significantly outperforms traditional methods [10], we skip the performance comparison with FCM and Hybrid for partial adaptation.

Evaluation Results on Wiki-NYT. For the DS setting, we consider the setting of (1) **unsupervised adaptation** in which the target labels are removed, (2) **supervised adaptation** in which the target labels are retained to fine-tune our model.

Unsupervised Adaptation. Target relations are unnecessary in unsupervised adaptation. We report the results of our approach and various baselines: **wRAN** is our unsupervised adaptation approach, **PCNN (No DA)** and **CNN (No DA)** are methods trained on the source by PCNN [18] and CNN [50] and tested on the target. Following [18], we perform both held-out evaluation as the *precision-recall curves* shown in Figure 6 and manual evaluation of *top-500 prediction results*, as shown in Table 5.

Figure 5: Precision values of the top 100, 200 and 500 sentences for unsupervised and supervised adaptation. * indicates $p_{value} < 0.01$ for a paired t-test evaluation.
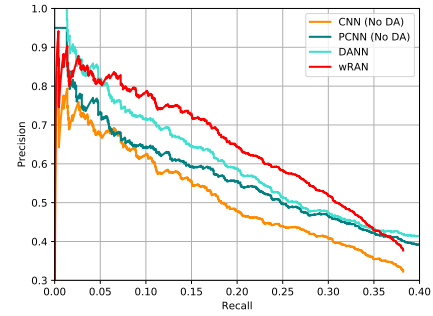
Figure 6: Unsupervised adaptation results.

Precision	Top 100	Top 200	Top 500	Avg.
CNN (No DA)	0.62	0.60	0.59	0.60
PCNN (No DA)	0.66	0.63	0.61	0.63
CNN+DANN	0.80	0.75	0.67	0.74
CNN	0.85	0.80	0.69	0.78
PCNN	0.87	0.84	0.74	0.81
Rank+ExATT	0.89	0.84	0.73	0.82
wRAN	**0.85***	**0.83***	**0.73***	**0.80***
+25%	0.88	0.84	0.75	0.82
+50%	0.89	0.85	0.76	0.82
+75%	0.90	0.85	0.77	0.83
+100%	0.88	0.86	0.77	0.83

We observe that (1) Our approach achieves the best performance among all the other unsupervised adaptation models, including CNN+DANN. This further demonstrates the effectiveness of the hybrid weights mechanism. (2) Our unsupervised adaptation model achieves nearly the same performance even with the supervised approach CNN; however, it does not outperform PCNN. This setting could be advantageous as in many practical applications, the knowledge bases in a vertical target domain may not exist at all or must be built from scratch.

Supervised Adaptation. Supervised Adaptation does require labeled target data; however, the target labels might be few or noisy.

In this setting, we fine-tune our model with target labels. We report the results of our approach and various baselines: **wRAN+k%** implies fine-tuning our model using k% of the target domain data, **PCNN** and **CNN** are the methods trained on the target domain by PCNN [18] and CNN [50], and **Rank+ExATT** is the method trained on the target domain which integrates PCNN with a pairwise ranking framework [47].
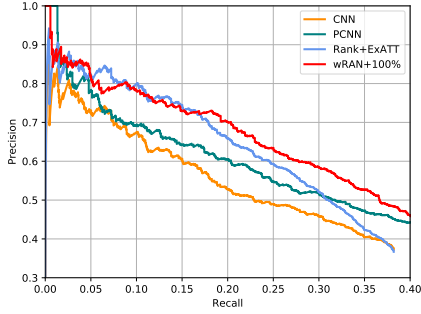

Figure 7: Supervised adaptation results.

As shown in Figure 7 and Table 5, we observe that: (1) Our fine-tuned model +100% outperforms both CNN and PCNN and achieves results comparable to that of Rank+ExATT. (2) The extent of improvement from using 0% to 25% of target training data is consistently more significant than others such as using 25% to 50%, and fine-tuned model with only thousands labeled samples (+25%) matches the performance of training from scratch with 10× more data, clearly demonstrating the benefit of our approach. (3) The top 100 precision of fine-tuned model degrades from 75% to 100%. This indicates that there exits noisy data which contradict with the data from the source domain. We will address this by adopting additional denoising mechanisms like reinforcement learning, which will be part of our future work.

4.5 Ablation Study

To better demonstrate the performance of different strategies in our model, we separately remove the relation and instance weights. The experimental results of relation extraction and link prediction are summarized in Table 5, Table 6, and Table 7, respectively. **wRAN** is our method; **w/o gate** is the method without relation-gate (α is fixed.); **w/o relation** is the method without relation weights ($\alpha = 1$); **w/o instance** is the method without instance weights ($\alpha = 0$); **w/o both** is the method without both weights.

Table 5: Ablation study results for relation extraction on Wiki-NYT dataset.

Precision	Top 100	Top 200	Top 500	Avg.
wRAN	**0.85**	**0.83**	**0.73**	**0.80**
w/o gate	0.85	0.79	0.70	0.78
w/o relation	0.81	0.76	0.66	0.74
w/o instance	0.84	0.78	0.69	0.77
w/o both	0.80	0.75	0.65	0.73

Table 6: Ablation study results of triple classification on the FB1.5M and FB15K-237-low dataset.

Method	FB1.5M	FB15K-237-low	Avg.
wRAN	**69.2**	**90.9**	**80.0**
w/o gate	67.2	89.5	78.3
w/o relation	66.0	88.3	77.2
w/o instance	65.9	87.6	76.7
w/o both	65.2	87.2	76.2

We observe that: (1) Performance significantly degrades when we remove "relation-gate." This is reasonable because the relation category and instances play different roles for different relations, while **w/o gate** treats weights equally which hurts performance. (2) Performance degrades when we remove "relation weights" or "instance weights." This is reasonable because different weights have different effects on de-emphasizing outlier relations or instances.

4.6 Parameter Analysis

Relation-Gate. To further explore the effects of relation-gate, we visualize α for all target relations on both FB1.5M and Wiki-NYT dataset. From the results shown in Figure 8, we observe the following: (1) The instance and relation weights have different influences on performance for different relations. Our relation-gate mechanism is powerful to find that instance weights is more important for those relation (e.g., *educated_at*, *live_in* a.k.a., relations with highest α score in Figure 8 (b)) while relation weights are more useful for other relations. (2) The relation weights have relatively more influence on the performance than instance weights for most of the relations due to the noise and variations in instances; however, the relation weights are averaged on all target data and thus less noisy.
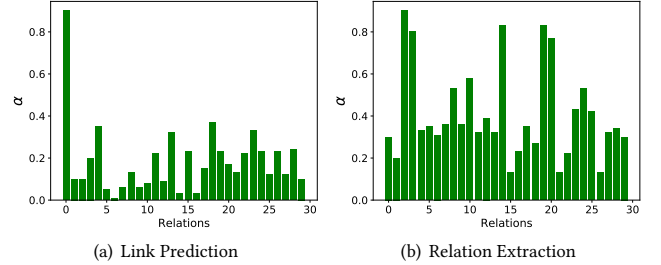

(a) Link Prediction   (b) Relation Extraction

Figure 8: α w.r.t #Relations.

Different Number of Target Relations. We investigated the transferability of link prediction[6] (triple classification) and relation extraction by varying the number of target relations in the FB1.5M and Wiki-NYT dataset[7]. Figure 9 shows that when the number of target relations decreases, the performance of KG-BERT/CNN+DANN degrades quickly, implying the severe negative transfer. We observe that RAN outperforms KG-BERT/CNN+DANN when the number of target relations decreases. Note that, wRAN performs comparably to KG-BERT/CNN+DANN in standard-setting when the number of target relations is 60. This means that the weights mechanism will not wrongly filter out relations when there are no outlier relations.

---

[6] We choose 60 source relations randomly.

[7] The target relations are sampled three times randomly, and the results are averaged.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Jiaoyan Chen, Wei Zhang, and Huajun Chen

Table 7: Ablation study results of entity prediction on the FB1.5M and FB15k-237-low dataset.

Model	FB1.5M					FB15K-237-low				
	MRR	MR	HIT@1	HIT@3	HIT@10	MRR	MR	HIT@1	HIT@3	HIT@10
wRAN	**0.141**	**9998**	**0.199**	**0.284**	**0.302**	**0.262**	**139**	**0.212**	**0.354**	**0.415**
w/o gate	0.130	11020	0.155	0.286	0.291	0.252	155	0.192	0.343	0.399
w/o relation	0.126	11560	0.134	0.275	0.282	0.243	181	0.175	0.340	0.382
w/o instance	0.120	11900	0.126	0.270	0.278	0.238	202	0.171	0.332	0.376
w/o both	0.113	12520	0.100	0.263	0.253	0.229	254	0.166	0.326	0.364

Table 8: Examples for Case 1, 2 and 3, w^c and w^i denote relation and instance weights, respectively.

Triple Instances	Relations	w^r	w^i	α
(*Rio_de_Janeiro*, capital_of, *Brazil*)	capital_of	0.2	0.3	0.2
(*Malini_22_Palayamkottai*, director, *Sripriya*)	director	0.8	0.9	0.3
(*George_Stevens*, director, *The_Nitwits*)	director	0.6	0.6	0.5
(*Camp*, director, *Todd_Graff*).	director	0.5	0.4	0.5
(*Chris Bohjalian*, edu_in, *Amherst College*)	educated_in	0.4	0.7	0.9

Sentence Instances	Relations	w^r	w^i	α
He was born in *Rio_de_Janeiro*, *Brazil* to a German father and a Panama nian mother.	capital_of	0.1	0.2	0.1
In 2014, he made his Tamil film debut in *Malini_22_Palayamkottai* directed by *Sripriya*.	director	0.7	0.8	0.4
Sandrich was replaced by *George_Stevens* for the teams 1935 film *The_Nitwits*.	director	0.7	0.5	0.5
Camp is a 2003 independent musical_film written and directed by *Todd_Graff*.	director	0.7	0.3	0.4
Chris Bohjalian graduated from *Amherst College*	educated_in	0.4	0.7	0.9

(a) Link Prediction   (b) Relation Extraction
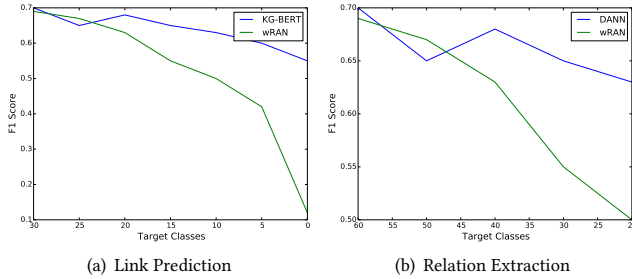
Figure 9: F1 w.r.t #Target Relations.

4.7 Case Study

Case 1: Relation-gate. We give some examples of how our relation-gate balance weights for relations and instances. In Table 8, we display the α of different relations. For relation *capital_of*, there are lots of dissimilar relations, so relation weights are more important, which results in a small α. For relation *educated_in*, the instance difference is more important so α is relatively large.

Case 2: Relation Correlation. We give some examples of how our approach assigns different weights for relations to mitigate the negative effect of outlier relations. In Table 8, we display the sentences from shared relations and outlier relations. The relation *capital_of* is an outlier relation whereas *director* is a shared relation. We observe that our model can automatically find outlier relations and assign lower weights to them.

Case 3: Instance Transfer Ability. We give some examples of how our approach assigns different weights for instances to de-emphasize the nontransferable samples. In Table 8, we observe that: (1) Our model can automatically assign lower weights to instances in outlier relations. (2) Our model can assign different weights for instances in the same relation space to down-weight the negative effect of nontransferable instances. (3) Although our model can identify some nontransferable instances, it still assigns incorrect weights to some instances (The end row in Table 8) which is semantically similar and transferable. We will address this by adopting additional mechanisms like transferable attention [42], which will be part of our future work.

5 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel model of *Weighted Relation Adversarial Network* for low resource knowledge graph completion. Extensive experiments demonstrate that our model achieves results that are comparable with that of state-of-the-art link prediction and relation extraction baselines. In the future, we plan to improve the results by jointly modeling link prediction and relation extraction. It will also be promising to apply our approach to other low resource NLP scenarios such as emerging event extraction, low resource question answering.

REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data.* AcM, 1247–1250.

[2] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS.* 2787–2795.

[3] Liwei Cai and William Yang Wang. 2017. Kbgan: Adversarial learning for knowledge graph embeddings. *arXiv preprint arXiv:1711.04071* (2017).

[4] Zhangjie Cao, Mingsheng Long, Jianmin Wang, Michael I Jordan, and MOE KLiss. 2017. Partial transfer learning with selective adversarial networks. *arXiv preprint arXiv:1707.07901* (2017).

[5] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. 2018. Partial Adversarial Domain Adaptation. In *Procddings of ECCV*, Vol. 1. 4.

[6] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta relational learning for few-shot link prediction in knowledge graphs. *arXiv preprint arXiv:1909.01515* (2019).

[7] Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. 2018. Re-Weighted Adversarial Adaptation Network for Unsupervised Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 7976–7985.

[8] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *AAAI.* AAAI Press, 1811–1818.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[10] Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. 2017. Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of IJCNLP*, Vol. 2. 425–429.

[11] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495* (2014).

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *JMLR* 17, 1 (2016), 2096–2030.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *processing of NIPS.* 2672–2680.

[14] Zhengqiu He, Wenliang Chen, Zhenghua Li, Meishan Zhang, Wei Zhang, and Min Zhang. 2018. SEE: Syntax-aware Entity Embedding for Neural Relation Extraction. In *Proceedings of AAAI.*

[15] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge Graph Embedding via Dynamic Mapping Matrix. *Proceedings of ACL* (2015), 687–696.

[16] Guoliang Ji, Kang Liu, Shizhu He, Jun Zhao, et al. 2017. Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions.. In *AAAI.* 3060–3066.

[17] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI.* AAAI Press, 2181–2187.

[18] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, Vol. 1. 2124–2133.

[19] Hanxiao Liu, Yuexin Wu, and Yiming Yang. 2017. Analogical Inference for Multi-relational Embeddings. In *Proceedings ICML.* 2168–2178.

[20] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. (2013).

[21] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430* (2009).

[22] Minh Luan Nguyen, Ivor W Tsang, Kian Ming A Chai, and Hai Leong Chieu. 2014. Robust domain adaptation for relation extraction via clustering consistency. In *Proceedings of ACL*, Vol. 1. 807–817.

[23] Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *Proceedings of ACL*, Vol. 2. 68–74.

[24] Thien Huu Nguyen and Ralph Grishman. 2016. Combining neural networks and log-linear models to improve relation extraction. In *Proceedings of IJCAI Workshop DLAI.*

[25] Thien Huu Nguyen, Barbara Plank, and Ralph Grishman. 2015. Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction. In *Proceedings of ACL*, Vol. 1. 635–644.

[26] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A Three-Way Model for Collective Learning on Multi-Relational Data. *Proceedings of ICML* (2011), 809–816.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP.* 1532–1543.

[28] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).* Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[29] Barbara Plank and Alessandro Moschitti. 2013. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Proceedings of ACL*, Vol. 1. 1498–1507.

[30] Pengda Qin, Weiran Xu, and William Yang Wang. 2018. DSGAN: Generative Adversarial Training for Distant Supervision Relation Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics, Melbourne, Australia, 496–505. https://doi.org/10.18653/v1/P18-1046

[31] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. 2005. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, Vol. 898. 1–4.

[32] Darsh J Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial Domain Adaptation for Duplicate Question Detection. *arXiv preprint arXiv:1809.02255* (2018).

[33] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein Distance Guided Representation Learning for Domain Adaptation.. In *Proceedings of AAAI.*

[34] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2019. Transferable Curriculum for Weakly-Supervised Domain Adaptation. (2019).

[35] Daniil Sorokin and Iryna Gurevych. 2017. Context-Aware Representations for Knowledge Base Relation Extraction. In *Proceedings of EMNLP.* 1784–1789.

[36] Zhiqing Sun, Zhi Hong Deng, Jian Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *Proceedings of ICLR* (2019).

[37] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* 1499–1509.

[38] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. *Proceedings ICML* (2016), 2071–2080.

[39] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of CVPR*, Vol. 1. 4.

[40] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[41] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *WWW.* ACM, 2000–2010.

[42] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. 2019. Transferable Attention for Domain Adaptation. (2019).

[43] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. *Proceedings of AAAI* (2014), 1112–1119.

[44] Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2018. One-shot relational learning for knowledge graphs. *arXiv preprint arXiv:1808.09040* (2018).

[45] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *Proceedings of ICLR* (2015).

[46] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for Knowledge Graph Completion. (2019).

[47] Hai Ye, Wenhan Chao, Zhunchen Luo, and Zhoujun Li. 2017. Jointly Extracting Relations with Class Ties via Effective Deep Ranking. In *Proceedings of ACL*, Vol. 1. 1810–1820.

[48] Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved Neural Relation Detection for Knowledge Base Question Answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics, Vancouver, Canada, 571–581. https://doi.org/10.18653/v1/P17-1053

[49] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP.* 1753–1762.

[50] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING.* 2335–2344.

[51] Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating relation paths in neural relation extraction. In *Proceddings of EMNLP*.

[52] Xiangrong Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Large scaled relation extraction with reinforcement learning. In *Processings of AAAI*, Vol. 2. 3.

[53] Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006* (2015).

[54] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. 2018. Importance Weighted Adversarial Nets for Partial Domain Adaptation. In *Proceedings of CVPR*. 8156–8164.

[55] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Jiaoyan Chen, Wei Zhang, and Hua-jun Chen. 2019. Transfer Learning for Relation Extraction via Relation-Gated Adversarial Learning. *arXiv preprint arXiv:1908.08507* (2019).

[56] Ningyu Zhang, Shumin Deng, Zhanling Sun, Xi Chen, Wei Zhang, and Huajun Chen. 2018. Attention-Based Capsule Networks with Dynamic Routing for Relation Extraction. In *Proceedings of EMNLP*.

[57] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 3016–3025. https://doi.org/10.18653/v1/N19-1306

[58] Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail Relation Extraction via Knowledge Graph Embeddings and Graph Convolution Networks. *arXiv preprint arXiv:1903.01306* (2019).

[59] Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. 2019. Iteratively Learning Embeddings and Rules for Knowledge Graph Reasoning. In *WWW*. ACM, 2366–2377.

[60] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1441–1451. https://doi.org/10.18653/v1/P19-1139