Learning Query Adaptive Anchor Representation for Inductive Relation Prediction

Zhiwen Xie[1], **Yi Zhang**[2], **Jin Liu**[1]*, **Guangyou Zhou**[2]*, **Jimmy Xiangji Huang**[3]
[1]School of Computer Science, Wuhan University
[2]School of Computer Science, Central China Normal University
[3]Information Retrieval and Knowledge Management Research Lab, York University
xiezhiwen@whu.edu.cn, yizhang@mails.ccnu.edu.cn,
jinliu@whu.edu.cn, gyzhou@mail.ccnu.edu.cn, jhuang@yorku.ca

Abstract

Relation prediction on knowledge graphs (KGs) attempts to infer the missing links between entities. Most previous studies are limited to the transductive setting where all entities must be seen during the training, making them unable to perform reasoning on emerging entities. Recently, the inductive setting is proposed to handle the entities in the test phase to be unseen during training, However, it suffers from the inefficient reasoning under the enclosing subgraph extraction issue and the lack of effective entity-independent feature modeling. To this end, we propose a novel Query Adaptive Anchor Representation (QAAR) model for inductive relation prediction. First, we extract one opening subgraph and perform reasoning by one time for all candidate triples, which is more efficient when the number of candidate triples is large. Second, we define some query adaptive anchors which are independent on any specific entity. Based on these anchors, we take advantage of the transferable entity-independent features (relation-aware, structure-aware and distance features) that can be used to produce entity embeddings for emerging unseen entities. Such entity-independent features is modeled by a query-aware graph attention network on the opening subgraph. Experimental results demonstrate that our proposed QAAR outperforms state-of-the-art baselines in inductive relation prediction task.

1 Introduction

Knowledge graphs (KGs) store a large amount of facts regarding the real-world knowledge in the form of (h, r, t), where h, t are head and tail entities and r is the relation between h and t. They have played important roles in various downstream applications, including but not limited to question answering (Yasunaga et al., 2021; Li and Xiong, 2022), semantic search (Xiong et al., 2017a; Zhang et al., 2021) and many more. Nowadays many large-scale KGs such as Freebase (Bollacker et al., 2008), Dbpedia (Lehmann et al., 2015), Yago (Hoffart et al., 2013; Mahdisoltani et al., 2015) and Wikidata (Vrandecic and Krötzsch, 2014) have been constructed, but they still suffer from an incompleteness problem. To make up such limitation, relation prediction or knowledge graph embedding (KGE) has been proposed to infer the missing links in KGs.

The conventional relation prediction models, such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), RotatE (Sun et al., 2019), TuckER (Balazevic et al., 2019), CompGCN (Vashishth et al., 2020), KBGAT (Nathani et al., 2019), ReInceptionE (Xie et al., 2020), DisenKGAT (Wu et al., 2021), PairRE (Chao et al., 2021), FFTAttH (Xiao et al., 2022), KGTuner (Zhang et al., 2022) have successively achieved satisfactory performance on relation prediction. They all follow a general strategy to represent the entities in the low-dimension vector space by using the KG structure and neighborhood information in the **transductive setting**, where all entities must be seen in the training phase. However, many KGs are evolving and new entities are emerging every day in the real world. These conventional relation prediction models in the transductive setings cannot effectively learn the embeddings for the new entities unless re-training the whole KGs from scrath. Recently, researchers have focused on relation prediction in the **inductive setting**, where new entities are emerging while testing. Compared to relation prediction in the transductive setting, the inductive setting is more challenging since the missing links can be inferred in KGs with unseen entities.

Currently, models on relation predictions in the inductive setting can be divided into two categories (Mai et al., 2021; Lin et al., 2022): rule-based methods and GCN-based methods. Rule-based methods (e.g., NeuralLP (Yang et al., 2017), DRUM (Sadeghian et al., 2019), RuleN (Meilicke et al.,

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 14041–14053

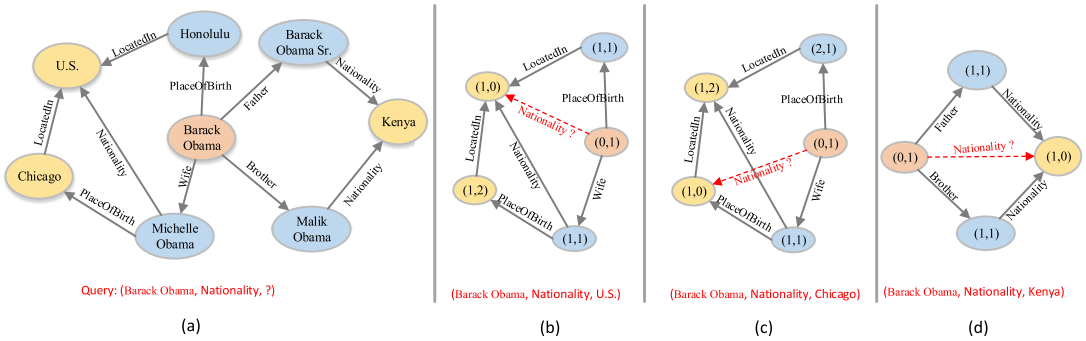

Figure 1: An example of node labeling of previous inductive relation prediction methods, such as GraIL (Teru et al., 2020), CoMPILE (Mai et al., 2021), TACT (Chen et al., 2021), ConGLR (Lin et al., 2022). The subfigure (a) is a subgraph surrounding entity *Barack Obama* and the query is (*Barack Obama*, *Nationality*, ?). The subfigures (b), (c) and (d) are extracted enclosing subgraphs for candidate triples (*Barack Obama*, *Nationality*, *U.S.*), (*Barack Obama*, *Nationality*, *Chicago*) and (*Barack Obama*, *Nationality*, Kenya). And the nodes in (b), (c), (d) are labeled by using the distances between each node and head (tail) node in corresponding candidate triple, which is widely used in previous works.

2018) etc.) are able to handle the relation prediction with new/unseen entities since the rules are independent of any specific entities, their performance is usually insufficient due to the limitation of the flexible issue. GCN-based methods, such as GraIL (Teru et al., 2020), CoMPILE (Mai et al., 2021), Meta-iKG (Zheng et al., 2021), TACT (Chen et al., 2021), ConGLR (Lin et al., 2022), require to extract a k-hop enclosing subgraph[1] for every candidate triple (h, r, t) and then perform reasoning on all these candidate subgraphs. However, these methods suffer from the inefficient issue since the number of candidate triples is usually large. Figure 1 shows an example of the query (*Barack Obama*, *Nationality*, ?). The subfigures (b), (c) and (d) are enclosing subgraphs extracted from graph (a) for three different candidate triples. Though the subgraphs (b) and (c) have the same structure, these previous methods have to perform reasoning two times, which will spend more reasoning time. Moreover, previous methods only use the distance as features and ignore rich local structure information in subgraphs, which leads to sub-optimal performance. As shown in Figure 1, the entity *Honolulu* in subfigure (b) is labeled as "(1,1)", which is the same with the entity *Malik Obama* in subfigure (d). This leads to similar node labeling for the nodes in (b) and (d), although these two subgraphs are totally different. Using these weak node features will result in poor performance since it is hard for the model to distinguish different subgraphs.

Based on the above analysis, we are motivated to address two research issues: (1) How to develop a model to perform inductive reasoning for all candidate triples with only one subgraph and one-time node labeling and reasoning. (2) How to extract adequate transferable entity-independent anchor[2] knowledge to produce more expressive entity embeddings for emerging entities.

To address these issues, we propose a novel query adaptive anchor representation (QAAR) model. Specifically, for a given query $(h, r, ?)$ where ? denotes a missing entity to be inferred, instead of extracting an enclosing subgraph for every candidate triple, we first extract a k-hop opening subgraph[3] surrounding the query entity h for all candidates. This enables our model to perform reasoning by one time to rank all candidates, which is more efficient when the number of candidates is large. Then, we propose a dynamic anchor-based node labeling method to label each node with rich entity-independent information. Specifically, we construct three kinds of anchors for each query $(h, r, ?)$, including relation types, the center node of the subgraph and the one-hop neighbors of the center node which is adaptively selected according to the query entity h. Based on these anchors, we can label each node in the subgraph with three kinds of features, namely relation-aware, structure-aware and distance features. These rich node features en-

[1]Following (Teru et al., 2020), a k-hop enclosing subgraph between nodes h and t contains all nodes that appear on the graph from h to t with the maximum k-hop distance.

[2]The anchors in this paper denote the transferable meta knowledge, which is independent of any specific entities and can be transferred to new graphs with unseen entities.

[3]Unlike the enclosing subgraph, the k-hop opening subgraph of entity h contains all neighboring nodes with the maximum k-hop distance.

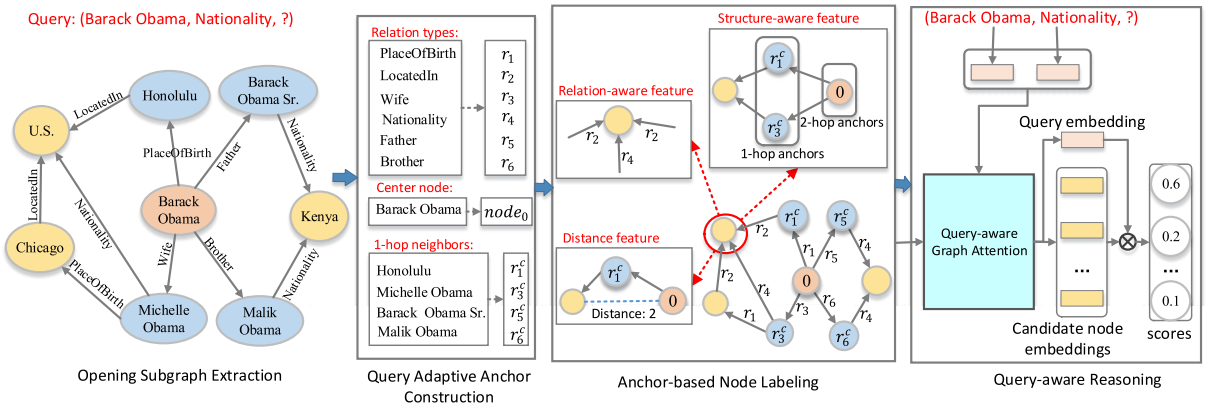

Figure 2: The overview of the proposed QAAR model.

able our model to learn expressive representation for subgraphs. Finally, a query-aware graph attention neural network is proposed to perform reasoning on subgraphs by gathering query-related neighborhood features. Experimental results on three benchmark datasets demonstrate that our proposed QAAR significantly outperforms previous methods, and achieves the new state-of-the-art (SOTA) performance for inductive relation prediction.

2 Our Approach

Figure 2 illustrates the overview of the proposed QAAR, which consists of four modules: (1) opening subgraph extraction; (2) query adaptive anchor construction; (3) anchor-based node labeling; and (4) query-aware reasoning. Next, we will describe each module in details.

2.1 Opening Subgraph Extraction

Multi-hop subgraphs can provide rich evidence to infer missing links due to the implicit correlation between different relations. For example, the relation $(A, farther, B)$ can be inferred according to the two-hop facts $(A, mother, C)$ and $(C, husband, B)$. Thus, we first extract a multi-hop subgraph for inductive relation prediction. Different from GraIL (Teru et al., 2020) and its extensions (e.g., TACT (Chen et al., 2021), CoMPILE (Mai et al., 2021) and ConGLR (Lin et al., 2022)), which extract a specific enclosing subgraph for every candidate triple (h, r, t), we attempt to extract one opening subgraph for all candidate entities so that we can score all candidates with one-time reasoning on the subgraph. For a given query $(h, r, ?)$, we first construct a k-hop opening subgraph for the query entity h by extracting the neighboring nodes surrounding h with maximum k-hop distance. We denote the k-hop opening subgraph of h as $\mathcal{G}_k(h)$.

2.2 Query Adaptive Anchor Construction

To learn transferable entity-independent node features in a given subgraph, we first design some anchors which can be shared by different subgraphs. Ideally, the anchors should be independent of any specific entities and have the ability to express local structure information of the given subgraph. To this end, we introduce the following three kinds of anchors:

Relation types. Intuitively, the relation types are common information for subgraphs with different entities. Even on a new KG with unseen entities, the same set of relation types can be used to express the relations among entities. The set of relation types are denoted as $R = \{r_1, r_2, \cdots, r_{|R|}\}$, where r_i is a relation in the relation set and $|R|$ is the number of relations.

Center node. In our paper, each subgraph is extracted surrounding the given query entity h which can be considered as the center node of the subgraph and each opening subgraph has a unique center node. We use the center node as an anchor and denote it as $node_0$[4]. With the center node, the model can be aware of where is the start node to perform reasoning.

One-hop neighbors of center node. Besides, we also define the one-hop neighbors of the center node as anchors to better learn the local structure of subgraphs. For inductive relation prediction, we cannot directly use entity-specific embeddings to represent these one-hop neighbors. To address this problem, we leverage relationship between each

[4]Note that h denotes a specific query entity in KG, while $node_0$ is a special symbol to represent the center node of a subgraph, which is transferable to any subgraph.

neighboring node and the center node to denote the neighbors. Specifically, for a neighboring entity t which is connected to the center node h with a triple (h, r_i, t), the neighboring entity t can be denoted as r_i^c which means a node connected to the center node with relation r_i. Take the triple (*Barack Obama*, *PlaceOfBirth*, *Honolulu*) as an example, the entity *Honolulu* can be denoted as "the PlaceofBirth of Barack Obama". Thus, all neighboring nodes of the center node can be represented by combining the relation and center node, which can yield a set of one-hop neighborhood anchors $D = \{r_i^c | (h, r_i, t) \in \mathcal{T}_k(h)\}$, where $\mathcal{T}_k(h)$ denotes the set of triples in $\mathcal{G}_k(h)$.

According to the above definition, we can obtain a set of anchor relations R and a set of anchor nodes $Anc = \{node_0\} \cup D$. These anchor relations and the nodes are independent on any specific entity, which can be acted as the transferred meta-knowledge to bridge the gap between the training KG and the test KG.

2.3 Anchor-based Node Labeling

In this subsection, we describe how to label each node in the opening subgraph using the anchors R and Anc.

Relation-aware features. Given a node n in the subgraph $\mathcal{G}_k(h)$, we extract the node's neighboring relations denoted as $\mathcal{R}(n)$. We can capture some important type information about the node from these neighboring relations. For example, the tail entities of relation *PlaceOfBirth* are entities with a place type. Therefore, we use these relations to learn embedding for node n:

$$\mathbf{v}_n^r = \sum_{r \in \mathcal{R}(n)} \mathbf{r} \tag{1}$$

where $\mathbf{r}$ is the embedding for the relation r.

Structure-aware features. Only using the relation information cannot adequately express the structure of a given subgraph. In this paper, we propose a structure-aware labeling method to enrich the node features with local structure information. In a given subgraph, the center node and its one-hop neighboring nodes are viewed as anchor nodes Anc as defined in Section 2.2. Thus, we can leverage neighboring relationship between each node with these anchor nodes to learn local structure of each node in the subgraph. Formally, let $\mathcal{A}_j(n) = \{a_i | a_i \in \mathcal{N}_j(n) \bigcap Anc\}$ be the j-th hop neighboring anchor nodes of node n, where $\mathcal{N}_j(n)$ denotes the j-th hop neighbors of node n. The j-th hop structure feature of node n can be denoted as:

$$\mathbf{v}_n^j = \sum_{a_i \in \mathcal{A}_j(n)} \mathbf{a}_i \tag{2}$$

where $\mathbf{a}_i$ is the embedding of anchor node a_i. Note that by gathering neighboring anchor nodes from different hops, we can obtain multi-hop structure-aware features $\{\mathbf{v}_n^0, \mathbf{v}_n^1, \cdots, \mathbf{v}_n^J\}$, where J is the maximum hop of the structure-aware features[5].

Distance features. The distance can reveal the degree of correlation between each node and the center node. Therefore, we also leverage distance features to enrich the representation of each node. Specifically, we visit the neighboring nodes from different hops of the center node and obtain the shortest distance between each node and the center node. Let d_n denote the distance between node n and the center node, the embedding of distance feature is denoted as $\mathbf{v}_n^d$.

Finally, the relation-aware, structure-aware and distance features are combined to learn the initial embedding of each node n:

$$\mathbf{e}_n^0 = \mathbf{v}_n^r + \mathbf{v}_n^0 + \cdots + \mathbf{v}_n^J + \mathbf{v}_n^d \tag{3}$$

2.4 Query-aware Reasoning

In the above sections, we have described how to label the nodes for a extracted opening subgraph with relation-aware, structure-aware and distance features. Then, we can perform reasoning on the opening subgraph. In this paper, we use a query-aware graph attention neural network (GAT) to propagate neighborhood information in the opening subgraph. . Given a query $(h, r, ?)$ and a k-hop subgraph $\mathcal{G}_k(h)$, we can aggregate the neighborhood features for each node n in subgraph $\mathcal{G}_k(h)$:

$$\mathbf{e}'_n = \sum_{(r,p) \in \mathcal{N}(n)} \alpha_p^l [\mathbf{e}_p^l || \mathbf{r}_p] \mathbf{W}^l + \mathbf{b}^l \tag{4}$$

where $\mathcal{N}(n) = \{(r_p, p) | (n, r_p, p) \in \mathcal{T}_k(h)\}$ is the set of neighbors of node n, $\mathcal{T}_k(h)$ is the set of triples in $\mathcal{G}_k(h)$. $\mathbf{W}^l$ and $\mathbf{b}^l$ are trainable parameters in the l-th GAT layer, α_p^l is the attention weight for the neighbor (r_p, p) which is computed as:

$$\alpha_p^l = \sigma(\mathbf{r}_a^T([\mathbf{e}_n^l || \mathbf{r}_p || \mathbf{e}_p^l] \mathbf{W}_a^l + \mathbf{b}_a^l)) \tag{5}$$

[5]Note that k in $\mathcal{G}_k(h)$ denotes the maximum hop of the opening subgraph surrounding the center node h, while J denotes the maximum hop of structure-aware neighboring anchors for each node n in the subgraph $\mathcal{G}_k(h)$.

		WN18RR			FB15k-237			NELL-995		
		#Rel	#Ent	#Tri	#Rel	#Ent	#Tri	#R	#Ent	#Tri
V1	train	9	2746	6678	183	2000	5226	14	10915	5540
	test	9	922	1991	146	1500	2404	14	225	1034
V2	train	10	6954	18968	203	3000	12085	88	2564	10109
	test	10	2923	4863	176	2000	5092	79	4937	5521
V3	train	11	12078	32150	218	4000	22394	142	4647	20117
	test	11	5084	7470	187	3000	9137	122	4921	9668
V3	train	9	3861	9842	222	5000	33916	77	2092	9289
	test	9	7208	15157	204	3500	14554	61	3294	8520

Table 1: Statistics of the inductive datasets. "#Rel" denotes the number of relations, "#Ent" denotes the number of entities,"#Tri" denotes the number of triples.

where $\mathbf{W}_a^l$ and $\mathbf{b}_a^l$ are the trainable parameters, $\mathbf{r}_a$ is the attention embedding of the query relation r, σ is the sigmoid activation function.

To alleviate the vanishing gradient problem, we use a gated layer to integrate the node features as well as its neighborhood features.

$$\mathbf{e}_n^{l+1} = g \times \mathbf{e}_n^l + (1-g) \times \mathbf{e}_n' \tag{6}$$

where g is the gate to control the weight between the l-th layer node feature $\mathbf{e}_n^l$ and the neighborhood feature $\mathbf{e}_n'$, which is defined as:

$$g = \sigma([\mathbf{e}_n^l || \mathbf{e}_n'] \mathbf{W}_g^l + \mathbf{b}_g^l) \tag{7}$$

where $\mathbf{W}_g^l$ and $\mathbf{b}_g^l$ are learnable parameters of the gate mechanism.

In this paper, we stack query-aware GAT with L layers to propagate multi-hop neighborhood features and obtain the final node embeddings by concatenating representations of all layers, namely $\mathbf{e}_n = [\mathbf{e}_n^0 || \mathbf{e}_n^1 || \cdots || \mathbf{e}_n^L]$. Then, an average-pooling layer is applied to obtain the representation of the subgraph $\mathcal{G}_k(h)$:

$$\mathbf{e}_{\mathcal{G}} = \frac{1}{|\mathcal{E}_k|} \sum_{n \in \mathcal{E}_k} \mathbf{e}_n \tag{8}$$

Sequentially, we can compute the embedding for the given query $(h, r, ?)$ by taking into account the subgraph embedding $\mathbf{e}_{\mathcal{G}}$, the query entity embedding $\mathbf{e}_h$ and the query relation embedding $\mathbf{r}$:

$$\mathbf{e}_{query} = [\mathbf{e}_{\mathcal{G}} || \mathbf{e}_h || \mathbf{r}] \mathbf{W}_q + \mathbf{b}_q \tag{9}$$

where $\mathbf{W}_q$ and $\mathbf{b}_q$ are the trainable parameters of the query embedding layer.

Finally, the score for a candidate answer entity t can be obtained using the inner product of the query embedding $\mathbf{e}_{query}$ and the candidate answer entity embedding $\mathbf{e}_t$:

$$f(h, r, t) = \sigma(\mathbf{e}_{query}^T \mathbf{e}_t) \tag{10}$$

Note that all the candidate answer entities are in the same subgraph $\mathcal{G}_k(h)$, Therefore, we can compute the scores for all candidates by performing reasoning on the subgraph $\mathcal{G}_k(h)$ by one time. This is different from previous methods which need extract and process a specific subgraph for every candidate triple.

Following (Teru et al., 2020), we randomly sample a negative answer entity t' for each query $(h, r, ?)$ and correct answer t. The goal is to predict a higher score for each positive answer entity and lower score for the negative answer entity. Thus, we use the following cross entropy loss function to train the model:

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{G}(h)} \log(f(h, r, t')) - \log(f(h, r, t)) \tag{11}$$

For the query $(?, r, t)$, which needs to predict the missing head entity, we use a inverse form $(t, r^{-1}, ?)$ to convert it into a query with a missing tail entity.

3 Experiments

3.1 Datasets and Evaluate Metrics

To evaluate the inductive relation prediction performance of our QAAR, we conduct extensive experiments on three widely used datasets which are proposed in GraIL (Teru et al., 2020). These inductive datasets are extracted from WN18RR (Dettmers et al., 2018), FB15k-237(Toutanova et al., 2015) and NELL-995 (Xiong et al., 2017b). Each inductive dataset consists of train and test KGs and these

Models	WN18RR					FB15k-237				
	v1	v2	v3	v4	Avg.	v1	v2	v3	v4	Avg.
Neural-LP†	74.37	68.93	46.18	67.13	64.15	52.92	58.94	52.90	55.88	55.16
DRUM†	74.37	68.93	46.18	67.13	64.15	52.92	58.73	52.90	55.88	55.10
RuleN†	80.85	78.23	53.39	71.59	71.01	49.76	77.82	87.69	85.60	75.21
Meta-iKG	-	-	-	-	-	66.96	74.08	71.89	72.28	71.30
MorsE	84.14	81.50	70.92	79.61	79.04	83.17	95.67	95.69	95.89	92.61
GraIL†	82.45	78.68	58.43	73.41	73.24	64.15	81.80	82.83	89.29	79.51
CoMPILE†	83.60	79.82	60.69	75.49	74.90	67.64	82.98	84.67	87.44	80.68
TACT†	84.04	81.63	67.97	76.56	77.55	65.76	83.56	85.20	88.69	80.80
ConGLR†	85.64	**92.93**	70.74	**92.90**	85.55	68.29	85.98	88.61	89.31	82.93
NodePiece	83.00	88.60	78.50	80.70	82.70	87.30	93.90	94.40	94.90	92.35
QAAR	**88.82**	86.87	**88.18**	89.15	**88.25**	**90.48**	**95.81**	**95.87**	**96.21**	**94.59**

Table 2: Hits@10 results on the inductive benchmark datasets extracted from WN18RR and FB15k-237. † means the results are taken from (Lin et al., 2022). Other results are taken from their original papers. "Avg." denotes the average values of the four versions. The best results are in **bold** while the second best scores are in underline.

Models	NELL-995				
	v1	v2	v3	v4	Avg.
Neural-LP†	40.78	78.73	82.71	80.58	70.70
DRUM†	19.42	78.55	82.71	80.58	65.31
RuleN†	53.50	81.75	77.26	61.35	68.46
Meta-iKG	64.20	77.91	77.41	73.12	73.16
MorsE	65.20	80.70	87.67	53.44	71.75
GraIL†	59.50	93.25	91.41	73.19	79.33
CoMPILE†	58.38	93.87	92.77	75.19	80.05
TACT*	79.80	88.91	94.02	73.78	84.12
ConGLR*	81.07	94.92	94.36	81.61	87.99
NodePiece	89.00	90.10	93.60	89.30	90.50
QAAR	**89.50**	**96.32**	**94.56**	**91.15**	**92.88**

Table 3: Hits@10 results on the inductive benchmark datasets extracted from NELL-995. The best results are in **bold** while the second best scores are in underline.

two KGs have no overlapping entities. And there are four versions with different sizes for each inductive dataset. The details of these datasets are shown in Table 1.

Following the prior work (Teru et al., 2020; Lin et al., 2022), we use hits@10 as the evaluation metric and compute hits@10 by ranking each test triple against 50 negative triples and check whether the positive triple is ranked in top 10. We run each experiment five times and report the mean results.

3.2 Implementation Details

We implement our model using Pytorch. We optimize our model using Adam (Kingma and Ba, 2015). We choose the optimal hyper-parameters via grid search. We select the batch size from {4,8,16,32}, the learning rate from {0.0005,0.001,0.002}, weight decay from {1e-2,1e-3,1e-4}, dropout rate from {0.1,0.2,0.3}, the number of GAT layers from {1,2,3}, the subgraph size k from {2,3,4,5}, the maximum depth J of structure-aware features from {1,2,3}, the dimension size of output entity embedding from {30,50,80}. The best hyper-parameters are selected according to the hits@10 on validation set. Finally, the learning rate is set to 0.001, the batch size is set to 16, the weight decay is set to 1e-3, the dropout rate is set to 0.1, the number of GAT layers is set to 2, and the subgraph size k is set to 4, the maximum depth of structure features is set to $J = 2$, the dimension size of output entity embedding is set to 80.

3.3 Baselines

We compare our QAAR with the following rule-based models: Neural-LP (Yang et al., 2017), DRUM (Sadeghian et al., 2019), RuleN (Meilicke et al., 2018), and GCN-based models: Meta-iKG (Zheng et al., 2021), MorsE (Chen et al., 2022), GraIL (Teru et al., 2020), CoMPILE (Mai et al., 2021), TACT (Chen et al., 2021), ConGLR (Lin et al., 2022) and NodePiece (Galkin et al., 2022).

3.4 Experimental Results

Table 2 and Table 3 show the experimental results on the four versions from WN18RR, FB15k-237 and NELL-995 under the evaluation metric of hits@10. From Table 2 and Table 3, we can have two interesting findings: (1) The GCN-based methods, which leverage GCNs to aggregate and propagate neighborhood information, performs better than rule-based methods. This indicates that rule-based methods are insufficient to obtain good performance for inductive relation prediction due to the brittleness of logical rules. Instead, GCN-based methods are more robust to capture complex structure patterns in KGs. (2) Our QAAR achieves better performance than most of SOTA GCN-based

Models	WN18RR					FB15k-237				
	v1	v2	v3	v4	Avg.	v1	v2	v3	v4	Avg.
QAAR	88.82	86.87	88.18	89.15	88.25	90.48	95.81	95.87	96.21	94.59
QAAR w/o D	87.23	81.74	86.44	86.91	85.58	85.85	90.79	93.29	94.18	91.03
Δ	-1.59	-5.13	-1.74	-2.24	-2.67	-4.63	-5.02	-2.58	-2.03	-3.56
QAAR w/o R	86.17	86.16	86.77	89.43	87.13	84.63	92.46	93.64	95.64	91.59
Δ	-2.65	-0.71	-1.41	+0.28	-1.12	-5.85	-3.35	-2.23	-0.57	-3.00
QAAR w/o S	85.63	79.59	86.61	88.17	85.00	87.31	94.35	94.21	95.29	92.79
Δ	-3.19	-7.28	-1.57	-0.98	-3.25	-3.17	-1.46	-1.66	-0.92	-1.80
QAAR w/o A	86.96	81.51	87.76	88.38	86.15	84.39	94.97	94.91	95.36	92.41
Δ	-1.86	-5.36	-0.42	-0.77	-2.10	-6.09	-0.84	-0.96	-0.85	-2.18

Table 4: Ablation study on the inductive benchmark datasets extracted from WN18RR and FB15k-237. Δ denotes the differences between each ablation model and the original QAAR model. Δ denotes the differences between each ablation model and the original QAAR model.

Models	NELL-995				
	v1	v2	v3	v4	Avg.
QAAR	**89.50**	**96.32**	**94.56**	**91.15**	**92.88**
QAAR w/o D	87.00	90.33	90.61	4.81	88.19
Δ	-2.50	-5.99	-3.95	-6.34	-4.69
QAAR w/o R	85.50	93.27	92.83	89.19	90.20
Δ	-4.00	-3.05	-1.73	-1.96	-2.68
QAAR w/o S	82.50	92.85	92.58	90.28	88.55
Δ	-7.00	-3.47	-1.98	-0.87	-3.33
QAAR w/o A	86.00	91.38	92.09	90.15	89.91
Δ	-3.50	-4.94	-2.47	-1.00	-2.97

Table 5: Ablation study on the inductive benchmark datasets extracted from NELL-995.

methods in term of hits@10 (e.g., our model achieves the best performance on 10 out of 12 versions of the inductive datasets). On the average hits@10 metrics of these three datasets, our QAAR outperforms other SOTA models. Compared to previous SOTA method ConGLR, our QAAR can achieve 2.7%, 11.63% and 4.89% improvements on the average hits@10 metrics of WN18RR, FB15k-237 and NELL-995 datasets respectively. And compared to the SOTA method NodePiece, our QAAR outperforms NodePiece by 5.55%, 2.24% and 2.38% margins on these three datasets. Compared with previous SOTA GCN-based methods, such as GraIL, TACT, CoMPILE, ConGLR and NodePiece, the proposed QAAR is able to capture richer transferable entity-independent features (e.g., relation-aware, structure-aware and distance features) to learn the inductive embedddings for the nodes in a subgraph. The experimental results demonstrate that our QAAR is superior to current SOTA models.

3.5 Ablation Study

In this subsection, we perform an ablation study to investigate how each module in QAAR affects the performance. Specifically, for the node labeling, we use relation-aware, structure-aware and distance features to label each node. To study the contributions of these labeling strategies, we construct three variant models: "QAAR w/o R" denotes the model removing relation-aware features from QAAR model, "QAAR w/o S" denotes the model without using structure-aware features, and "QAAR w/o D" denotes the model without using distance features. Moreover, we also investigate the effect of query-aware graph attention layers by replacing it with GCN layers and obtain a variant model "QAAR w/o A".

Table 4 and Table 5 show the experimental results of ablation study on the inductive datasets regarding the evaluation metric of hits@10. From the results, we can see that the performances of all variant models have decreased to some extent. This demonstrates all modules in QAAR play important roles for the performance improvement. Concretely, different components in QAAR have different effect on these inductive datasets. For WN18RR dataset, the structure-aware and distance features have more positive impact on the performance, while the relation-aware features have less positive impact and even slightly hurt the performance on v4 of WN18RR. This is because the relations in WN18RR express little entity type information. Take the most frequent relation "*hypernym*" in WN18RR as an example, we cannot infer the entity type for it. Instead, for FB15k-237 and NELL-995 datasets, the relations express more useful entity type information, which can help to improve the performance. Take the relation "*PlaceOfBirth*" in FB15k-237 as an example, we can infer that the head entity of this relation is a person and the tail entity is a place. On average, all three types of features can contribute to the improvements of the performance. Moreover, the query-aware graph

#Neg.	WN18RR		FB15k-237		NELL-995	
	GraIL	QAAR	GraIL	QAAR	GraIL	QAAR
20	35.82	5.78	147.66	11.63	63.87	15.85
50	52.38	5.92	244.78	12.08	79.19	15.98
80	66.67	5.98	324.33	12.28	91.55	16.21
120	81.52	6.23	427.29	12.46	107.02	16.73
150	93.07	6.25	498.32	12.53	136.73	16.85

Table 6: Reasoning time for different number of candidate triples on the version v1 of inductive test datasets extracted from WN18RR, FB15k-237 and NELL-995. "#Neg." denotes the number of negative candidates to be ranked. The reasoning times are measured in seconds.

attention reasoning is also beneficial to relation prediction (QAAR vs. "QAAR w/o A"), which indicates that the query-aware graph attention layers can aggregate more relevant neighborhood information during reasoning.

3.6 Efficiency Analysis

In this section, we investigate the reasoning efficiency regarding the enclosing subgraph extraction vs. the opening subgraph extraction. We note that most previous inductive relation prediction models, such as CoMPILE (Mai et al., 2021), Meta-iKG (Zheng et al., 2021), TACT (Chen et al., 2021), ConGLR (Lin et al., 2022), are all based on the enclosing subgraph extraction on the basis of GraIL (Teru et al., 2020), and take more time to perform reasoning since they consider more additional mechanisms. Therefore, we choose GraIL as the representative baseline. Table 6 shows the reasoning time for different number of candidate triples on the version v1 of inductive test datasets extracted from WN18RR, FB15k-237 and NELL-995. It can be seen that our QAAR can spend less reasoning time than the GraIL model on these three datasets. Specifically, when the number of negative candidates is 50, the reasoning time of QAAR is about 9x faster than GraIL on WN18RR, 20x faster on FB15k-237, 5x faster on NELL-995. We can also find that the reasoning time of GraIL will dramatically increases when the number of negative candidates increases. Different from GraIL, the reasoning time of our QAAR remains at a stable level for different number of candidates. The reason is that our QAAR is able to rank all candidates in one opening subgraph with one-time reasoning, while GraIL will extract a specific enclosing subgraph for every candidate and perform multiple reasoning on all these subgraphs. These empirical results verify the claim that our QAAR can perform more efficient reasoning than previous methods.

4 Conclusion

In this paper, we propose a novel query adaptive anchor representation (QAAR) model for inductive relation prediction. First, we develop a model to rank all candidate triples in one opening subgraph. Second, we design three types of query adaptive anchors for subgraphs and extract three kinds of entity-independent features including relation-aware, structure-aware and distance features, which are useful for inductive relation prediction. Finally, we use a query-aware graph attention model to perform inductive reasoning on subgraphs. The experimental results on 12 versions of inductive relation prediction datasets demonstrate the effectiveness and efficiency of our proposed QAAR model.

Limitations

Although our QAAR achieves better performance on inductive relation prediction, it still suffers from some limitations. First, for a give query we extract a k-hop subgraph without using any sampling method, which will require large GPU memory when the extracted subgraph is large. Second, our QAAR does not leverage logical rules to enhance the performance which has shown useful in previous methods (Lin et al., 2022). We believe that our method can be further improved by incorporating logical rules. We will leave these opening issues in the future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants 61972290 and 61972173, the Fundamental Research Funds for the Central Universities (No. CCNU22QN015), and Wuhan Knowledge Innovation Project (No. 2022010801010278). This work was also supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, an NSERC CREATE award in ADERSIM[6] , the York Research Chairs (YRC) program and an ORF-RE (Ontario Research Fund-Research Excellence) award in BRAIN Alliance[7]. We thank anonymous reviewers for their thorough review comments on this paper.

[6]http://www.yorku.ca/adersim
[7]http://brainalliance.ca

References

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5184–5193.

Kurt D. 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 ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, pages 1247–1250. ACM.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. 2021. PairRE: Knowledge graph embeddings via paired relation vectors. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4360–4369. Association for Computational Linguistics.

Jiajun Chen, Huarui He, Feng Wu, and Jie Wang. 2021. Topology-aware correlations between relations for inductive link prediction in knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6271–6278. AAAI Press.

Mingyang Chen, Wen Zhang, Yushan Zhu, Hongting Zhou, Zonggang Yuan, Changliang Xu, and Huajun Chen. 2022. Meta-knowledge transfer for inductive knowledge graph embedding. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 927–937. ACM.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818.

Mikhail Galkin, Etienne G. Denis, Jiapeng Wu, and William L. Hamilton. 2022. Nodepiece: Compositional and parameter-efficient representations of large knowledge graphs. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.*, 194:28–61.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Junzhuo Li and Deyi Xiong. 2022. Kafsp: Knowledge-aware fuzzy semantic parsing for conversational question answering over a large-scale knowledge base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 461–473.

Qika Lin, Jun Liu, Fangzhi Xu, Yudai Pan, Yifan Zhu, Lingling Zhang, and Tianzhe Zhao. 2022. Incorporating context graph with logical reasoning for inductive relation prediction. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 893–903. ACM.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. 2015. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*.

Sijie Mai, Shuangjia Zheng, Yuedong Yang, and Haifeng Hu. 2021. Communicative message passing for inductive relation reasoning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4294–4302. AAAI Press.

Christian Meilicke, Manuel Fink, Yanjie Wang, Daniel Ruffinelli, Rainer Gemulla, and Heiner Stuckenschmidt. 2018. Fine-grained evaluation of rule- and

embedding-based systems for knowledge graph completion. In *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part I*, volume 11136 of *Lecture Notes in Computer Science*, pages 3–20. Springer.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4710–4723.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. DRUM: end-to-end differentiable rule mining on knowledge graphs. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15321–15331.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Komal K. Teru, Etienne G. Denis, and William L. Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9448–9457. PMLR.

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, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1499–1509.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

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

Junkang Wu, Wentao Shi, Xuezhi Cao, Jiawei Chen, Wenqiang Lei, Fuzheng Zhang, Wei Wu, and Xiangnan He. 2021. DisenKGAT: Knowledge graph embedding with disentangled graph attention network. In *in: Proceedings of the 30th ACM International Conference on Information and Knowledge Management, CIKM 2021, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 2140–2149. ACM.

Huiru Xiao, Xin Liu, Yangqiu Song, Ginny Y. Wong, and Simon See. 2022. Complex hyperbolic knowledge graph embeddings with fast fourier transform. *CoRR*, abs/2211.03635.

Zhiwen Xie, Guangyou Zhou, Jin Liu, and Jimmy Xiangji Huang. 2020. ReInceptionE: Relation-aware inception network with joint local-global structural information for knowledge graph embedding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5929–5939.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017a. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1271–1279. ACM.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017b. Deeppath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 564–573. Association for Computational Linguistics.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Fan Yang, Zhilin Yang, and William W. Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2319–2328.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 535–546. Association for Computational Linguistics.

Ningyu Zhang, Qianghuai Jia, Shumin Deng, Xiang Chen, Hongbin Ye, Hui Chen, Huaixiao Tou, Gang Huang, Zhao Wang, Nengwei Hua, and Huajun Chen. 2021. AliCG: Fine-grained and evolvable conceptual graph construction for semantic search at alibaba. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3895–3905. ACM.

Yongqi Zhang, Zhanke Zhou, Quanming Yao, and Yong Li. 2022. KGTuner: Efficient hyper-parameter search for knowledge graph embedding. In *Proceedings of the 60th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 2715–2735.

Shuangjia Zheng, Sijie Mai, Ya Sun, Haifeng Hu, and Yuedong Yang. 2021. Subgraph-aware few-shot inductive link prediction via meta-learning. *CoRR*, abs/2108.00954.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 6 Limitations

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4.1

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4.1

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A. Statistic of Datasets

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.6

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.1

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.