Enhancing Document-Level Relation Extraction by Entity Knowledge Injection

Xinyi Wang[1], Zitao Wang[1], Weijian Sun[3], and Wei Hu[1,2](✉)

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
xywang.nju@gmail.com, ztwang.nju@gmail.com, whu@nju.edu.cn
[2] National Institute of Healthcare Data Science, Nanjing University, Nanjing, China
[3] Huawei Technologies Co., Ltd., Shanghai, China
sunweijian@huawei.com

Abstract. Document-level relation extraction (RE) aims to identify the relations between entities throughout an entire document. It needs complex reasoning skills to synthesize various knowledge such as coreferences and commonsense. Large-scale knowledge graphs (KGs) contain a wealth of real-world facts, and can provide valuable knowledge to document-level RE. In this paper, we propose an entity knowledge injection framework to enhance current document-level RE models. Specifically, we introduce coreference distillation to inject coreference knowledge, endowing an RE model with the more general capability of coreference reasoning. We also employ representation reconciliation to inject factual knowledge and aggregate KG representations and document representations into a unified space. The experiments on two benchmark datasets validate the generalization of our entity knowledge injection framework and the consistent improvement to several document-level RE models.

Keywords: Relation extraction · Knowledge injection · Knowledge graph

1 Introduction

Relation extraction (RE) aims to recognize the semantic relations between entities in texts, which is beneficial to a variety of AI applications such as language understanding and knowledge graph (KG) construction. Early methods [5,34,36] mainly cope with sentence-level RE, which detects the relations in a single sentence. However, a large number of relations span across multiple sentences [32], which calls for document-level RE in recent years. Compared with sentence-level RE, document-level RE is more challenging. It needs the RE models to conduct complex reasoning, e.g., coreference reasoning, factual reasoning and logical reasoning, throughout an entire document.

Figure 1 shows a real example. A document-level RE model is asked to find the relations between three named entities *IBM Research Brazil*, *São Paulo* and

National Natural Science Foundation of China (No. 61872172).

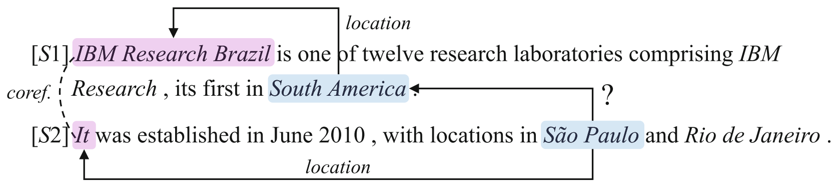
location
[S1] IBM Research Brazil is one of twelve research laboratories comprising IBM Research , its first in South America .
coref.
?
[S2] It was established in June 2010 , with locations in São Paulo and Rio de Janeiro .
location

Fig. 1. An example of document-level RE excerpted from [32]

South America. From $S1$, *IBM Research Brazil* is located in *South America* may be first recognized by the model. Then, with the help of coreference knowledge that connects the pronoun *It* in $S2$ to *IBM Research Brazil* in $S1$, the model can recognize that *IBM Research Brazil* is located in *São Paulo*. Since the model may not know the exact types of entities, only with the aid of extra knowledge in KGs like *São Paulo* is a city and *South America* is a continent, then it can confidently determine that the relation between them is *continent* rather than others. The entire reasoning process demands the document-level RE model to synthesize various knowledge and have powerful reasoning capabilities.

Recent years have witnessed that large-scale KGs, e.g., Wikidata [26] and DBpedia [1], become a valuable asset in information extraction [2,4,12,19–21,25]. A KG contains a collection of real-world facts, in which a fact is structured in the form of a triple $(entity, property, value)$. *Property* can be either an *attribute* or a *relation*, and *value* can be either a *literal* for attribute triple or an *entity* for relation triple. Particularly for the RE task, the works in [2,8,21,25] exploit one or very few attribute and relation triples (e.g., *rdfs:label*) in KGs to enhance their models. Furthermore, they overlook the heterogeneity between KG representations and document representations, and aggregate them in a simple way like vector concatenation.

In this paper, we propose a novel entity knowledge injection framework to enhance existing document-level RE models. Specifically, we introduce a general *knowledge injection layer* between the encoding layer and the prediction layer of popular RE models. Based on it, we focus on injecting various entity knowledge from KGs into the document-level RE models. We tackle two key challenges:

First, *how to inject coreference knowledge into document-level RE models?* Coreference resolution plays a vital role in RE. However, the coreferences derived from coreference resolution tools and aliases in KGs may contain errors. If we directly import them into an RE model as strong guidance information, such as the edges in a document graph [15], it is likely to bring a downside effect. Therefore, we present *coreference distillation* to distill knowledge from the coreferences and inject it into an RE model, so that the model can ultimately acquire generalized coreference knowledge.

Second, *how to inject factual knowledge into document-level RE models?* KG contains a wealth of facts related to entities, which we want to exploit for RE. However, the representations of entities in a KG and the text representations of a document are learned in two different spaces, which demand to be reconciled together. We present *representation reconciliation* to fuse KG representations and document representations into a unified space, endowing the RE model with the factual knowledge of entities.

In summary, our main contributions in this paper are twofold:

- We define a general knowledge injection framework KIRE and design various knowledge injection tasks for document-level RE, such as coreference distillation for coreference knowledge and representation reconciliation for factual knowledge. These knowledge injection and RE tasks are optimized together by multi-task learning. (Sections 3 and 4)
- We perform the experiments on two benchmark datasets DocRED [32] and DWIE [33] for document-level RE. The result comparison between seven RE models and the models after knowledge injection validates the generalization and stable improvement of our framework. (Section 5)

2 Related Work

Document-Level RE. Document-level RE has attracted vast attention in the past few years. A considerable number of studies have been conducted, which can be generally divided into graph-based models [11,13,15,24,27,31,35] as well as sequence-based models [7,18,28,30,38]. Graph-based models build document graphs to capture the semantic information in a document, and design various neural networks to carry out inference on the built document graphs. DISCREX [15] models words in a document as nodes and intra/inter-sentential dependencies as edges. Following this idea, Peng et al. [13] make use of graph LSTM while BRAN [24] employs Transformer to encode document graphs. Recently, LSR [11], GAIN [35] and GLRE [27] define more sophisticated document graphs to reserve more dependency information in a document.

Sequence-based models adopt neural encoders like BERT to implicitly capture dependencies in a document, instead of explicitly building document graphs. Wang et al. [28] use BERT to encode a document and design a two-step pipeline, which predicts whether a relation exists between two entities first, and then predicts the specific relation types. HIN [18] also makes use of BERT but design a hierarchical model that integrates the inference information from the entity, sentence and document levels. Huang et al. [7] extract three types of paths which indicate how the head and tail entities can be possibly related in the context, and predict the relations based on the extracted evidence sentences. ATLOP [38] proposes localized context pooling to transfer attentions from pre-trained language models and adaptive thresholding to resolve the multi-label and multi-entity problem. SSAN [30] modifies the attention mechanism in BERT to model the coreference and co-occurrence structures between entities, to better capture the semantic information in the context.

In this paper, our focus is injecting knowledge into these document-level RE models. Our entity knowledge injection framework KIRE is applicable to various models as long as they fall into our framework formulation.

Knowledge Injection. A few works have studied how to inject external knowledge such as a KG into the RE task for performance improvement. RESIDE [21] uses entity types and aliases while $\text{BERT}_{\text{EM+TM}}$ [4] only uses entity types. They both consider very limited features of entities. RECON [2] proposes separate models to encode attribute triples and relation triples in a KG and obtain corresponding attribute context embeddings and relation context embeddings, which are combined into sentence embeddings. KB-both [25] utilizes entity representations learned from either hyperlinked text documents (Wikipedia) or a KG (Wikidata) to raise the information extraction performance including document-level RE. Different from all above, we integrate more types of knowledge including coreferences, attributes and relations symbiotically with more effective knowledge injection methods to address the document-level RE task.

Additionally, a few studies [10,29,37] explicitly exploit incorporating knowledge from various sources such as encyclopedia knowledge, commonsense knowledge and linguistic knowledge into pre-trained language models with different injection strategies to improve the performance of language models in downstream tasks. However, the goal of these studies is orthogonal to this paper.

3 Framework Formulation

According to [18,32,38], we formulate the document-level RE task as a *multiple binary classification* problem. Given a document annotated with entities and their corresponding textual mentions, the task aims to predict the relations for each entity pair in the document, where a relation is either a predefined type (e.g., *country*) or *N/A* for no relation. Note that there may be more than one relation for an entity pair.

A basic neural network model [32] for document-level RE contains an encoding layer and a prediction layer. The encoding layer encodes an input document to obtain the context-sensitive representations of tokens (words) in it, and the prediction layer generates entity representations and predicts relations using the entity representations. In this paper, we add a *knowledge injection layer* between the encoding layer and the prediction layer, and many document-level RE models such as [27,32,38] can be used as the basic model.

We regard a KG as the knowledge source for injection. A KG is defined as a 7-tuple $\mathcal{G} = (U, R, A, V, X, Y, C)$, where U, R, A and V denote the sets of entities, relations, attributes and literal values, respectively. $X \subseteq U \times R \times U$ denotes the set of relation triples, $Y \subseteq U \times A \times V$ denotes the set of attribute triples, and C denotes the set of coreference triples derived from $\mathcal{G}$. By the alias information (e.g., *skos:altLabel*) in $\mathcal{G}$, any two aliases of an entity can constitute one coreference triple (m_s, m_t, p_{cr}), where m_s, m_t are two alias mentions and p_{cr} is the coreference probability. We employ off-the-shelf coreference resolution models to find more coreference knowledge for pronouns (e.g., *it* and *he*), possessives (e.g., *herself*), noun phrases (e.g., *this work*), etc., in the document. p_{cr} is set to the resolution confidence. Due to the main scope of this paper, we follow [10,37] and reuse entity linking tools to link the entities in the document to those in the KG.

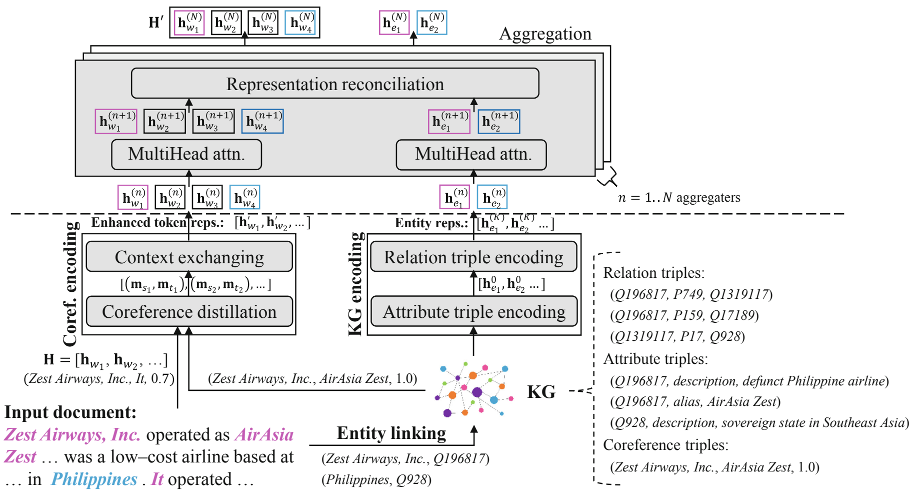
H′
Aggregation
Representation reconciliation
MultiHead attn.
MultiHead attn.
n = 1..N aggregaters
Enhanced token reps.:
Entity reps.:
Context exchanging
Coreference distillation
Coref. encoding
Relation triple encoding
Attribute triple encoding
KG encoding
H = [h_{w_1}, h_{w_2}, …]
(Zest Airways, Inc., It, 0.7)
(Zest Airways, Inc., AirAsia Zest, 1.0)
KG
Input document:
Zest Airways, Inc. operated as AirAsia Zest … was a low–cost airline based at … in Philippines . It operated …
Entity linking
(Zest Airways, Inc., Q196817)
(Philippines, Q928)
Relation triples:
(Q196817, P749, Q1319117)
(Q196817, P159, Q17189)
(Q1319117, P17, Q928)
Attribute triples:
(Q196817, description, defunct Philippine airline)
(Q196817, alias, AirAsia Zest)
(Q928, description, sovereign state in Southeast Asia)
Coreference triples:
(Zest Airways, Inc., AirAsia Zest, 1.0)

Fig. 2. Architecture of the knowledge injection layer

Framework. Given a document $\mathcal{D} = \{w_1, \dots, w_J\}$, where w_j denotes the j^{th} token in $\mathcal{D}$, and a KG $\mathcal{G}$, the framework of document-level RE with entity knowledge injection is

$$\begin{aligned} \mathbf{H} &= [\mathbf{h}_{w_1}, \dots, \mathbf{h}_{w_J}] = \text{Encode}(\mathcal{D}), \\ \mathbf{H}' &= \text{KnowledgeInject}(\mathcal{D}, \mathbf{H}, \mathcal{G}), \\ \mathbf{z} &= \text{Predict}(\mathbf{H}'), \end{aligned} \tag{1}$$

where $\mathbf{h}_{w_j}$ denotes the hidden representation of w_j, and $\mathbf{z}$ denotes the prediction probability distribution of relations. □

4 Knowledge Injection

The architecture of the proposed knowledge injection framework KIRE is depicted in Fig. 2, which accepts the document $\mathcal{D}$, the hidden representation $\mathbf{H}$ of $\mathcal{D}$ and the relevant KG $\mathcal{G}$ as input. It injects the entity knowledge from the coreference triples, attribute triples and relation triples into an RE model, and outputs the final hidden representation $\mathbf{H}'$.

Specifically, we inject the coreference triples into the basic document-level RE model with coreference distillation and context exchanging. Apart from this, the attribute triples are semantically encoded with AutoEncoder [16], and the encoded results are then reused to initialize the representations of relation triples. We use a relational graph attention network (R-GAT) [3] to encode the relation triples and generate the KG representations of entities. Finally, the KG representations of entities and the token representations that have been enhanced by coreference knowledge are aggregated by representation reconciliation. Details are described in the following subsections.

4.1 Coreference Encoding

This module leverages coreference triples to exchange the contextual information between aliases, and thus the representations of alias mentions can be closer.

Coreference Distillation. A simple method is to model the coreference triples as a new type of edges in the document graph and reuse graph-based models [11,13,24]. However, such a method cannot be generalized to the sequence-based models since they do not construct document graphs. Furthermore, the accuracy of existing coreference resolution tools is still far from perfect, even they are trained on large corpora. To alleviate error accumulation, it is inappropriate to directly add the edges as strong guidance information in the RE models.

Knowledge distillation [6,19], as a model compression technique and a solution to integrate external knowledge into a target model, has been used in a wide range of NLP tasks. In this paper, we leverage the idea of knowledge distillation and propose coreference distillation to inject coreference triples into the RE models. Our main idea is to leverage a pre-trained coreference resolution model which has been trained on a large coreference dataset as the *teacher* model, and then force the *student* model (i.e., the RE model) to generate a prediction probability distribution that approximates the teacher model on the coreference triples. Finally, the student model learns the coreference knowledge and generalization ability in the teacher model. Formally, for a coreference triple (m_s, m_t, p_{cr}), its coreference probability generated by the teacher model is defined as

$$P_{\text{tea}}(m_s, m_t) = p_{cr}. \tag{2}$$

The student model generates the coreference probability with a multi-layer perceptron (MLP):

$$P_{\text{stu}}(m_s, m_t) = \text{MLP}\Big(\big[\mathbf{m}_s; \mathbf{m}_t; \mathbf{\Delta}(\psi(m_s, m_t))\big]\Big), \tag{3}$$

where $\mathbf{m}_s$ and $\mathbf{m}_t$ denote the hidden representations of alias mentions m_s and m_t, respectively, which are calculated by averaging the hidden representations of tokens in m_s and m_t, that is, $\mathbf{m}_s = \text{avg}_{w_j \in m_s}(\mathbf{h}_{w_j}), \mathbf{m}_t = \text{avg}_{w_j \in m_t}(\mathbf{h}_{w_j})$. ";" is the concatenation operation, and $\psi(m_s, m_t)$ denotes the shortest distance between m_s, m_t in the document. We divide the distance into $\{1, 2, \ldots, 2^{\beta}\}$ bins, and associate each bin with a trainable distance vector. $\mathbf{\Delta}(\cdot)$ associates each ψ to the distance vector of relevant bin. Empirically, aliases with different distances should have different impacts on each other. Therefore, we propose trainable distance vectors to model and utilize such difference.

We enforce the student model to learn from the teacher model using the following coreference loss:

$$\mathcal{L}_{cr} = \sum_{(m_s, m_t) \in C} \text{KL}\big(P_{\text{tea}}(m_s, m_t) \parallel P_{\text{stu}}(m_s, m_t)\big), \tag{4}$$

where $\text{KL}(\cdot)$ is the Kullback-Leibler divergence.

Context Exchanging. Based on the learned coreference knowledge, we further enable each alias mention to interact with its most similar counterpart, so as to exchange the semantic information between them. Specifically, given an alias mention m_s, we update its hidden representation through $\mathbf{m}_s = \mathbf{m}_s + \mathbf{m}_{t^*}$, where $t^* = \arg\max_t \big\{P_{\text{stu}}(m_s, m_t) \mid (m_s, m_t) \in C\big\}$. In this way, the representations of the pronouns in particular can be enriched via their referents.

Finally, we obtain the token representations enhanced by coreference knowledge through the representations of alias mentions (if exists):

$$\mathbf{h}'_{w_j} = \begin{cases} \mathbf{m}_s, & \text{if } w_j \in m_s \\ \mathbf{h}_{w_j}, & \text{otherwise} \end{cases}. \tag{5}$$

In coreference encoding, the MLP contains $d_{\text{MLP}}(2d_{\text{token}} + d_{\text{dist}}) + 2d_{\text{token}}$ parameters, where $d_{\text{MLP}}, d_{\text{token}}, d_{\text{dist}}$ are the dimensions of MLP hidden layers, token representations and trainable distance vectors, respectively.

4.2 Knowledge Graph Encoding

This module aims to encode the attribute triples and the relation triples to generate the KG representations of entities.

Attribute Triple Encoding. A KG defines a set of common attributes, e.g., *rdfs:label* and *schema:description*, to describe its entities. We encode the attribute triples in the KG and generate the attribute representations for corresponding entities. For each attribute triple of an entity, we concatenate the attribute name a and attribute value v into a token sequence $q = [a; v] = (w_1, \ldots, w_M)$. In order to cope with the out-of-vocabulary problem, we define a lookup function to convert each token to a token embedding:

$$\text{LP}(w_j) = \begin{cases} \text{WordEmb}(w_j), & \text{if } w_j \text{ has word emb.} \\ \text{CharEmb}(w_j), & \text{otherwise} \end{cases}, \tag{6}$$

where WordEmb$(\cdot)$ returns the word embedding in GloVe, and CharEmb$(\cdot)$ offers the average of character embeddings pre-trained with Skip-gram. Our method can work with other word or character embeddings easily.

Next, we leverage AutoEncoder to encode a sequence of token embeddings into an attribute triple embedding in an unsupervised way:

$$\mathbf{q} = \text{AutoEncoder}\Big(\big[\text{LP}(w_1); \ldots; \text{LP}(w_M)\big]\Big), \tag{7}$$

where AutoEncoder is pre-trained on the attribute triples. We conduct self-supervised training, and both encoder and decoder of AutoEncoder use BiLSTM. AutoEncoder has good capacity for feature extraction and compression. In our model, the input of AutoEncoder is a concatenation vector of an entity and its attributes. The reconstruction loss of AutoEncoder can help extract a better compressed feature representation while preserving the attribute knowledge.

Finally, we stack all attribute triple embeddings of an entity into a one-dimensional CNN to obtain the attribute representation of the entity:

$$\mathbf{h}_{e_i}^0 = \text{MaxPooling}\big(\text{CNN}_{\text{1D}}(\|_j\ \mathbf{q}_j)\big), \tag{8}$$

where $\|$ denotes the stack operation, and $\mathbf{h}_{e_i}^0$ is the attribute representation of entity e_i, which would be used as the input representation for relation triple encoding below. Here, we choose CNN since the convolutional layer is a good feature extractor to learn high-level representations from value embeddings while reducing the dimension of output representations. Furthermore, we use the 1D convolution kernel as its invariance to the order of attribute embeddings.

Relation Triple Encoding. The relation triples present in the form of an entity-relation graph structure, and the topology and relation types are the key to encode such knowledge. Based on the attribute representations of entities, we employ a R-GAT [3] with K layers to convolute the entity-relation graph. R-GAT incorporates relation types using different embeddings and calculates attention scores on all adjacent nodes based on entity embeddings and relation embeddings. Specifically, the node forward-pass update for the $(k+1)^{\text{th}}$ layer is

$$\begin{aligned}
\mathbf{e}_{ij}^{(k,b)} &= \mathbf{W}_{\text{out}}^{(k,b)^T}\big[\mathbf{W}_{\text{in}}^{(k,b)}\mathbf{h}_i^{(k)}; \mathbf{W}_{\text{in}}^{(k,b)}\mathbf{h}_j^{(k)}; \mathbf{M}(r_{ij})\big],\\
\alpha_{ij}^{(k,b)} &= \frac{\exp\big(\text{LeakyReLU}(\mathbf{e}_{ij}^{(k,b)})\big)}{\sum_{l\in U_i}\exp\big(\text{LeakyReLU}(\mathbf{e}_{il}^{(k,b)})\big)},\\
\mathbf{h}_i^{(k+1)} &= \frac{1}{B}\sum_{b=1}^{B}\sigma\Big(\sum_{l\in U_i}\alpha_{il}^{(k,b)}\mathbf{W}_{\text{in}}^{(k,b)}\mathbf{h}_l^{(k)}\Big),
\end{aligned} \tag{9}$$

where $\mathbf{W}_{\text{in}}^{(k,b)}$ and $\mathbf{W}_{\text{out}}^{(k,b)}$ denote two trainable parameters of the b^{th} attention head ($1 \leq b \leq B$) at the k^{th} layer. $\mathbf{h}_i^{(k)}$ and $\mathbf{h}_j^{(k)}$ are the node representations of entities e_i and e_j at the k^{th} layer, respectively. $\mathbf{M}$ is a trainable mapping matrix corresponding to the relation types in the KG. r_{ij} is the relation type between e_i and e_j. LeakyReLU$(\cdot)$ and $\sigma(\cdot)$ are the activation functions. U_i is the neighbor set of e_i. In this way, the entity representations are updated via their all adjacent entity embeddings and relation embeddings at the previous layer.

We refer to the representations of entities after graph convolution as the *KG representations* of entities, which encode the knowledge in both attribute triples and relation triples. We simply denote the KG representation of entity e_i by $\mathbf{h}_{e_i}$.

In the KG encoding, the attribute encoding has $d_{\text{Auto}}N_{\text{max}}N_{\text{kernel}}(d_{\text{kernel}}^2+1)$ parameters, where d_{Auto} is the output dimension of AutoEncoder, N_{max} is the maximum number of attributes that an entity has, N_{kernel} is the number of kernels, and d_{kernel} is the kernel size of CNN. The R-GAT network in the relation triple encoding contains $2(N_{\text{layer}}-1)N_{\text{head}}d_{\text{RGAT}}^2+d_{\text{RGAT}}d_{\text{ent}}$ parameters, where N_{layer} is the number of layers in R-GAT, N_{head} is the number of attention heads at each layer of R-GAT, d_{RGAT} is the hidden dimension of R-GAT hidden layers, and d_{ent} is the dimension of entity representations.

4.3 Representation Reconciliation

The token representations and the KG representations of entities capture different knowledge in independent semantic spaces. Following ERNIE [37] and K-BERT [10], we employ a representation reconciliation module to exchange the knowledge from entities in the KG with their linked tokens in the document.

The representation reconciliation module consists of N-stacked aggregators. For the n^{th} $(1 \leq n \leq N)$ aggregator, given the token representations $\{\mathbf{h}_{w_1}^{n-1}, \ldots, \mathbf{h}_{w_J}^{n-1}\}$ and the KG representations of entities $\{\mathbf{h}_{e_1}^{n-1}, \ldots, \mathbf{h}_{e_I}^{n-1}\}$ from the preceding aggregator, the fusion phase is formulated as

$$\tilde{\mathbf{h}} = \begin{cases} \sigma(\tilde{\mathbf{W}}_w^{(n)}\tilde{\mathbf{h}}_{w_j}^{(n)} + \tilde{\mathbf{W}}_e^{(n)}\tilde{\mathbf{h}}_{e_i}^{(n)} + \tilde{\mathbf{b}}^{(n)}), & \text{if } w_j, e_i \text{ align} \\ \sigma(\tilde{\mathbf{W}}_w^{(n)}\tilde{\mathbf{h}}_{w_j}^{(n)} + \tilde{\mathbf{b}}^{(n)}), & \text{otherwise} \end{cases}, \tag{10}$$

where $\tilde{\mathbf{h}}_{w_j}^{(n)}, \tilde{\mathbf{h}}_{e_i}^{(n)}$ are the token representation and KG representation after the multi-head self-attention [22], respectively. $\tilde{\mathbf{W}}_w^{(n)}, \tilde{\mathbf{W}}_e^{(n)}, \tilde{\mathbf{b}}^{(n)}$ are three trainable parameters. The information in the two semantic spaces is mutually integrated.

Then, the reconstruction phase leverages $\tilde{\mathbf{h}}$ to refine the output representations of each token and entity in the aligned token-entity pairs:

$$\begin{aligned} \mathbf{h}_{w_j}^{(n)} &= \sigma(\mathbf{W}_w^{(n)}\tilde{\mathbf{h}} + \mathbf{b}_w^{(n)}), \\ \mathbf{h}_{e_i}^{(n)} &= \sigma(\mathbf{W}_e^{(n)}\tilde{\mathbf{h}} + \mathbf{b}_e^{(n)}). \end{aligned} \tag{11}$$

Here, the aligned token representations and entity representations are updated and enhanced by the integrated information. Note that the representations of entities without aligned tokens would not be updated.

Finally, we obtain the token representation sequence $\{\mathbf{h}_{w_1}^N, \ldots, \mathbf{h}_{w_J}^N\}$ from the last aggregator, which would constitute $\mathbf{H}'$ and be fed to the prediction layer.

To supervise the above process, we employ a token-entity alignment task. For each aligned token-entity pair (w_j, e_i), we predict the aligned KG entity e_i based on the token w_j. We only ask the model to predict entities within a given entity candidate set. By default, all linked entities in the document form the candidate set. For the token sequence $\{w_1, \ldots, w_J\}$ and the corresponding candidate entities $\{e_1, \ldots, e_I\}$, the token-entity alignment loss is

$$\mathcal{L}_{kg} = \sum_{j=1}^{J}\sum_{i=1}^{I} f_{j,i}^{*} * P(e_i \,|\, w_j), \tag{12}$$

where $f_{j,i}^{*} \in \{0, 1\}$ is the true alignment label between w_j and e_i, and $P(e_i \,|\, w_j) = \frac{\exp\left(\text{Linear}(\mathbf{h}_{w_j}^{(N)}) \cdot \mathbf{h}_{e_i}\right)}{\sum_{l=1}^{I}\exp\left(\text{Linear}(\mathbf{h}_{w_j}^{(N)}) \cdot \mathbf{h}_{e_l}\right)}$ returns the probability that e_i can be predicted by w_j.

We optimize the RE loss, coreference loss and token-entity alignment loss with multi-task learning. The final loss is

$$\mathcal{L} = \alpha_1 \cdot \mathcal{L}_{re} + \alpha_2 \cdot \mathcal{L}_{cr} + \alpha_3 \cdot \mathcal{L}_{kg}, \tag{13}$$

where α_1, α_2 and α_3 are the weight hyperparameters.

Table 1. Dataset statistics. Inst. denotes relation instances excluding N/A relation.

Datasets		#Doc	#Rel	#Inst	#N/A Inst
DocRED	Training set	3,053	96	38,269	1,163,035
	Validation set	1,000	96	12,332	385,263
	Test set	1,000	96	12,842	379,316
DWIE	Training set	544	66	13,524	492,057
	Validation set	137	66	3,488	121,750
	Test set	96	66	2,453	78,995

In the representation reconciliation, for each aggregator, the multi-head self-attention networks contain $4d_{\text{token}}^2 + 4d_{\text{ent}}^2$ parameters, the fusion phase contains $d_{\text{out}}(N_{\text{token}} + N_{\text{align}}) + N_{\text{token}}$ parameters, and the reconstruction phase contains $2N_{\text{align}}(d_{\text{out}} + 1)$ parameters, where d_{out} is the output dimension of multi-head self-attention networks, N_{token} is the number of tokens, and N_{align} is the number of aligned token-entity pairs. Therefore, the parameters of representation reconciliation are $N_{\text{agg}}\big[4d_{\text{token}}^2 + 4d_{\text{ent}}^2 + d_{\text{out}}(N_{\text{token}} + N_{\text{align}}) + N_{\text{token}} + 2N_{\text{align}}(d_{\text{out}} + 1)\big]$, where N_{agg} is the number of aggregators.

Model Complexity. The total parameter number of KIRE is $d_{\text{MLP}}(2d_{\text{token}} + d_{\text{dist}}) + 2d_{\text{token}} + d_{\text{Auto}}N_{\max}(d_{\text{kernel}}^2 + 1) + 2(N_{\text{layer}} - 1)N_{\text{head}}d_{\text{RGAT}}^2 + d_{\text{RGAT}}d_{\text{ent}} + N_{\text{agg}}\big[4d_{\text{token}}^2 + 4d_{\text{ent}}^2 + d_{\text{out}}(N_{\text{token}} + N_{\text{align}}) + N_{\text{token}} + 2N_{\text{align}}(d_{\text{out}} + 1)\big]$.

5 Experiments and Results

We develop KIRE with PyTorch 1.7.1, and test on an X86 server with two Xeon Gold 5117 CPUs, 250 GB memory, two Titan RTX GPUs and Ubuntu 18.04.

5.1 Experiment Setup

Datasets. We select two benchmark datasets in our experiments: (1) DocRED [32] is a crowdsourced dataset for document-level RE. The relation labels in its test set are not public. (2) DWIE [33] is a new dataset for document-level multi-task information extraction. We use the data relevant to RE only. Since DWIE does not have the validation set, we randomly split its training set into 80% for training and 20% for validation. Table 1 lists the statistical data.

Knowledge Graph. We select Wikidata (2020–12-01) as our KG due to its coverage and popularity [2,21]. The numbers of its relation and attribute triples are 506,809,195 and 729,798,070, respectively. To prevent the test leakage, we filter out all the relation triples with entity pairs to be labeled in the test sets.

Evaluation Metrics. We measure F1-score (F1) and Ignore F1-score (Ign F1) in our experiments. We repeat five times using the same hyperparameters but different random seeds, and report the means and standard deviations.

Implementation Details. To achieve good generalization, we do not carry out excessive feature engineering on Wikidata. Numerical attributes are regarded as texts, and their semantics are captured by the word embeddings [14]. We employ NeuralCoref 4.0 as our coreference resolution tool and also use the annotations provided in DocRED and DWIE. We use a two-stage method for training, which first trains a basic RE model and then fine-tunes this model to train the knowledge injection layer. The training procedure is optimized with Adam. Moreover, to compare fairly, the basic RE model and its corresponding KIRE adopt the same hyperparameter values. We set the batch size to 4 and the learning rate to 0.0005. We use three R-GAT layers and two aggregators. Moreover, $\alpha_1, \alpha_2, \alpha_3$ are 1, 0.01 and 0.01, respectively. The dimension of hidden layers in MLP is 256, the dimensions of GloVe and Skip-gram are 100, and the dimension of hidden layers in AutoEncoder is 50. See the source code for more details.

Table 2. Comparison of result improvement on baseline models

Models	DocRED				DWIE			
	Validation set		Test set		Validation set		Test set	
	Ign F1	F1	Ign F1	F1	Ign F1	F1	Ign F1	F1
CNN	43.91$_{\pm 0.02}$	45.99$_{\pm 0.05}$	42.61$_{\pm 0.06}$	44.80$_{\pm 0.08}$	38.21$_{\pm 0.04}$	49.09$_{\pm 0.06}$	40.06$_{\pm 0.08}$	51.21$_{\pm 0.13}$
+ KIRE	**46.18**$_{\pm 0.04}$	**48.21**$_{\pm 0.06}$	**45.24**$_{\pm 0.05}$	**47.27**$_{\pm 0.08}$	**39.68**$_{\pm 0.05}$	**50.49**$_{\pm 0.09}$	**42.09**$_{\pm 0.04}$	**53.16**$_{\pm 0.08}$
+ RESIDE	45.03$_{\pm 0.06}$	47.12$_{\pm 0.08}$	43.79$_{\pm 0.05}$	45.96$_{\pm 0.09}$	39.24$_{\pm 0.04}$	50.13$_{\pm 0.07}$	41.31$_{\pm 0.06}$	52.47$_{\pm 0.11}$
+ RECON	45.57$_{\pm 0.04}$	47.64$_{\pm 0.07}$	44.53$_{\pm 0.07}$	46.68$_{\pm 0.10}$	39.42$_{\pm 0.03}$	50.34$_{\pm 0.06}$	41.73$_{\pm 0.07}$	52.74$_{\pm 0.09}$
+ KB-graph	45.49$_{\pm 0.03}$	47.58$_{\pm 0.08}$	44.46$_{\pm 0.06}$	46.61$_{\pm 0.09}$	39.34$_{\pm 0.06}$	50.26$_{\pm 0.09}$	41.65$_{\pm 0.08}$	52.63$_{\pm 0.12}$
LSTM	48.49$_{\pm 0.05}$	50.41$_{\pm 0.07}$	47.41$_{\pm 0.04}$	49.47$_{\pm 0.10}$	52.79$_{\pm 0.03}$	63.61$_{\pm 0.08}$	54.87$_{\pm 0.07}$	65.17$_{\pm 0.14}$
+ KIRE	**50.41**$_{\pm 0.03}$	**52.49**$_{\pm 0.06}$	**49.55**$_{\pm 0.06}$	**51.72**$_{\pm 0.09}$	**54.11**$_{\pm 0.04}$	**64.86**$_{\pm 0.08}$	**56.74**$_{\pm 0.05}$	**66.91**$_{\pm 0.07}$
+ RESIDE	49.58$_{\pm 0.04}$	51.49$_{\pm 0.08}$	48.52$_{\pm 0.06}$	50.51$_{\pm 0.09}$	53.87$_{\pm 0.02}$	64.56$_{\pm 0.06}$	55.96$_{\pm 0.06}$	66.29$_{\pm 0.12}$
+ RECON	50.03$_{\pm 0.03}$	51.98$_{\pm 0.08}$	49.07$_{\pm 0.07}$	51.12$_{\pm 0.12}$	53.98$_{\pm 0.03}$	64.69$_{\pm 0.07}$	56.35$_{\pm 0.04}$	66.51$_{\pm 0.08}$
+ KB-graph	49.94$_{\pm 0.04}$	51.89$_{\pm 0.07}$	48.98$_{\pm 0.05}$	51.04$_{\pm 0.09}$	53.91$_{\pm 0.05}$	64.61$_{\pm 0.08}$	56.27$_{\pm 0.06}$	66.43$_{\pm 0.09}$
BiLSTM	48.51$_{\pm 0.04}$	50.54$_{\pm 0.08}$	47.58$_{\pm 0.05}$	49.66$_{\pm 0.11}$	53.95$_{\pm 0.05}$	63.96$_{\pm 0.07}$	54.91$_{\pm 0.09}$	65.39$_{\pm 0.11}$
+ KIRE	**50.46**$_{\pm 0.02}$	**52.65**$_{\pm 0.05}$	**49.69**$_{\pm 0.04}$	**51.98**$_{\pm 0.07}$	**55.86**$_{\pm 0.05}$	**65.77**$_{\pm 0.09}$	**56.88**$_{\pm 0.05}$	**67.02**$_{\pm 0.08}$
+ RESIDE	49.64$_{\pm 0.03}$	51.59$_{\pm 0.06}$	48.62$_{\pm 0.04}$	50.71$_{\pm 0.10}$	55.04$_{\pm 0.06}$	65.01$_{\pm 0.09}$	56.16$_{\pm 0.05}$	66.47$_{\pm 0.12}$
+ RECON	49.97$_{\pm 0.04}$	52.06$_{\pm 0.07}$	49.14$_{\pm 0.06}$	51.32$_{\pm 0.09}$	55.42$_{\pm 0.04}$	65.38$_{\pm 0.08}$	56.51$_{\pm 0.06}$	66.63$_{\pm 0.09}$
+ KB-graph	49.89$_{\pm 0.03}$	51.98$_{\pm 0.07}$	49.05$_{\pm 0.05}$	51.26$_{\pm 0.08}$	55.35$_{\pm 0.03}$	65.31$_{\pm 0.09}$	56.42$_{\pm 0.07}$	66.55$_{\pm 0.11}$
Context-aware	49.79$_{\pm 0.03}$	51.84$_{\pm 0.04}$	48.73$_{\pm 0.07}$	50.91$_{\pm 0.12}$	54.68$_{\pm 0.04}$	64.29$_{\pm 0.06}$	56.53$_{\pm 0.07}$	65.91$_{\pm 0.09}$
+ KIRE	**51.07**$_{\pm 0.03}$	**53.25**$_{\pm 0.07}$	**50.43**$_{\pm 0.05}$	**52.75**$_{\pm 0.10}$	**56.58**$_{\pm 0.03}$	**65.62**$_{\pm 0.07}$	**58.41**$_{\pm 0.04}$	**67.37**$_{\pm 0.08}$
+ RESIDE	50.43$_{\pm 0.04}$	52.59$_{\pm 0.07}$	49.58$_{\pm 0.05}$	51.86$_{\pm 0.09}$	55.74$_{\pm 0.03}$	65.11$_{\pm 0.07}$	57.64$_{\pm 0.05}$	66.78$_{\pm 0.08}$
+ RECON	50.78$_{\pm 0.03}$	52.89$_{\pm 0.06}$	49.97$_{\pm 0.04}$	52.27$_{\pm 0.08}$	56.12$_{\pm 0.05}$	65.48$_{\pm 0.08}$	58.02$_{\pm 0.06}$	66.94$_{\pm 0.10}$
+ KB-graph	50.69$_{\pm 0.05}$	52.81$_{\pm 0.07}$	49.88$_{\pm 0.06}$	52.19$_{\pm 0.11}$	56.03$_{\pm 0.04}$	65.39$_{\pm 0.09}$	57.94$_{\pm 0.05}$	66.89$_{\pm 0.11}$

5.2 Main Results

Improvement on Baseline Models. To validate the effectiveness and versatility of KIRE, we pick four baseline models in [32]. The first three models directly employ CNN, LSTM and BiLSTM to encode documents, while the fourth model is called context-aware, which leverages the attention mechanism with BiLSTM. These four models are native to the DocRED dataset and widely chosen as the competitors in many RE studies [7,11,18,27,28,30,35,38].

Table 2 depicts the result improvement, and we observe that: (1) KIRE consistently improves the performance of all baselines on DocRED and DWIE, which demonstrates the good generalization of KIRE. Small standard deviations also tells the good stability of KIRE. (2) KIRE obtains a significant improvement of Ign F1/F1 up to 2.63/2.47 on DocRED and 2.03/1.95 on DWIE, respectively. This is mainly because the ways that the baseline models encode a document are too simple to capture some part of important contextual information in the document. External knowledge from KIRE makes up for this part, and therefore effectively improves the model performance. (3) CNN performs poorly, because the text order is important for RE while CNN cannot process such order well.

Table 3. Result improvement on state-of-the-art models

Models	DocRED				DWIE			
	Validation set		Test set		Validation set		Test set	
	Ign F1	F1	Ign F1	F1	Ign F1	F1	Ign F1	F1
ATLOP	$59.25_{\pm 0.03}$	$61.14_{\pm 0.07}$	$58.32_{\pm 0.05}$	$60.44_{\pm 0.08}$	$69.12_{\pm 0.04}$	$76.32_{\pm 0.09}$	$73.85_{\pm 0.08}$	$80.38_{\pm 0.12}$
+KIRE	$\mathbf{59.58}_{\pm 0.04}$	$\mathbf{61.45}_{\pm 0.09}$	$\mathbf{59.35}_{\pm 0.06}$	$\mathbf{61.39}_{\pm 0.11}$	$\mathbf{69.75}_{\pm 0.05}$	$\mathbf{76.75}_{\pm 0.08}$	$\mathbf{74.43}_{\pm 0.07}$	$\mathbf{80.73}_{\pm 0.15}$
SSAN	$56.68_{\pm 0.03}$	$58.95_{\pm 0.04}$	$56.06_{\pm 0.05}$	$58.41_{\pm 0.06}$	$51.80_{\pm 0.05}$	$62.87_{\pm 0.10}$	$57.49_{\pm 0.09}$	$67.77_{\pm 0.12}$
+KIRE	$57.29_{\pm 0.05}$	$59.31_{\pm 0.06}$	$56.31_{\pm 0.06}$	$58.65_{\pm 0.08}$	$52.67_{\pm 0.06}$	$63.64_{\pm 0.10}$	$60.57_{\pm 0.09}$	$69.58_{\pm 0.12}$
GLRE	$56.57_{\pm 0.06}$	$58.43_{\pm 0.09}$	$55.40_{\pm 0.07}$	$57.40_{\pm 0.13}$	$63.11_{\pm 0.03}$	$71.21_{\pm 0.06}$	$62.95_{\pm 0.05}$	$72.24_{\pm 0.09}$
+KIRE	$57.31_{\pm 0.05}$	$59.45_{\pm 0.10}$	$56.54_{\pm 0.09}$	$58.49_{\pm 0.14}$	$65.17_{\pm 0.05}$	$71.68_{\pm 0.09}$	$64.32_{\pm 0.06}$	$73.35_{\pm 0.11}$

Comparison with Existing Knowledge Injection Models. We choose three recent models: RESIDE [21], RECON [2] and KB-graph [25], which inject extra knowledge into RE models. Specifically, we use KB-graph instead of the full version KB-both since it selects Wikipedia as another knowledge source, which is unfair to other models. To compare fairly, we only adopt the knowledge injection modules of the above models to enhance the token representations in the documents, and the representations are used by the baseline RE models to predict the relation labels.

Table 2 presents the comparison results, and we obtain several findings: (1) KIRE is consistently superior to RESIDE, RECON and KB-graph with an improvement of Ign F1/F1 up to 0.71/0.66 on DocRED and 0.39/0.43 on DWIE, respectively. Given that the test sets contain (ten) thousand relation instances, we think that the improvement makes sense. For example, on the validation set of DocRED, KIRE can correctly predict an average of 478 more instances than the second best method RECON. Such improvement brought by KIRE attributes to that KIRE absorbs more knowledge like coreferences and fuses the knowledge better. (2) The improvement brought by RESIDE is the lowest since it only injects limited knowledge like entity types and relation aliases. RECON and KB-graph explore more knowledge from the KG, but they still ignore the coreference knowledge. Besides, the methods that they employ to integrate knowledge are representation average or concatenation, which may lose part of semantic information in the injected knowledge.

Improvement on State-of-the-Art Models. We employ two sequence-based models, ATLOP [38] and SSAN [30], as well as a graph-based model, GLRE [27], due to their good performance and open source. Enhancing these models is very challenging, since they have already explored various information in the documents and achieved state-of-the-art results. Due to the limit of GPU RAM, we use the BERT-base versions of ATLOP, SSAN and GLRE and re-run them according to the hyperparameters reported in their papers and source code.

The result improvement is shown in Table 3, and we have several findings: (1) For the two sequence-based models, KIRE obtains an improvement of Ign F1/F1 up to 1.03/0.95 on DocRED and 3.08/1.81 on DWIE, respectively. This mainly attributes to the fact that the extra knowledge injected by KIRE can effectively help the models identify and capture more interactions between entity pairs especially across sentences. (2) For the graph-based model, KIRE obtains an improvement of Ign F1/F1 up to 1.14/1.09 on DocRED and 1.37/1.11, respectively. This is largely due to the fact that the extra knowledge injected by KIRE can enrich the representations of mention nodes and entity nodes in the document graphs for more accurate reasoning between entity pairs especially of longer distance. (3) This also verifies that our knowledge injection framework can be generalized to a broad range of document-level RE models.

5.3 Detailed Analysis

Ablation Study. We conduct an ablation study on the four baseline models. For "w/o distill", we disable the coreference distillation module and directly use the original coreferences as the injected knowledge. For "w/o attr.", we initialize the relation triple representations by max pooling the word embeddings of entity labels. For "w/o rel.", we directly adopt the attribute representations of entities as KG representations. For "w/o KG", we disable the whole KG encoding module. Additionally, we replace KIRE with three simple variants for knowledge injection. For "w/rep. avg", we average the hidden representations of alias mentions, and the token representations are averaged with KG representations of entities. For "w/rep. concat", we concatenate the representations of alias mentions, and the KG representations of entities are concatenated after the aligned token representations. For "w/MLP", we leverage two MLP layers to fuse the representations of alias mentions and the KG representations of entities with the aligned token representations, respectively.

From Fig. 3, we can see that: (1) Ign F1/F1-scores reduce when we disable any modules, showing their contributions. (2) The changes caused by removing one type of knowledge are not obvious, mainly due to the crossovers among the three types of knowledge in the information space. (3) The results decline if we disable the coreference distillation, due to the coreference errors in the injected knowledge. (4) If we remove the KG encoding, the results drop drastically, as the baseline models cannot generate extra relation and attribute knowledge. (5) Compared to the three variants, the larger increase brought by KIRE validates the effectiveness of coreference distillation and representation reconciliation.

Influence of Mention Number. We measure the effectiveness of KIRE w.r.t. average mention number for each entity pair. For DocRED, we evaluate it on the validation set. The results are shown in Table 4. We observe that KIRE gains higher performance for the entity pairs with more mentions, in particular when the average mention number > 3. This is because KIRE injects knowledge into the RE models by updating the token representations of entity mentions, which has a greater impact on the entities with more mentions.

Comparison with Alternative Graph Encoders. We compare R-GAT with GCN [9], GAT [23] and R-GCN [17]. We remove the coreference encoding and attribute triple encoding to eliminate their interference. From Fig. 4, the performance of R-GCN and R-GAT is better than GCN and GAT, as they can capture the relation information in the entity-relation graphs. The results of

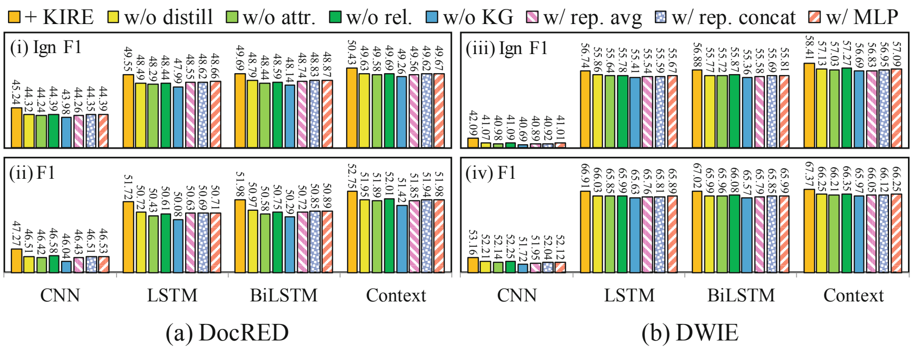

Fig. 3. Results of ablation study

Table 4. Results w.r.t. average mention number

Models	DocRED						DWIE					
	1		$(1, 3]$		> 3		1		$(1, 3]$		> 3	
	Ign F1	F1	Ign F1	F1	Ign F1	F1	Ign F1	F1	Ign F1	F1	Ign F1	F1
CNN	42.24	44.27	44.42	46.45	45.38	47.35	39.35	50.44	40.81	51.96	41.94	53.05
+ KIRE	↑1.64	↑1.75	↑2.71	↑2.89	↑3.21	↑3.43	↑1.52	↑1.49	↑2.35	↑2.32	↑2.95	↑2.98
LSTM	47.39	49.35	48.59	50.53	50.09	52.08	54.14	64.41	55.97	66.14	57.21	67.55
+ KIRE	↑1.76	↑1.87	↑2.65	↑2.82	↑3.37	↑3.51	↑1.31	↑1.26	↑1.87	↑1.89	↑2.35	↑2.29
BiLSTM	47.35	49.32	48.61	50.54	50.21	52.28	54.07	64.61	56.01	66.25	57.30	67.79
+ KIRE	↑1.83	↑1.95	↑2.73	↑2.91	↑3.31	↑3.43	↑1.49	↑1.12	↑1.91	↑1.86	↑2.34	↑2.16
Context-aware	48.33	50.19	49.63	51.64	51.10	53.34	55.98	65.16	58.02	67.01	58.72	68.41
+ KIRE	↑1.43	↑1.56	↑2.35	↑2.47	↑2.87	↑2.98	↑1.35	↑0.91	↑1.92	↑1.57	↑2.32	↑1.95

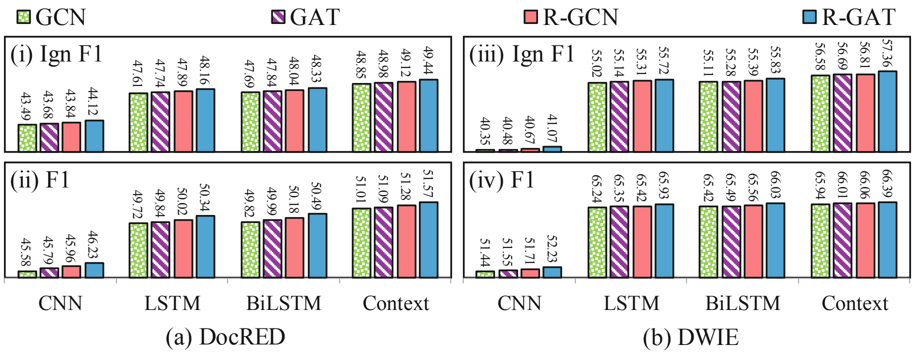

Fig. 4. Result comparison of graph encoders

GAT are greater than GCN, as GAT can selectively aggregate the neighboring information by self-attention. Similarly, R-GAT slightly outperforms R-GCN.

Case Study. We depict two successful cases and a failed case in Table 5. We still use CNN, LSTM, BiLSTM and Context-aware as baselines.

Table 5. Case study. ***Target entities*** and ***related entities*** are colored.

[$S1$]	***The Waterloo Moraine*** ... was created as a moraine in the ***Regional Municipality of Waterloo***, in Ontario, Canada.
[$S2$]	***It*** covers ... and some parts of the townships of Wellesley and ***North Dumfries***.
Case 1	Gold: *P131* Baseline models: *N/A* +KIRE: *P131*
[$S1$]	The news that British's Prince Harry is engaged to his partner ***Meghan Markle*** has attracted widespread attention from England, ***America*** and around the world.
[$S10$]	***Markle***'s parents ***Thomas Markle*** and Doria Ragland said in a statement: ...
Case 2	Gold: *citizen_ of* Baseline models: *N/A* +KIRE: *citizen_ of*
[$S1$]	***Robert Kingsbury Huntington*** ... was a naval aircrewman and member of Torpedo Squadron 8 (or VT-8).
[$S2$]	***He*** was radioman/gunner to ***Ensign George Gay***'s TBD Devastator aircraft ...
[$S4$]	Born in Los Angeles ... ***he*** was enlisted in ***the United States Navy*** 21 Apr. 1941.
Case 3	Gold: *P241* Baseline models: *N/A* +KIRE: *N/A*

– **Case 1.** To identify the relation between *North Dumfries* in $S2$ and *Regional Municipality of Waterloo* in $S1$, we use the extra knowledge (*Regional Municipality of Waterloo*, *instance of*, *regional municipality of Ontario*) and the coreference of *It* and *The Waterloo Moraine* from NeuralCoref to correctly infer the relation *P131 (located in the administrative territorial entity).*
– **Case 2.** With the aid of the extra knowledge (*Meghan Markle*, *country of citizenship*, *United States of America*) and the coreference of *Meghan Markle* and *Markle* from NeuralCoref, we successfully detect the relation *citizen_ of* between *Thomas Markle* in $S10$ and *America* in $S1$.

– **Case 3.** To recognize the relation between *Ensign George Gay* in $S2$ and *the United States Navy* in $S4$, we require a bridge entity *Robert Kingsbury Huntington*. Through the coreference of *Robert* and *He* from NeuralCoref, we identify that *Robert* and *George* are comrades. Then from $S4$, we find out that *Robert* is enlisted in *the US Navy*. According to this reasoning chain, we can see that the relation between *George* and *the US Navy* is *P241 (military branch)*. KIRE fails to run such complex reasoning involving three sentences.

6 Conclusion

In this paper, we propose KIRE, an entity knowledge injection framework for enhancing document-level RE. Coreference knowledge is injected by coreference distillation, while factual knowledge is injected and fused with document representations via representation reconciliation. Our experiments validate the generalization and the stable performance increase of KIRE to various RE models. For future work, we plan to exploit other knowledge injection frameworks and integrate more knowledge sources.

Supplemental Material Statement: Source code for KIRE is available from Github at https://github.com/nju-websoft/KIRE. Datasets are available from [32,33].

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K. (ed.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
2. Bastos, A., Nadgeri, A., Singh, K., Mulang, I.O., Shekarpour, S., Hoffart, J., Kaul, M.: RECON: Relation extraction using knowledge graph context in a graph neural network. In: WWW, pp. 1673–1685. ACM, Online (2021)
3. Busbridge, D., Sherburn, D., Cavallo, P., Hammerla, N.Y.: Relational graph attention networks. CoRR abs/1904.05811 (2019)
4. Fernàndez-Cañellas, D., et al.: Enhancing online knowledge graph population with semantic knowledge. In: Pan, J.Z. (ed.) ISWC 2020. LNCS, vol. 12506, pp. 183–200. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62419-4_11
5. Heist, N., Paulheim, H.: Language-agnostic relation extraction from wikipedia abstracts. In: d'Amato, C. (ed.) ISWC 2017. LNCS, vol. 10587, pp. 383–399. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_23
6. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR abs/1503.02531 (2015)
7. Huang, Q., Zhu, S., Feng, Y., Ye, Y., Lai, Y., Zhao, D.: Three sentences are all you need: Local path enhanced document relation extraction. In: ACL, pp. 998–1004. ACL, Online (2021)
8. Ji, G., Liu, K., He, S., Zhao, J.: Distant supervision for relation extraction with sentence-level attention and entity descriptions. In: AAAI, pp. 3060–3066. AAAI Press, San Francisco, CA, USA (2017)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR, OpenReview.net, Toulon, France (2017)

10. Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P.: K-BERT: Enabling language representation with knowledge graph. In: AAAI, pp. 2901–2908. AAAI Press, New York, NY, USA (2020)
11. Nan, G., Guo, Z., Sekulic, I., Lu, W.: Reasoning with latent structure refinement for document-level relation extraction. In: ACL, pp. 1546–1557. ACL, Online (2020)
12. Pan, J.Z., Zhang, M., Singh, K., Harmelen, F., Gu, J., Zhang, Z.: Entity enabled relation linking. In: Ghidini, C. (ed.) ISWC 2019. LNCS, vol. 11778, pp. 523–538. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30793-6_30
13. Peng, N., Poon, H., Quirk, C., Toutanova, K., Yih, W.t.:Cross-sentence N-ary relation extraction with graph LSTMs.Trans. Assoc. Comput. Linguist. **5**, 101–115 (2017)
14. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: EMNLP, pp. 1532–1543. ACL, Doha, Qatar (2014)
15. Quirk, C., Poon, H.: Distant supervision for relation extraction beyond the sentence boundary. In: EACL, pp. 1171–1182. ACL, Valencia, Spain (2017)
16. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
17. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A. (ed.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
18. Tang, H., et al.: HIN: hierarchical inference network for document-level relation extraction. In: Lauw, H.W., Wong, R.C.-W., Ntoulas, A., Lim, E.-P., Ng, S.-K., Pan, S.J. (eds.) PAKDD 2020. LNCS (LNAI), vol. 12084, pp. 197–209. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-47426-3_16
19. Tong, M., Xu, B., Wang, S., Cao, Y., Hou, L., Li, J., Xie, J.: Improving event detection via open-domain trigger knowledge. In: ACL, pp. 5887–5897. ACL, Online (2020)
20. Türker, R., Zhang, L., Alam, M., Sack, H.: Weakly supervised short text categorization using world knowledge. In: Pan, J.Z. (ed.) ISWC 2020. LNCS, vol. 12506, pp. 584–600. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62419-4_33
21. Vashishth, S., Joshi, R., Prayaga, S.S., Bhattacharyya, C., Talukdar, P.P.: RESIDE: improving distantly-supervised neural relation extraction using side information. In: EMNLP, pp. 1257–1266. ACL, Brussels, Belgium (2018)
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS, pp. 5998–6008. Curran Associates Inc, Long Beach, CA, USA (2017)
23. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: ICLR, OpenReview.net, Vancouver, BC, Canada (2018)
24. Verga, P., Strubell, E., McCallum, A.: Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In: NAACL, pp. 872–884. ACL, New Orleans, LA, USA (2018)
25. Verlinden, S., Zaporojets, K., Deleu, J., Demeester, T., Develder, C.: Injecting knowledge base information into end-to-end joint entity and relation extraction and coreference resolution. In: Findings of ACL, pp. 1952–1957. ACL, Online (2021)
26. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. CACM **57**(10), 78–85 (2014)
27. Wang, D., Hu, W., Cao, E., Sun, W.: Global-to-local neural networks for document-level relation extraction. In: EMNLP, pp. 3711–3721. ACL, Online (2020)
28. Wang, H., Focke, C., Sylvester, R., Mishra, N., Wang, W.: Fine-tune Bert for DocRED with two-step process. CoRR abs/1909.11898 (2019)

29. Wei, X., Wang, S., Zhang, D., Bhatia, P., Arnold, A.O.: Knowledge enhanced pre-trained language models: A comprehensive survey. CoRR abs/2110.08455 (2021)
30. Xu, B., Wang, Q., Lyu, Y., Zhu, Y., Mao, Z.: Entity structure within and throughout: Modeling mention dependencies for document-level relation extraction. In: AAAI, pp. 14149–14157. AAAI Press, Online (2021)
31. Xu, W., Chen, K., Zhao, T.: Document-level relation extraction with reconstruction. In: AAAI, pp. 14167–14175. AAAI Press, Online (2021)
32. Yao, Y., et al.: DocRED: A large-scale document-level relation extraction dataset. In: ACL, pp. 764–777. ACL, Florence, Italy (2019)
33. Zaporojets, K., Deleu, J., Develder, C., Demeester, T.: DWIE: an entity-centric dataset for multi-task document-level information extraction. IPM **58**(4), 102563 (2021)
34. Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J.: Relation classification via convolutional deep neural network. In: COLING, pp. 2335–2344. ACL, Dublin, Ireland (2014)
35. Zeng, S., Xu, R., Chang, B., Li, L.: Double graph based reasoning for document-level relation extraction. In: EMNLP, pp. 1630–1640. ACL, Online (2020)
36. Zhang, Y., Qi, P., Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. In: EMNLP, pp. 2205–2215. ACL, Brussels, Belgium (2018)
37. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: Enhanced language representation with informative entities. In: ACL, pp. 1441–1451. ACL, Florence, Italy (2019)
38. Zhou, W., Huang, K., Ma, T., Huang, J.: Document-level relation extraction with adaptive thresholding and localized context pooling. In: AAAI, pp. 14612–14620. AAAI Press, Online (2021)