KGPool: Dynamic Knowledge Graph Context Selection for Relation Extraction

Abhishek Nadgeri[1,2], **Anson Bastos**[1,3], **Kuldeep Singh**[1,5], **Isaiah Onando Mulang'**[1,7]
Johannes Hoffart[6], **Saeedeh Shekarpour**[4], and **Vijay Saraswat**[6]

[1]Zerotha Research, [2]RWTH Aachen, [3]IIT Hyderabad, [4]University of Dayton
[5]Cerence GmbH, [6]Goldman Sachs, [7]IBM Research, Africa

`abhishek.nadgeri@rwth-aachen.de, cs20resch11002@iith.ac.in`
`kuldeep.singh1@cerence.com, mulang.onando@ibm.com`
`sshekarpour1@udayton.edu,{johannes.hoffart,vijay.saraswat}@gs.com`

Abstract

We present a novel method for relation extraction (RE) from a single sentence, mapping the sentence and two given entities to a canonical fact in a knowledge graph (KG). Especially in this presumed sentential RE setting, the context of a single sentence is often sparse. This paper introduces the *KGPool* method to address this sparsity, dynamically expanding the context with additional facts from the KG. It learns the representation of these facts (entity alias, entity descriptions, etc.) using neural methods, supplementing the sentential context. Unlike existing methods that statically use all expanded facts, *KGPool* conditions this expansion on the sentence. We study the efficacy of *KGPool* by evaluating it with different neural models and KGs (Wikidata and NYT Freebase). Our experimental evaluation on standard datasets shows that by feeding the *KGPool* representation into a Graph Neural Network, the overall method is significantly more accurate than state-of-the-art methods.

1 Introduction

Knowledge graphs (KGs) are the foundation for many downstream applications and are growing ever larger. However, due to the sheer volume of knowledge and the world's dynamic nature where new entities emerge and unknown facts about them are learned, KGs need to be continuously updated. Distantly supervised Relation Extraction (RE) is an important KG completion task aiming at finding a semantic relationship between two entities annotated on the unstructured text with respect to an underlying knowledge graph (Ye and Ling, 2019). In the literature, researchers mainly studied two variants in the RE: 1) multi-instance RE and 2) sentential RE. The multi-instance RE assumes that in a given bag of sentences, if two entities participate in a relation, there exists at least one sentence with these two entities, which
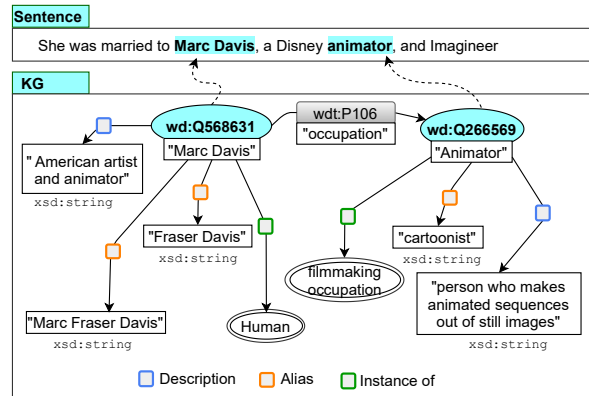

Figure 1: Illustration of Knowledge Graph Context associated with the annotated entities in a sentence. Here, entity aliases do not play any role in the understanding of the sentence for finding the KG relation. [a]

---

[a]sentence taken from (Sorokin and Gurevych, 2017)

may contain the target relation (Riedel et al., 2010; Vashishth et al., 2018). In this setting, researchers aim to incorporate contextual signals from the previous occurrences of an entity pair into the neural models to support relation extraction (Ye and Ling, 2019; Xu and Barbosa, 2019; Wu et al., 2019). In contrast, sentential RE restricts the scope of document context only to the input sentence (disregards other occurrences of entity pairs) while predicting the KG relation (Sorokin and Gurevych, 2017). Hence, sentential RE makes the multi-instance setting more difficult by limiting the available context.

Recent approaches for RE not only base KGs for relation inventory but also consider it for extending contextual knowledge for further improvement of RE task (Vashishth et al., 2018; Bastos et al., 2021). A few multi-instance RE methods rely on entity attributes (properties) such as descriptions, aliases, and types (as additional context) along with entity pair occurrences from previous sentences to improve the overall extraction quality (Ji et al., 2017; Vashishth et al., 2018). For

the sentential RE, the RECON approach (Bastos et al., 2021) aims to effectively encode KG context derived from the entity attributes and entity neighborhood triples. RECON employs a Graph Neural Network (GNN) as a context aggregator for combining sentential context (annotated entities and sentence) and structured KG representation. Although the additional KG context has a positive effect on the overall relation extraction in multi-instance and sentential RE settings, not all KG context forms are necessary for every input sentence. Consider Figure 1, where the task is to infer a semantic relation 'occupation' between two entities *wd:Q568631 (Marc Davis)*[1] and *wd:Q266569 (animator)*. Wikidata (Vrandecic, 2012) KG provides semantic information such as description, instance-of, and aliases about entities. Here, the entity alias (Marc Fraser Davis, Fraser Davis for *wd:Q568631*; and cartoonist for *wd:Q266569*) has no impact on understanding the sentence because the entities are explicitly mentioned in the sentence. Furthermore, there is empirical evidence in the literature that for several sentences, statically adding all KG context offered minimal or negative impact (Bastos et al., 2021). Hence, there are open research questions as to how an RE approach can dynamically utilize the sufficient context from KG and whether or not the selected KG context positively impacts the overall performance?

This paper studies these concerning questions proposing the *KGPool* approach. *KGPool* utilizes a self-attention mechanism in a Graph Convolution Network (GCN) (Kipf and Welling, 2017) for selecting a sub-graph from the KG to extend the sentential context. The concept of dynamically mapping the structural representation of a KG to a latent representation of a sentence has not been widely studied in prior literature. In RE, *KGPool* is the initial attempt. The existing approaches (Bastos et al., 2021; Xu and Barbosa, 2019; Wu et al., 2019; Vashishth et al., 2018) feed all the available context (either derived from a bag of sentences or a KG or both) into a neural model and relied on the model to figure out the consequences, resulting in limited performance in many cases (Bastos et al., 2021). Conversely, we study the efficacy of *KGPool* in dynamically choosing KG context for the sentential RE task using two standard community datasets (NYT Freebase (Riedel et al., 2010), and Wikidata (Sorokin

and Gurevych, 2017)). Our work makes the following key contributions:

- The *KGPool* approach dynamically selects structural knowledge and transform it into a representation suitable to supplement the latent representation of sentential context learned using a neural model. We deduce that *KGPool* is the first approach that works independently of the underlying context aggregators used in the literature (Graph Neural Network (Zhu et al., 2019) or LSTM-based (Sorokin and Gurevych, 2017)).

- We are the first to map the task of KG Context Selection to a Graph Pooling Problem. Therefore, our proposed approach legitimizes the application of graph pooling algorithms for choosing the relevant context.

- *KGPool*, paired with a GNN as context aggregator, significantly outperforms the existing baselines on both datasets, in one experiment increasing the precision by 12 points over to baseline (P@30 on NYT Freebase). Furthermore, our empirical results (cf., Table 3) conclude that an LSTM model paired with *KGPool* is able to notably outperform a GNN-based approach (Zhu et al., 2019) and nearly all multi-instance baselines (Ye and Ling, 2019; Wu et al., 2019; Vashishth et al., 2018) published in the recent years.

This paper is structured as follows: Section 2 reviews the related work. Section 3 formalizes the problem and the proposed approach is described in Section 4. Section 5 describes the experimental setup. The results are in Section 6. We conclude in Section 7.

2 Related Work

Multi-instance RE: a few multi-instance RE approaches use convolution neural network (dos Santos et al., 2015), attention CNN (Wang et al., 2016) and attention-based recurrent neural models for relation extraction (Zhou et al., 2016). Other approaches such as (Ji et al., 2017; Vashishth et al., 2018) incorporate entity descriptions, entity and relation aliases from KG to supplement context from the previous sentences. Work in (Vashishth et al., 2018) employs a graph convolution network to encode entity and relation aliases derived from Wikidata. HRERE (Xu and Barbosa, 2019) proposes an approach for jointly

---
[1]wd: binds to https://www.wikidata.org/wiki/

learning sentence and KG representation using cross-entropy loss function. To effectively capture the available entity context in the documents, Ye and Ling (2019) suggest an approach incorporating intra-bag and inter-bag attentions. For a detailed survey, we point readers to (Smirnova and Cudré-Mauroux, 2018).

Sentential RE: researchers (Sorokin and Gurevych, 2017) utilized additional relations present in the sentence to assist the process of extracting the target relation using an LSTM-based model. GP-GNN (Zhu et al., 2019) generates parameters of GNN based on the input sentence, which enables GNNs to process-relational reasoning on unstructured text inputs. RECON (Bastos et al., 2021) is an approach that uses the entity attributes (aliases, labels, descriptions, instance-of) and KG triples to signal an underlying GNN model for sentential RE. Authors conclude that the multi-instance requirement can be relaxed provided a good representation of KG context to enrich the sentential RE model. However, RECON and multi-instance approaches (Xu and Barbosa, 2019; Vashishth et al., 2018) utilize statically derived context from the KG, i.e., KG context does not vary depending the sentence.

Graph Pooling and Dynamic Context Selection: researchers proposed several models for the graph classification aka. graph pooling task (Cangea et al., 2018; Ying et al., 2018; Gao and Ji, 2019). These models employ various approaches such as graph topology-based (Rhee et al., 2018), and by learning the hierarchical graph-structure (Ying et al., 2018). Another graph pooling model relies on node features and topological information using self-attention (Lee et al., 2019) in which a specific number of nodes are always eliminated. In *KGPool*, the elimination of nodes depends on a context coefficient and node importance (Section 4). For context selection, a recent work focuses on dynamically selecting the KG context to optimize a Pre-Trained Language Model (PLM) for entity typing and relation classification (Su et al., 2020). *KGPool* has the following fundamental differences compared to (Su et al., 2020): *KGPool* inspires its self-attention mechanism from (Lee et al., 2019; Vaswani et al., 2017) to learn a representation of the KG context. Hence, *KGPool* works agnostic of the underlying model used for the context aggregation (unlike Su et al. (2020), which is tightly coupled with PLM). Approaches such as

(Liu et al., 2017; Zhang et al., 2018; Kang et al., 2020) also perform dynamic context selection for respective tasks. However, these approaches are not focused on knowledge graph context selection.

3 Problem Statement

We define the KG as a tuple given by $KG = (\mathcal{E}, \mathcal{R}, \mathcal{T}^+)$ where $\mathcal{E}$ denotes the set of all vertices in the graph representing entities, $\mathcal{R}$ is the set of edges representing relations, and $\mathcal{T}^+ \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of all KG triples. The *RE Task* predicts the target relation $r^c \in \mathcal{R}$ between a given pair of entities $\langle e_i, e_j \rangle$ from the sentence $\mathcal{W} = (w_1, w_2, ..., w_l)$. If no relation is inferred, it returns 'NA' label. We aim for the <u>sentential RE</u> task which put a constraint that the sentence within which a given pair of entities occurs is the only visible sentence from the bag of sentences. We view RE as a classification task similar to (Sorokin and Gurevych, 2017). In a KG triple $\tau = (e_h, r, e_t) \in \mathcal{T}^+$, the relation $r \in \mathcal{R}$, e_h is the head entity (relation origin) while e_t is the tail entity. For each entity, associated semantic properties such as entity label, description, instance-of, and aliases are known as entity attribute (At^e) (cf., graph construction step of Figure 2). We aim to model KG contextual information to improve the classification. This is achieved by learning the effective representations of the sets At^e, e_h, e_t, and $\mathcal{W}$ (cf. section 4).

4 KGPool Approach

KGPool consists of three components illustrated in Figure 2: 1) *Graph Construction* aggregates the sentence, entities and its attributes as a Heterogeneous Information Graph (HIG) for input representation. 2) *Context Pooling* step utilizes a self-attention mechanism in a graph convolution to calculate attention scores for entity attributes using node features and graph topology. The pooling process allows *KGPool* to construct a Context Graph (CG), which is a contextualized representation of HIG with lesser number of nodes. 3) *Context Aggregator* takes as input the sentence, entities, contextual representations of HIG, and classifies the target relation between the entities. We detail the approach in the following.

4.1 Graph Construction

As first step, we extract entity attributes (At^e) from public dumps of Freebase (Bollacker et al.,
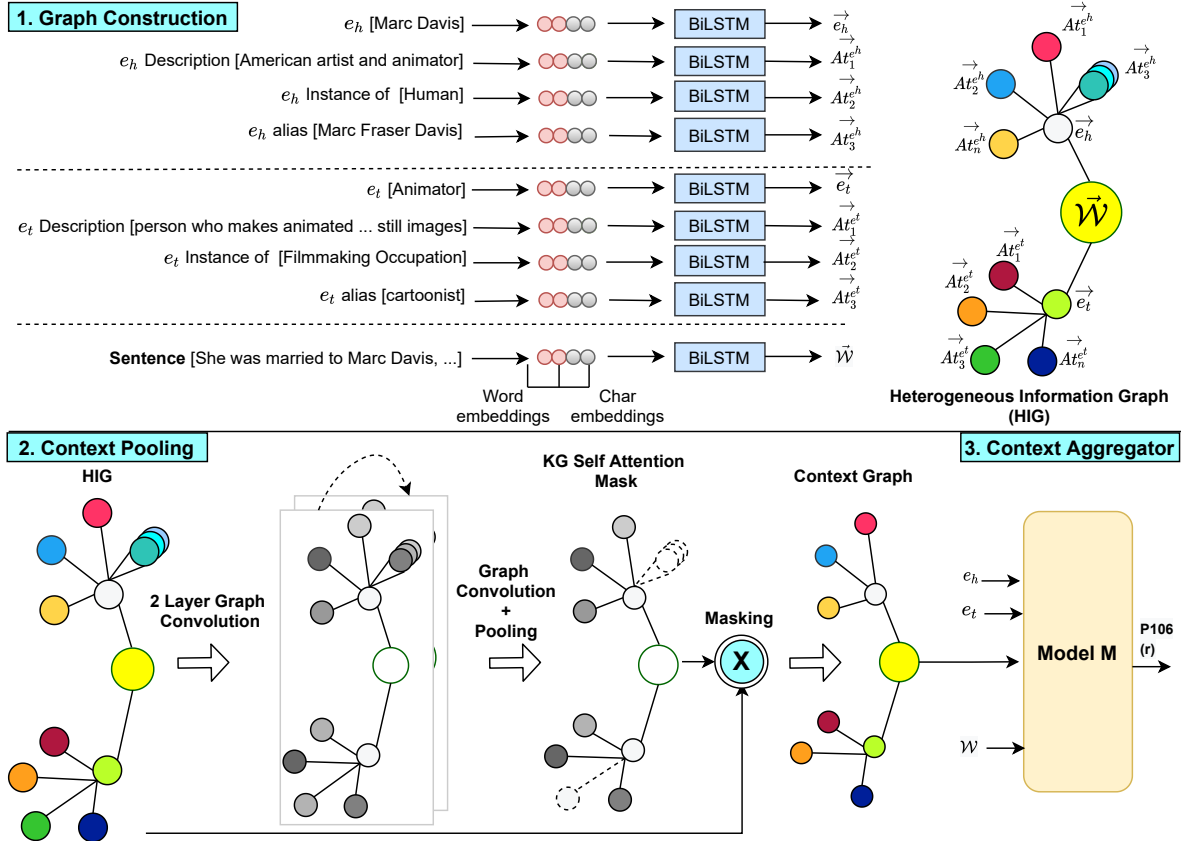
Figure 2: *KGPool* approach has three components to supplement sentential context with necessary KG context.

2007) and Wikidata (Vrandecic, 2012) KGs depending on the dataset. For the KG context, we rely on the most commonly available properties of the entities as suggested by (Bastos et al., 2021): aliases, description, instance-of, and label. An example of various entity attributes is given in Figure 2 at the Graph Construction step. Then, the sentence $\mathcal{W}$ is transformed to another representation using Bi-LSTM (Schuster and Paliwal, 1997) by concatenating its word and character embeddings:

$$\vec{\mathcal{W}} = BiLSTM(\mathcal{W}) \tag{1}$$

Similar representation is created for each entity e_i where $i = (h, t)$:

$$\vec{e_i} = BiLSTM(e_i) \tag{2}$$

For entity e_i, its KG contexts (entity attributes) $At_j^{e_i}$ (where $j = [0...N]$) are independently converted into associated embedding representations:

$$\vec{At_j}^{e^i} = BiLSTM(At_j^{e_i}) \tag{3}$$

For a knowledge representation of the KG context concerning the sentential context (sentence and annotated entities), we introduce the special graph $HIG = (A, F)$, a Heterogeneous Informa-

tion Graph, represented in the adjacency matrix $A \in \{0, 1\}^{n \times n}$, where n is the maximum number of neighboring nodes for an entity e_i. Here, $F \in \mathbb{R}^{n \times f}$ is the node feature matrix assuming each node has f features learned from the Bi-LSTM in the equations 1, 2, and 3. In these equations, BERT (Devlin et al., 2019), or any other recent Transformer-based model can be used. Due to hardware limitations, we are bound to Bi-LSTM using Glove embeddings (Pennington et al., 2014).

4.2 Context Pooling

Context pooling is built upon three layers of Graph Convolutional Networks (GCN) and a readout layer associated with each of them. Moreover, the last layer of GCN is coupled with a pooling layer (cf., ablation studies for architectural design choice experiments).

4.2.1 Graph Convolution

Since *KGPool* is expected to select the sufficient context, the Context Graph CG is a reduction of HIG using the mapping $\Psi : HIG \longrightarrow CG$. The challenge here is the no natural notion of spatial locality, i.e., it is not viable to pool together all context nodes in an "m × m" patch on HIG be-

cause the complex topological structure of graphs prevents any straightforward, deterministic definition of a "patch". Furthermore, entities nodes have a varying number of neighboring nodes, making the graph pooling challenging (similar to other graph classification problems (Ying et al., 2018)). In HIG, entity nodes do not contain information of their neighbors. Hence, we aim to enrich each entity node with the adjacent node's contextual information. Therefore, we employ a GNN variant to utilize its message-passing architecture to learn node embeddings from a message propagation function. The message propagation function depends on the adjacency matrix A, trainable parameters θ, and the node embeddings F (Ying et al., 2018). We rely on a GCN model by Kipf and Welling (2017). The GCN layer is defined as:

$$F^{(k)} = \text{ReLU}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}F^{(k-1)}\theta^{(k-1)}\right) \quad (4)$$

where $\tilde{A} = A + I$, $\tilde{D} = \sum_j \tilde{A}_{i,j}$ and $\theta^{(k)}$ is the trainable matrix. The GCN module might run k iteration and normally is in the range of two to six (Ying et al., 2018). A few graph representation learning approaches propose to use readout layer that aggregates node features to learn a fixed size representation (Xu et al., 2018; Cangea et al., 2018). We perform this summarizing after each block of the network (Equation 4), and aggregate all of the intermediate representation together by taking their sum. We define readout layer R as:

$$R^{(k)} = \frac{1}{N}\sum_{i=1}^{N} F_i^{(k)} \| \max_{i=1}^{N} F_i^{(k)} \quad (5)$$

where N is the number of nodes in the graph and F is the node feature embedding.

4.2.2 KG Self-Attention Mask

Until Equation 5, *KGPool* focuses on learning node features. Next, *KGPool* learns the importance of each entity attribute node using self-attention. Please note, in HIG, pooling happens only for entity attribute nodes ($\vec{At}_j^{e^i}$ from Equation 3). The sentence $\vec{\mathcal{W}}$ and entities $\vec{e_h}, \vec{e_t}$ remain intact. Hence, each entity representation $\vec{e_h}$ and $\vec{e_t}$ is enriched by the useful attribute context (KG context). The entity attribute nodes which do not provide relevant context are excluded from the graph. To choose the relevant entity attribute nodes, we use a self-attention score Z (Lee et al.,

2019) calculated as follows:

$$Z = \tanh\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}F^{(k)}\Theta_{att}\right) \quad (6)$$

where $\Theta_{att} \in \mathbb{R}^{F \times 1}$ is the only parameter of the pooling layer. For ranking, we take the attention score and pass it through a softmax layer where Z_{score} is the normalized self attention score.

$$Z_{score} = \exp\left(Z_i\right)/\sum_i \exp\left(Z_i\right) \quad (7)$$

After Equation 7, we compute the scores for each entity attribute node. Next, we propose a <u>node selection method</u> in which nodes are selected on the basis of **Context Coefficient** α which is a hyper parameter. The top nodes are selected as:

$$idx = \max\left(Z_{score}\right) - \alpha * \sigma(Z_{score}) \quad (8)$$

where $\sigma(Z_{score})$ is the standard deviation of Z_{score}, idx represents the node selection result, and Z_{mask} is the corresponding attention mask. Equation 8 acts as a soft constraint in selecting the context nodes for each HIG which depends on the value of α. Learning α during training may cause over-fitting. Hence, we decided to consider α as a trade-off parameter similar to λ in regularization (Bühlmann and Van De Geer, 2011). Next, the Context Graph (CG) is formed by pooling out the less essential entity attribute nodes as:

$$F' = F_{\text{idx},:}^{(k)}, F_{out} = F' \odot Z_{mask}, A_{out} = A_{\text{idx,idx}} \quad (9)$$

In addition to the dynamically selected nodes, we also inherit the intermediate node and graph representations of $k-1$ layers similar to ResNET (He et al., 2016). The intermediate representations ($k-1$) and the CG (k^{th} layer) is given as follows:

$$\begin{aligned} \vec{e_h}' &= F_{e_h}^{(1)} \oplus F_{e_h}^{(2)} \oplus ....F_{e_h}^{(k)} \\ \vec{e_t}' &= F_{e_t}^{(1)} \oplus F_{e_t}^{(2)} \oplus ....F_{e_t}^{(k)} \\ \vec{\mathcal{W}}' &= F_{\mathcal{W}}^{(1)} \oplus F_{\mathcal{W}}^{(2)} \oplus ....F_{\mathcal{W}}^{(k)} \\ \vec{R}' &= R^{(1)} \oplus R^{(2)} \oplus ....R^{(k)} \end{aligned} \quad (10)$$

where in the ith layer: $F_{e_l}^{(i)}$ is the node embedding of e_l, $l = (h,t)$, $F_{\mathcal{W}}^{(i)}$ is the node embedding of sentence $\mathcal{W}$, and $R^{(i)}$ is the readout. In the kth layer, $F^{(k)}$ is the F_{out} from Equation 9. The $\oplus$ is the concatenation among the vectors.

4.3 Context Aggregator

Finally, *KGPool* combines the latent representation (sentential context) with the structured representation learned in Equation 10. As such, we employ a model M which learns latent relation vec-

tor $\vec{r^j}$. In the state-of-the-art approaches that use KG context, the representation of $\vec{r^j}$ is learned using sentential and all static KG context (Vashishth et al., 2018; Bastos et al., 2021). However, in *KGPool*, relation r is realized based on the sentential context and dynamically chosen KG context. Hence, we employ context aggregators similar to the baselines (section 5.3) for jointly learning the enriched KG context in the form of CG and sentential context. The final relation is:

$$P(r \mid e_h, e_t, W) = softmax($$
$$MLP(\vec{r^j} \oplus \vec{e_h}' \oplus \vec{e_t}' \oplus \vec{\mathcal{W}}' \oplus \vec{R}')) \quad (11)$$

5 Experimental Setup

5.1 Datasets

We consider two standard datasets (English version): Wikidata dataset (Sorokin and Gurevych, 2017) and NYT Freebase (Riedel et al., 2010). Both datasets were annotated using distant supervision (associated stats are in Table 1). Datasets include 'NA' as one of the target relations.

Dataset	#Train Sentences	#Test sentences	#Relations
Wikidata	372,059	360,334	353
NYT	455,771	172,448	53

Table 1: Statistics of the Datasets

5.2 KGPool Configurations

KGPool is configured with two context aggregator modules. We inherit context aggregators from existing sentential RE baselines. Our experimental aim is to assess as how *KGPool* performs along with the state-of-the-art context aggregators (comparative study). Our two settings are:

1. $KGPool_{+lstm}$: *KGPool* is coupled with a context aware LSTM model from (Sorokin and Gurevych, 2017) as context aggregator.

2. $KGPool_{+gnn}$: this implementation has *KGPool* plugged-in with a variant of GNN module used by (Zhu et al., 2019; Bastos et al., 2021).

5.3 Baseline Models

We consider the recent sentential RE approaches for our empirical study:

RECON (Bastos et al., 2021): induces KG context (entity attributes and 1&2 hop entity triples) along with the sentence in a GNN.

RECON-EAC (Bastos et al., 2021): a variant of RECON contains entity attributes as only KG context (same context as *KGPool*).

GP-GNN (Zhu et al., 2019): performs multi-hop reasoning using a GNN.

**Context-LSTM** (Sorokin and Gurevych, 2017): uses context from other sentential relations.

Sorokin-LSTM (Sorokin and Gurevych, 2017): the NYT dataset contains one relation per sentence, but Context-LSTM requires at least two relations in a sentence. Thus, the other setting is an LSTM model without a sentential relation context, is used as a baseline on the NYT dataset.

Multi-instance RE Approaches: Please note, the Wikidata dataset does not have multiple instances for an entity pair. Hence multi-instance baselines do not have values on it. We inherit the recent multi-instance baselines and all empirical values from (Bastos et al., 2021): (i) **HRERE** (Xu and Barbosa, 2019) (ii) **Wu-2019** (Wu et al., 2019), (iii) **Yi-Ling-2019** (Ye and Ling, 2019), (iii) **RE-SIDE** (Vashishth et al., 2018).
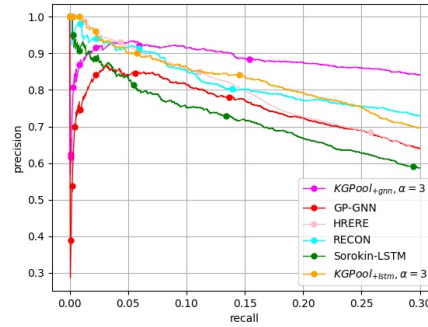
5.4 Metrics and Hyper-parameters

Graph Construction step (section 4.1) use a Bi-LSTM with one hidden layer of size 50 (Bastos et al., 2021). The word embedding dimension is 50 initialized by Glove embeddings (Pennington et al., 2014). The context pooling parameters are from (Lee et al., 2019). For model M, we used the default parameters provided by the authors (Zhu et al., 2019; Sorokin and Gurevych, 2017). For brevity, details are in the appendix.

Metric and Optimization: Our experiment settings are borrowed from (Bastos et al., 2021). Hence, on Wikidata dataset, we use (micro) precision (P), recall (R), and F-score (F1). On the NYT Freebase dataset, (micro) P@10 and P@30 is reported. P@K here represents precision at K percent recall. We also study the effect of Context Coefficient (α) for both *KGPool* configurations (trained end-to-end). We ignore the probability predicted for the NA relation during testing. We employ the Adam optimizer (Kingma and Ba, 2015) with categorical cross entropy loss where each model is run three times on the whole training set. For the P/R curves (with best α values of *KGPool* variants), the result from the first run of each model is selected. For ablation, we use the McNemar's test for statistical significance to find if the reduction in error in the *KGPool* configura-
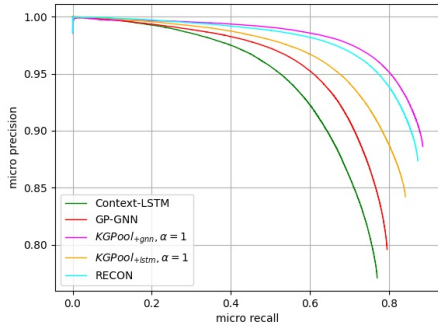
(a) Micro P-R Curve on Wikidata Dataset (b) Micro P-R Curve on NYT-Freebase Dataset

Figure 3: *KGPool*'s best configuration (Tables: 2, 3) perform better than baselines over the entire recall range.

tions are significant. The differences in the models is statistically significant if the $p-value < 0.05$ (Dieterich, 1998). We release all experiment code and data on a public GitHub[2].

6 Results

We conduct our experiments and analysis in response to the question **RQ**: "What is the efficacy of *KGPool* in dynamically selecting the KG context for the sentential RE task?" As such, we also compare *KGPool* against approaches that do not dynamically treat the context.

Model	P	R	F1
Context-LSTM	72.09	72.06	72.07
GP-GNN	82.30	82.28	82.29
RECON-EAC	85.44	85.41	85.42
RECON	87.24	87.23	87.23
$KGPool_{+lstm}$ (α=1)	84.20	82.19	84.20
$KGPool_{+lstm}$ (α=2)	84.12	84.13	84.12
$KGPool_{+lstm}$ (α=3)	84.00	83.97	83.98
$KGPool_{+lstm}$ (α=4)	83.81	83.79	83.80
$KGPool_{+gnn}$ (α=1)	**88.60**	**88.59**	**88.60**
$KGPool_{+gnn}$ (α=2)	88.57	88.56	88.57
$KGPool_{+gnn}$ (α=3)	88.54	88.55	88.54
$KGPool_{+gnn}$ (α=4)	88.52	88.50	88.51

Table 2: Comparison of *KGPool* configurations with sentential RE models on Wikidata dataset. Best score in bold.

Performance on Wikidata Dataset: Table 2 summarizes the performance of *KGPool* variants against the sentential RE models. Agnostic of the underlying aggregator (LSTM or GNN), *KGPool* effectively captures the KG context complimenting the sentential context. The $KGPool_{+gnn}$ (α=1) configuration outperforms other *KGPool* variants along with all sentential RE baselines. We

Model	P@10	P@30
HRERE	86.1	76.6
Wu-2019	81.7	61.8
RESIDE	73.6	59.5
Ye-Ling-2019	78.9	62.4
Sorokin-LSTM	75.4	58.7
GP-GNN	81.3	63.1
RECON-EAC	83.5	73.4
RECON	87.5	74.1
$KGPool_{+lstm}$ (α=1)	83.7	72.7
$KGPool_{+lstm}$ (α=2)	83.5	71.6
$KGPool_{+lstm}$ (α=3)	84.1	70.6
$KGPool_{+lstm}$ (α=4)	83.1	72.1
$KGPool_{+gnn}$ (α=1)	90.1	**86.7**
$KGPool_{+gnn}$ (α=2)	91.0	85.0
$KGPool_{+gnn}$ (α=3)	**92.3**	85.4
$KGPool_{+gnn}$ (α=4)	90.6	84.4

Table 3: Comparison of *KGPool* with sentential and multi-instance RE models on NYT Freebase dataset. Best score in bold.

can observe that even when the available context is limited to entity attributes, the $KGPool_{+gnn}$ variant surpasses RECON that also contains context from 1&2 hop triples besides the entity attributes. RECON-EAC and $KGPool_{+gnn}$ rely on entity attributes as KG context with the same context aggregator. When $KGPool_{+gnn}$ variants choose KG context dynamically, they perform better than RECON-EAC. It is interesting to notice that when an LSTM model is fed with the dynamically chosen context, the performance gain is more than ten absolute points ($KGPool_{+lstm}$ Vs Context-LSTM), even outperforming GP-GNN.

Performance on NYT Freebase Dataset: Similar to the Wikidata dataset, the $KGPool_{+gnn}$ variants significantly outperform all baselines (cf. Table 3). The P@30 is comparatively high for $KGPool_{+gnn}$ against baselines. The behavior could be interpreted as follows: dynamically adding context from the KG for the entity pairs

Compared Models	Contingency table		Statistic	p-value	Significance	Dataset
$KGPool_{+gnn}$ (α=3) vs RECON	160916 3169	4702 3613	298.18	$8.18 * 10^{-67}$	Statistically Significant	NYT-Freebase
$KGPool_{+gnn}$ (α=1) vs RECON	617266 29255	38652 55593	1300.08	$1.08 * 10^{-284}$	Statistically Significant	Wikidata

Table 4: The McNemar's test for statistical significance for KGPool's best configuration Vs previous baseline.

keeps the precision higher over a more extended recall range. For both datasets, *KGPool* configurations ($KGPool_{+gnn}$ and $KGPool_{+lstm}$) have the best-reported performance varying as per the α. This validates our choice to introduce a soft constraint in selecting the context nodes (cf., Equation 8). The P/R curves in Figure 3 show that *KGPool* performs better than baselines over the entire recall range. We conclude that the effective dynamic context selection by *KGPool* has a positive impact on the sentential RE task (which successfully answers our research question).

Models	DEG (HIG)	DEG (CG)	Dataset
$KGPool_{+gnn}$ (α=1)	5.33	1.15	Wikidata
$KGPool_{+gnn}$ (α=2)	5.33	1.52	
$KGPool_{+gnn}$ (α=3)	5.33	2.87	
$KGPool_{+gnn}$ (α=4)	5.33	4.71	
$KGPool_{+gnn}$ (α=1)	6.34	1.67	NYT Freebase
$KGPool_{+gnn}$ (α=2)	6.34	1.91	
$KGPool_{+gnn}$ (α=3)	6.34	2.73	
$KGPool_{+gnn}$ (α=4)	6.34	5.16	

Table 5: Effect of Context Pooling. 'DEG' denotes average degree of an entity node (e_i). 'DEG' of entity nodes in CG is drastically reduced wrt the HIG.

6.1 Ablation Studies

We conducted two ablation studies to understand the behavior of *KGPool* configurations:

Significance of Dynamic Context Selection: we perform McNemar's test for the best *KGPool* configuration against the previous sentential state-of-the-art (i.e. RECON). The results in Table 4 are statistically significant on both datasets, illustrating *KGPool*'s robustness. Although $KGPool_{+gnn}$ variants achieve statistically significant results against RECON, there exist several sentences for which our approach is unable to select supplementary KG context ((RW) values in the contingency table). It requires further investigation, and we plan it for our future work.

Effect on the Degree of Nodes for Entities: for studying the effect of context pooling (Section 4.2), we also conducted a study to understand the impact of *KGPool* on the reduction of the average degree of entity nodes (e_i) in the HIG. Table 5

summarizes the effect of Context Coefficient on the average degree of entity nodes. Irrespective of α, *KGPool* notably reduces the degree of e_i by removing less relevant nodes.

Architectural Choice Experiment: In *KGPool*, we chose to introduce pooling in the last layer of a three-layered architecture (three blocks). To support our choice, we performed several additional experiments by introducing pooling in various layers. We employ the Wikidata dataset for our experiments. We use best configuration of our model ($KGPool_{+gnn}$ (α=1)) and created several variants of it. For instance, $KGPool_{+gnn}$ (P=all) comprises the configuration where we introduce pooling in all three GCN blocks. The configuration $KGPool_{+gnn}$ (¶=2&3) has no pooling in the first layer but has a pooling layer in the remaining two GCN blocks. $KGPool_{+gnn}$ is the best configuration of *KGPool* where pooling is just in the final layer. In Table 6, we observe that $KGPool_{+gnn}$ with pooling only in the last GCN block has the superior performance compared to other two variants. Here, the first two layers are used to learn the node features, which are then employed with self-attention for node selection. Our experiments justify the architectural choice decision. However, with a newer graph pooling technique, such decisions will solely depend on the performance of the approach, and we can not generalize the results of these experiments.

Model	F1
$KGPool_{+gnn}$ (¶=all)	84.19
$KGPool_{+gnn}$ (¶=2&3)	86.87
$KGPool_{+gnn}$ (¶=3) (best)	**88.60**

Table 6: When we introduce pooling in all three layers or in two layers, the performance of *KGPool*'s variants drop. Hence, it justify our choice to add pooling only in the third layer that gives the best performance (values in bold). We use best configuration of our model ($KGPool_{+gnn}$ (α=1)).

Case-Studies: To understand the *KGPool*'s performance gain, we report a few top relations in Table 7. It can be observed from this table that in a few cases, with lesser context, *KGPool* can perform significantly better. In the next case study,

to understand the *KGPool*'s performance while adding additional context (more noise), we induce extra context in the form of 1&2-hop triples along with entity attributes. For the same, we considered *KGPool*'s best configurations on the Wikidata dataset. The configurations $KGPool_{+gnn}$ $(+T)$ and $KGPool_{+lstm}$ $(+T)$ represent *KGPool* fed with additional triple context. For both configurations agnostic of underlying aggregator, we observe a slight increase in performance (Table 8). There are several triples which are the irrelevant source of information not needed for a given sentence. *KGPool* can remove that information and does not suffer the performance drop due to added noise in the context. Details on error analysis, performance for worst performing individual relations, and on a human-annotated dataset are provided in the appendix.

Relation	KGPool	RECON	GP-GNN
vocal specialization	**1.00**	0.00	0.00
list of works	**1.00**	0.00	0.00
track gauge	**1.00**	0.92	0.00
position played	**0.99**	**0.99**	0.92
sport	**0.99**	**0.99**	0.97
record label	**0.95**	0.90	0.64
list of episodes	**0.95**	0.00	0.49
wing configuration	**0.94**	0.57	0.00
numeric value	**0.93**	0.27	0.46
vessel class	**0.87**	0.00	0.00

Table 7: Micro F-score of Top performing Relations for $KGPool_{gnn}$ (α=1) (on Wikidata dataset). Dynamically chosen context significantly improves performance for many relations.

Model	F1
$KGPool_{+gnn}$ $(+T)$	**88.85**
$KGPool_{+lstm}$ $(+T)$	84.42
$KGPool_{+gnn}$	88.60
$KGPool_{+lstm}$	84.12

Table 8: To scale the sources of the contexts, we induce additional triple context in the *KGPool* shown as $(+T)$ configurations. We use best configurations of our model ($KGPool_{+gnn}$ (α=1) and $KGPool_{+lstm}$ (α=1)). We observe a slight jump in the performance, however, *KGPool* is still able to pool irrelevant context.

7 Discussion and Conclusion

Although KGs are often employed for providing background context in the RE tasks (cf. Section 2), yet there is limited research about defining relevant context. In this work, we proposed *KG-Pool* and provide a set of experiments proving: 1) Given the limited context that is in individual sentences, dynamically bringing context from KG significantly improves the RE performance. 2) We introduced Context Coefficient (α), which acts as a soft constraint in determining the relevant entity context nodes. 3) Our approach *KGPool* is invariant of the context aggregator and enables us to learn effective knowledge representation of the required KG context for a given sentential context. Our evaluation concerns several key questions:

- **Data quality impact on an effective knowledge representation:** in spite *KGPool*'s significant performance, there exist several sentences for which our model finds a limitation compared to the baseline (cf. Table 4). One potential interpretation could be about the noise injected due to the data quality of the KG context (Weichselbraun et al., 2018). Hence, how does the quality of contextual data impact the performance of context selection approaches is an open direction.

- **Impact of additional sources of KG context:** In ablation, we provide a study by adding 1 & 2-hop triples in addition to entity attributes. There is no significant increase in the performance, although *KGPool* is able to remove irrelevant context for a given sentence. Furthermore, we did not consider edge features in HIG although *KGPool* can be extended to support edge features using techniques such as (Simonovsky and Komodakis, 2017). Additional experiments are needed to verify that our empirical observations hold in this setting, and we leave it for future work.

Overall, *KGPool* provides an effective knowledge representation for set-ups where sentence context is sparse. It is interesting to observe that effective knowledge representation learned using *KGPool* paired with an LSTM model outperforms GP-GNN (Zhu et al., 2019), and nearly all multi-instance baselines. Our conclusive results open a new research direction: *is it possible to apply effective context selection techniques coupled with deep learning models to other downstream NLP tasks?* For example, our results can encourage researchers to extend *KGPool* or develop novel context selection methods for the tasks where KGs have been extensively used as additional background knowledge, such as in entity linking (Mulang' et al., 2020; Mulang et al., 2020), KG completion (Wang et al., 2020; Shi et al., 2017), and recommendation system (Yang et al., 2020).

8   Ethics/ Impact Statement:

In this work, we present significant progress in solving sentential RE task. Harvesting knowledge is an essential goal that human beings seek along with the advancement of technology. This research and many RE approaches rely on additional signals from the public KGs to design systems that extract structured knowledge from unstructured contents. When it comes to who may be disadvantaged from this research, we do not think it is applicable since our study of addressing the KG context capabilities is still at an early stage. Having said so, we are fully supporting the development of ethical and responsible AI. The potential bias in the standard public datasets that may lead to wrong knowledge needs to be cleaned or corrected with validation mechanisms.

References

Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Isaiah Onando Mulang, Saeedeh Shekarpour, Johannes Hoffart, and Manohar Kaul. 2021. Recon: Relation extraction using knowledge graph context in a graph neural network. In *Proceedings of The Web Conference (WWW) (long papers)*, page :N/A.

Kurt D. Bollacker, Robert P. Cook, and Patrick Tufts. 2007. Freebase: A Shared Database of Structured General Human Knowledge. In *AAAI*.

Peter Bühlmann and Sara Van De Geer. 2011. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Catalina Cangea, Petar Velickovic, Nikola Jovanovic, Thomas Kipf, and Pietro Liò. 2018. Towards sparse hierarchical graph classifiers. *CoRR*, abs/1811.01287.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Thomas G Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Hongyang Gao and Shuiwang Ji. 2019. Graph u-nets. In *International Conference on Machine Learning*, pages 2083–2092.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3060–3066. AAAI Press.

Xiaomian Kang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2020. Dynamic context selection for document-level neural machine translation via reinforcement learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2242–2254. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *International Conference on Machine Learning*, pages 3734–3743.

Liping Liu, Francisco Ruiz, Susan Athey, and David Blei. 2017. Context selection for embedding models. *Advances in Neural Information Processing Systems*, 30:4816–4825.

Isaiah Onando Mulang', Kuldeep Singh, Chaitali Prabhu, Abhishek Nadgeri, Johannes Hoffart, and Jens Lehmann. 2020. Evaluating the impact of knowledge graph context on entity disambiguation models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2157–2160.

Isaiah Onando Mulang, Kuldeep Singh, Akhilesh Vyas, Saeedeh Shekarpour, Maria-Esther Vidal, and Soren Auer. 2020. Encoding knowledge graph entity aliases in attentive neural network for wikidata entity linking. In *International Conference on Web Information Systems Engineering*, pages 328–342. Springer.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Sungmin Rhee, Seokjun Seo, and Sun Kim. 2018. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3527–3534.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD*, volume 6323 of *Lecture Notes in Computer Science*, pages 148–163. Springer.

Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Jun Shi, Huan Gao, Guilin Qi, and Zhangquan Zhou. 2017. Knowledge graph embedding with triple context. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2299–2302.

Martin Simonovsky and Nikos Komodakis. 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 29–38. IEEE Computer Society.

Alisa Smirnova and Philippe Cudré-Mauroux. 2018. Relation extraction using distant supervision: A survey. *ACM Computing Surveys (CSUR)*, 51(5):1–35.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 1784–1789.

YuSheng Su, Xu Han, Zhengyan Zhang, Peng Li, Zhiyuan Liu, Yankai Lin, Jie Zhou, and Maosong Sun. 2020. Contextual knowledge selection and embedding towards enhanced pre-trained language models. *arXiv preprint arXiv:2009.13964*.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha P. Talukdar. 2018. RESIDE: improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Denny Vrandecic. 2012. Wikidata: a new platform for collaborative data collection. In *Proceedings of the 21st World Wide Web Conference, WWW 2012 (Companion Volume)*, pages 1063–1064.

Hongwei Wang, Hongyu Ren, and Jure Leskovec. 2020. Entity context and relational paths for knowledge graph completion. *arXiv preprint arXiv:2002.06757*.

Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307.

Albert Weichselbraun, Philipp Kuntschik, and Adrian MP Braşoveanu. 2018. Mining and leveraging background knowledge for improving named entity linking. In *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, pages 1–11.

Shanchan Wu, Kai Fan, and Qiong Zhang. 2019. Improving distantly supervised relation extraction with neural noise converter and conditional optimal selector. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 7273–7280. AAAI Press.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *ICML*.

Peng Xu and Denilson Barbosa. 2019. Connecting language and knowledge with heterogeneous representations for neural relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1*, pages 3201–3206.

Susen Yang, Yong Liu, Yonghui Xu, Chunyan Miao, Min Wu, and Juyong Zhang. 2020. Contextualized graph attention network for recommendation with item knowledge graph. *arXiv preprint arXiv:2004.11529*.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Distant supervision relation extraction with intra-bag and inter-bag attentions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 2810–2819.

Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2205–2215. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.

Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun. 2019. Graph neural networks with generated parameters for relation extraction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pages 1331–1339.

A Appendix

Due to page limit, we could not put several empirical results in the main paper. This section describes the remaining empirical studies.

A.1 Error Analysis

To understand the failure cases of *KGPool*, we conducted exhaustive error analysis. We calculated (micro) F1-Score of each relation in Sorokin dataset (Sorokin and Gurevych, 2017). Table 10 illustrates performance of ten relations on which *KGPool* performs the worst (ascending order of Micro-F1 score). To put the study in the right perspective, we also report all sentential RE baselines' performance on these relations. While analyzing the errors, we observe three patterns. First, all models fail in the relations for which number of instances are sparse. For example, the relation *mother* has only 190 instances (occurances) and the relation *killed by* has 48 instances. The scarcity in training data has made all models to fail on certain relations. Secondly, our model fails in very closed relations. For example, instead of predicting the relation *drafted by*[3], our model predicts *member of sport team*[4]. Similarly, in case of *unmarried partner*, our model predicts *spouse*. We believe that introducing logical reasoning in the model can help these borderline cases. The third observed pattern for errors is the quality of

---

[3]https://www.wikidata.org/wiki/Property:P647
[4]https://www.wikidata.org/wiki/Property:P54

context. It is worthwhile to mention that in GP-GNN and Context-LSTM, there is only a sentential context. RECON and *KGPool* use KG context. Still, performance is limited for many relations such as *use* and *different from* as reported in the table 10. The lack of quality context in the KG possibly a reason for limited performance for KG-context-induced models in erroneous cases. Detailed exploration is needed to understand the impact of data quality on *KGPool* performance, and we leave it for the future work.

A.2 Effect of Context Pooling

Models	DEG (HIG)	DEG (CG)	Dataset
$KGPool_{+lstm}$ (α=1)	5.33	1.06	Wikidata
$KGPool_{+lstm}$ (α=2)	5.33	2.12	
$KGPool_{+lstm}$ (α=3)	5.33	4.32	
$KGPool_{+lstm}$ (α=4)	5.33	4.81	
$KGPool_{+lstm}$ (α=1)	6.34	1.23	NYT
$KGPool_{+lstm}$ (α=2)	6.34	1.74	Freebase
$KGPool_{+lstm}$ (α=3)	6.34	3.05	
$KGPool_{+lstm}$ (α=4)	6.34	6.30	

Table 9: Effect of Context Pooling. 'DEG' denotes average degree of an entity node (e_i). We observe a reduction in the degree of entity nodes in CG compared to the HIG.

In the main paper, we presented the effect of context pooling on *KGPool*'s best configuration ($KGPool_{+gnn}$). Table 9 describes the reduction in the average degree of nodes for $KGPool_{+lstm}$ configuration for various context coefficient (α). On both datasets, there is a significant reduction in the degree of nodes. On Wikidata dataset (Sorokin and Gurevych, 2017), $KGPool_{+lstm}$ with (α=1) reports the highest value among its other configurations. For the same, the average degree of nodes is reduced from 5.33 to 1.06. Please note, the degree of nodes in HIG remains the same. However, for CG, the degree of nodes differs based on the context aggregator. We train the model end to end, and due to back-propagation, context weights adjust as per the context aggregator.

A.3 Results on a Human Annotated Dataset

The employed datasets Wikidata (Sorokin and Gurevych, 2017) and NYT Freebase (Riedel et al., 2010) are created using distant supervision techniques. Considering distant supervision techniques inherit a noise, to provide a comprehensive ablation study, (Zhu et al., 2019) provided a human evaluation setting. Following the same setting, RECON provided human-annotated data

Relation	*KGPool*	RECON	GP-GNN	Context-LSTM
has quality	0.00	0.00	0.00	0.00
enclave within	0.00	0.00	0.00	0.00
drafted by	0.01	0.08	0.00	0.02
different from	0.01	0.00	0.0	0.00
mother	0.03	0.05	0.02	0.00
unmarried partner	0.04	0.01	0.00	0.00
killed by	0.04	0.01	0.00	0.04
use	0.09	0.00	0.00	0.00
lyrics by	0.10	0.13	0.00	0.00
relative	0.12	0.10	0.00	0.00

Table 10: Micro F-score of 10-worst performing Relations for $KGPool_{gnn}$ (α=1) on Wikidata dataset. We also provide corresponding values of other sentential RE baselines. The main reason for limited performance across all models is the scarcity of training data for these relation types.

from Wikidata dataset (Sorokin and Gurevych, 2017). This is to verify that the distantly supervised dataset is correct for every pair of entities. Sentences accepted by all annotators are part of the human-annotated dataset. There are 500 sentences and 1846 triples in the test set. Table 11 reports $KGPool$'s performance against the sentential baselines. $KGPool_{+gnn}$ continues to outperform the baselines, maintaining similar behavior as seen on test sets of original datasets. The results further re-assure the robustness of our proposed approach.

Model	P	R	F1
Context Aware LSTM	77.77	78.69	78.23
GP-GNN	81.99	82.31	82.15
RECON-EAC	86.10	86.58	86.33
RECON	87.34	87.55	87.44
$KGPool_{+lstm}$ (α=1)	86.34	86.07	86.20
$KGPool_{+gnn}$ (α=1)	**89.36**	**89.31**	**89.33**

Table 11: Sentential RE performance on Human Annotation Dataset. *KGPool* again outperforms the baselines. We report Micro P,R, and F1 values. (Best score in bold)

Hyperparameters	Value
learning rate	0.001
batch size	50
hidden state size	128
context coefficient (α)	1,2,3,4
# of propagation layers	3

Table 12: Hyper-parameters for Context Pooling module

A.4 Datasets and Hyper-parameters

We augmented two datasets Wikidata dataset and Riedel Freebase dataset with our proposed KG context. The Wikidata dataset has 353 unique relations, 372,059 sentences in training, 123824 sentences in validation and 360,334 for testing. The number of sentences in the training and test

Hyperparameters	Value
learning rate	0.001
batch size	50
dropout ratio	0.5
hidden state size	256
non-linear activation	relu
# of propagation layers	3
entity embedding size	8
adjacent matrices	untied
optimizer	adam
β_1	0.9
β_2	0.999
ε	1e-08
pretrained embeddings	glove
word embedding dim	50

Table 13: Hyper-parameters for GNN-Aggregator module

Hyperparameters	Value
learning rate	0.001
batch size	50
dropout ratio	0.5
hidden state size	256
non-linear activation	relu
# of layers	1
optimizer	adam
pretrained embeddings	glove
word embedding dim	50

Table 14: Hyper-parameters for ContextAware-Aggregator module

Hyperparameters	Value
learning rate	0.001
batch size	50
initial embedding size	50
final embedding size	50
pretrained embeddings	glove
# of layers	1

Table 15: Hyper-parameters for Graph Construction module

set are 455,771 and 172,448 respectively in the Riedel dataset. No explicit validation set has been provided for Riedel dataset. For augmenting entity attribute context, we relied on public dumps of Wikidata and Freebase. From these dumps, we automatically extracted entities and its proper-

ties: labels, aliases, instance of, descriptions. For Wikidata, we used public API[5] using a SPARQL query and for Freebase, we took original depreciated dump[6].

We use the nltk english tokenizer for splitting the sentence into its corresponding tokens in the Riedel dataset. We do not do any further data preprocessing. We used 1 GPU NVIDIA TITAN X Pascal with 12GB of GPU storage to run our experiments. We train the models upto a maximum of 14 epochs and select the best performing model based on the micro F1 scores of the validation set. The tables 14, 15 and 12 detail the hyperparameter settings used in our experiments. We do not do any further hyper-parameter tuning.

---

[5]https://query.wikidata.org/
[6]https://developers.google.com/freebase