Two Training Strategies for Improving Relation Extraction over Universal Graph

Qin Dai[1], Naoya Inoue[2], Ryo Takahashi[1,3], Kentaro Inui[1,3]
[1]Tohoku University, Japan
[2]Stony Brook University, USA
[3]RIKEN Center for Advanced Intelligence Project, Japan
{daiqin, naoya-i, ryo.t, inui}@ecei.tohoku.ac.jp

Abstract

This paper explores how the Distantly Supervised Relation Extraction (DS-RE) can benefit from the use of a Universal Graph (UG), the combination of a Knowledge Graph (KG) and a large-scale text collection. A straightforward extension of a current state-of-the-art neural model for DS-RE with a UG may lead to degradation in performance. We first report that this degradation is associated with the difficulty in learning a UG and then propose two training strategies: (1) Path Type Adaptive Pretraining, which sequentially trains the model with different types of UG paths so as to prevent the reliance on a single type of UG path; and (2) Complexity Ranking Guided Attention mechanism, which restricts the attention span according to the complexity of a UG path so as to force the model to extract features not only from simple UG paths but also from complex ones. Experimental results on both biomedical and NYT10 datasets prove the robustness of our methods and achieve a new state-of-the-art result on the NYT10 dataset. The code and datasets used in this paper are available at https://github.com/baodaiqin/UGDSRE.

1 Introduction

Relation Extraction (RE) is an important task in Natural Language Processing (NLP). RE aims to turn unstructured texts into structured Knowledge Graph (KG), which is typically stored as (e_1, r, e_2) triplets, where e_1 is a *head entity*, r is a relation and e_2 is a *tail entity*, such as (*aspirin*, *may_treat*, *pain*) and (*Guy Maddin*, *place_lived*, *Winnpeg*). RE can be formulated as a classification task to predict a predefined relation r from entity pair (e_1, e_2) annotated evidences.

One obstacle that is encountered when building a RE system is the generation of a large amount of manually annotated training instances, which is expensive and time-consuming. For coping with this difficulty, Mintz et al. (2009) propose Distant Supervision (DS) to automatically generate training samples via linking KGs to texts. They assume that if (e_1, r, e_2) is in a KG, then all sentences that contain (e_1, e_2) (hereafter, *sentence evidences*) express the relation r. It is well known that the DS assumption is too strong and inevitably accompanies the wrong labeling problem, such as the sentence evidences (1b and 2) below, which fail to express *may_treat* and *place_lived* relation respectively.

(1) a. **Aspirin**$_{e_1}$ *is widely used for short-term treatment of* **pain**$_{e_2}$*, fever or colds.*

   b. *The tumor was remarkably large in size, and* **pain**$_{e_2}$ *unrelieved by* **aspirin**$_{e_1}$*.*

(2) *He is now finishing a documentary about* **Winnipeg**$_{e_2}$*, the final installment of a personal trilogy that began with "Cowards Bend the Knee" (a 2003 film that also featured a hapless hero named* **Guy Maddin**$_{e_1}$ *).*

Recently, neural network models with attention mechanism have been proposed to alleviate the wrong labeling problem and attend informative sentence evidences such as (1a) (Lin et al., 2016; Ji et al., 2017; Du et al., 2018; Jat et al., 2018; Han et al., 2018a,b). However, there can be a large portion of entity pairs that lack such informative sentence evidences that explicitly express their relation. This makes Distantly Supervised Relation Extraction (DS-RE) further challenging (Sun et al., 2019).

For compensating the lack of informative sentence evidences, Dai et al. (2019) utilize multi-hop paths connecting a target entity pair (hereafter, *path*) over a KG as extra evidences for DS-RE. An example of such multi-hop KG path can be seen in Figure 1, where p_1 depicts a multi-hop KG path
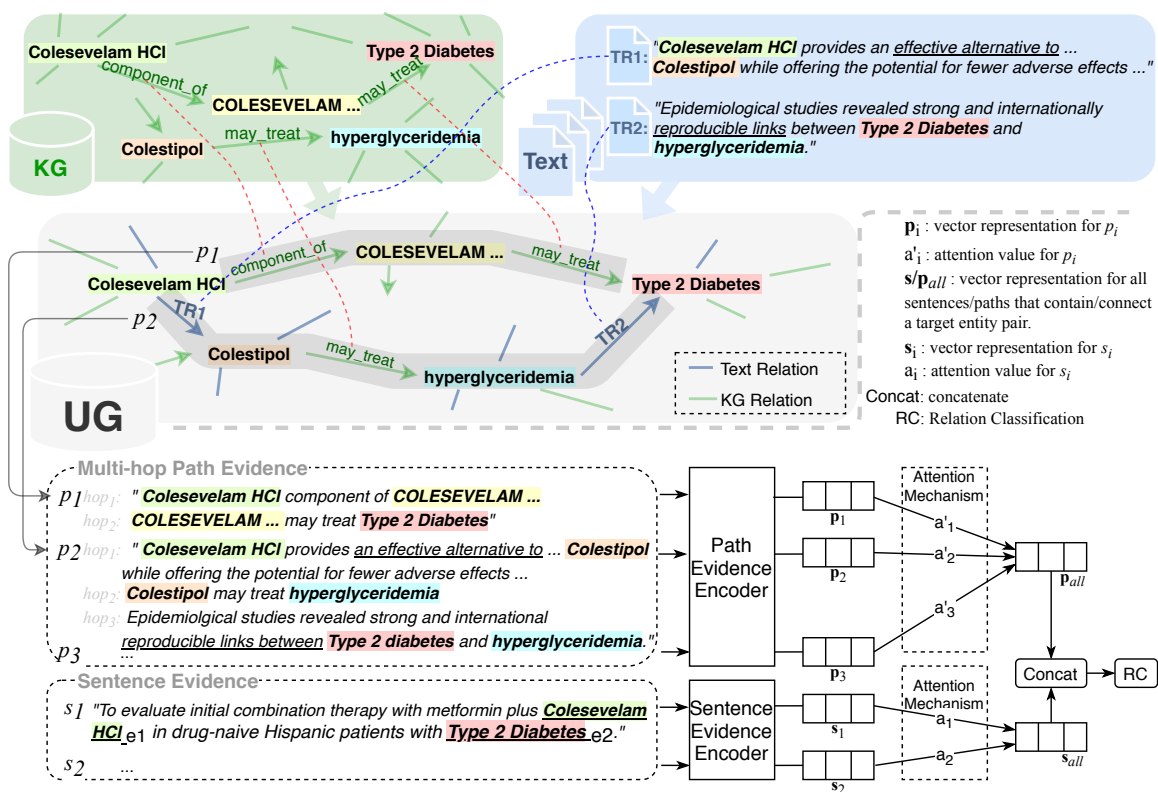
Figure 1: Overview of our UG-based framework, where **Colesevelam HCl** and **Type 2 Diabetes** are the target entities, **COLESEVELAM ...**, **Colestipol** and **hyperglyceridemia** are intermediate entities, each UG path consists of multiple hops and each hop represents a KG relation (such as "*Colestipol may treat hyperglyceridemia*") or Text (or Textual) relation (such as TR1 and TR2), which is the sentence containing two (target or intermediate) entities.

of the form of $e_1 \xrightarrow{component\_of} e_3 \xrightarrow{may\_treat} e_2$. The model of Dai et al. (2019) uses such multi-hop paths as additional features for predicting the relation between a given target entity pair (e_1, e_2), which is reported effective for performance improvement. However, KGs are often highly incomplete (Min et al., 2013) and may be too sparse to provide enough informative paths in practice, which may hamper the effectiveness of multi-hop paths.

Given this background, in this study, we take one step further, aiming for inducing maximal signals of distant supervision from both a KG and a large text collection (hereafter, *Text*). For this purpose, we consider using multi-hop paths over a Universal Graph (UG) as extra features for DS-RE. Here, we define a UG as a joint graph representation of both KG and Text, where each node represents an entity from KG or Text, and each edge indicates a KG relation or Textual relation, as shown in Figure 1. The path p_2 in the figure is an example of UG path, comprising a textual edge TR1, a KG edge *may_treat*, and another textual edge TR2. By augmenting the original KG with textual edges,

one can expect far more chances to find informative path evidences between any given target entity pairs, because the number of such textual edges is likely to be much larger than the number of KG edges (Note that one can collect as many textual edges as needed from a raw text corpus with an entity linker). Extending a KG to a UG, therefore, may allow a DS-RE model to learn richer distant supervision signals.

The idea of using multi-hop paths over a UG is not necessarily new on its own. For example, Toutanova et al. (2015) propose to use a UG for knowledge graph completion, and Das et al. (2017b) propose a model trained to reason over a UG for question answering. However, there is no prior study that has explored the effective way to use a UG for the task of DS-RE from text. In fact, finding an effective way of using a UG for DS-RE is not as simple as it may seem. As we report in this paper, a straightforward extension of the Dai et al. (2019) model to the UG setting may result in performance degradation.

Motivated by this, in this paper, we address how one can make effective use of UG for DS-RE. We

first report our observation that a straightforward extension of the Dai et al. (2019) model to the UG setting tends to allocate the majority of attention to only a limited set of UG paths such as short KG paths and miss out the learning from a wide range of UG paths (§4.1), which hinders performance gain. In order to alleviate the negative effect of the attention bias and realize the potential of UG paths, we propose two training (or debiasing) strategies: (1) Path Type Adaptive Pretraining (§4.2), which aims to improve the adaptability of the model to various UG paths; and (2) Complexity Ranking Guided Attention mechanism (§4.3), which enables the model to learn from both simple and complex UG paths. Experimental results on both biomedical and NYT10 (Riedel et al., 2010) datasets prove that: (1) UG paths have the potential to bring performance gain for DS-RE as compared with KG paths; (2) the proposed training methods are effective to fully exploit the potential of UG paths for DS-RE because the proposed methods significantly and consistently outperform several baselines on both datasets and especially achieve a new state-of-the-art result on the NYT10 dataset.

2 Related Work

To improve the performance of a DS-RE model, recently, researchers introduce various attention mechanisms. Lin et al. (2016) propose a relation vector based attention mechanism. Jat et al. (2018); Du et al. (2018) propose multi-level (e.g., word-level and sentence-level) structured attention mechanism. Ye and Ling (2019) apply both intra-bag and inter-bag attention for DS-RE. Han et al. (2018b) propose a relation hierarchy based attention mechanism. Han et al. (2018a) propose a joint model that adopts a KG embeddings based attention mechanism. Jia et al. (2019) propose an attention regularization framework to select informative sentence evidences for DS-RE. However, these models rely only on noisy sentence evidences from DS, neglecting the rich UG paths for DS-RE.

Besides the sentence evidences from DS, researchers also leverage external evidences for DS-RE. Ji et al. (2017) apply entity descriptions generated from Freebase and Wikipedia as extra evidences, Lin et al. (2017) utilize multilingual text as extra evidences and Vashishth et al. (2018) use multiple extra evidences including entity types, dependency and relation alias information for DS-RE. Alt et al. (2019) utilize pretrained language model

as background information for DS-RE. Sun et al. (2019) apply relational table extracted from Web as supplementary evidences for DS-RE.

To apply DS-RE beyond sentence boundary, Quirk and Poon (2017) utilize syntactic information to extract relation from neighboring sentences. Zeng et al. (2017) apply two-hop KG paths identified from two-hop textual paths as extra evidences for DS-RE. Different from this work, we directly use the rich UG paths as extra evidences. Dai et al. (2019) extend the framework of Han et al. (2018a) by introducing multiple KG paths as extra evidences for DS-RE. Neelakantan et al. (2015); Das et al. (2017a) use multiple reasoning paths over Text and KG for relation prediction in the paradigm of Knowledge Graph Completion. Our work differs from the ones mentioned above in two ways: (i) We utilize the UG paths as extra evidences for the task of DS-RE from text, (ii) We take into account the factor of attention bias while encoding UG paths and propose two effective debiasing methods to exploit the potential of UG paths for DS-RE.

3 Base Model

We select the DS-RE model proposed by Dai et al. (2019) as our base model and extend it into our UG setting. Given a target entity pair (e_1, e_2), a bag of corresponding sentence evidences $S_r = \{s_1, ..., s_n\}$ and a bag of UG paths $P_r = \{p_1, ..., p_m\}$, the base model aims to measure the probability of (e_1, e_2) having a predefined relation r (including the empty relation NA). The base model consists of four main modules: KG Encoder, Sentence Evidence Encoder, Path Evidence Encoder and Relation Classification Layer, as shown in Figure 2.

3.1 KG Encoder

Suppose we have a KG containing a set of fact triplets $\mathcal{O} = \{(e_1, r, e_2), ...\}$, where each fact triplet consists of two entities $e_1, e_2 \in \mathcal{E}$ and their relation $r \in \mathcal{R}$. Here $\mathcal{E}$ and $\mathcal{R}$ stand for the set of entities and relations respectively.

The KG Encoder then encodes $e_1, e_2 \in \mathcal{E}$ and their relation $r \in \mathcal{R}$ into low-dimensional vectors $\mathbf{h}, \mathbf{t} \in R^d$ and $\mathbf{r} \in R^d$ respectively, where d is the dimensionality of the embedding space. The KG Encoder adopts TransE (Bordes et al., 2013) to score a given triplet. Specifically, given a triplet (e_1, r, e_2), TransE evaluates its plausibility
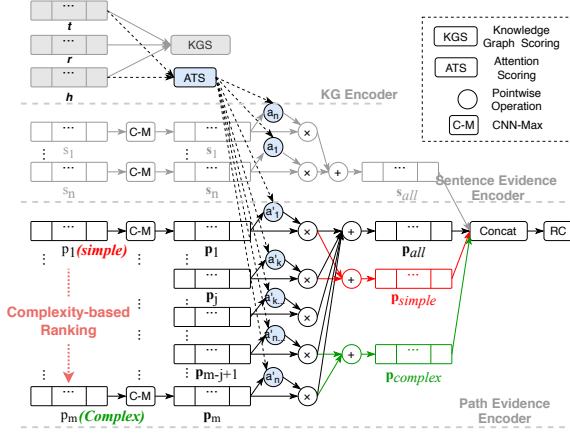
Figure 2: Overview of the base model and our proposed Complexity Ranking Guided Attention mechanism (§4.3). The base model takes the sentence evidences (e.g., s_1, ...) containing a target entity pair and the UG paths (e.g., p_1, ...) connecting the entity pair as input for predicting their relation. The KG embeddings of the entity pair (i.e., $\mathbf{h}$ and $\mathbf{t}$) are used for calculating the attention over these sentences and paths. The Complexity Ranking Guided Attention mechanism is proposed to force the model to attend both simple UG paths (e.g., $p_1 \sim p_j$) and complex ones (e.g., $p_{m-j+1} \sim p_m$).

via Equation 1:

$$f_r(e_1, e_2) = b - \|\mathbf{r}_{ht} - \mathbf{r}\|, \qquad (1)$$

$$\mathbf{r}_{ht} = \mathbf{t} - \mathbf{h}, \qquad (2)$$

where b is a bias constant and $\mathbf{r}_{ht}$ is a latent relation embedding for (e_1, e_2). The conditional probability can be formalized over all fact triplets $\mathcal{O}$ as follows:

$$P(e_1, r, e_2 | \theta_{\mathcal{E}}, \theta_{\mathcal{R}}) = \frac{\exp(f_r(e_1, e_2))}{\sum_{r' \in \mathcal{R}} \exp(f_{r'}(e_1, e_2))} \qquad (3)$$

where $\theta_{\mathcal{E}}$ and $\theta_{\mathcal{R}}$ are parameters for entities and relations respectively.

3.2 Sentence Evidence Encoder

Given a bag of sentence evidences $S_r = \{s_1, ..., s_n\}$, the Sentence Evidence Encoder applies CNN-Max (see Appendix §A.1) on each sentence, namely $\mathbf{s}_i = \text{CNN-Max}(s_i)$, to derive the sentence representations $\{\mathbf{s}_1, ..., \mathbf{s}_n\}$. The encoder then calculates the bag-level vector representation

$\mathbf{s}_{\text{all}}$ via Equation 4:

$$\mathbf{s}_{\text{all}} = \sum_{i=1}^{n} a_i \mathbf{s}_i, \qquad (4)$$

$$a_i = \frac{\exp(\langle \mathbf{r}_{ht}, \mathbf{x}_i \rangle)}{\sum_{k=1}^{n} \exp(\langle \mathbf{r}_{ht}, \mathbf{x}_k \rangle)},$$

$$\mathbf{x}_i = \tanh(\mathbf{W}\mathbf{s}_i + \mathbf{b})$$

where $\mathbf{r}_{ht}$ is from Equation 2, $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ is the bias vector, a_i is the weight for the i-th sentence in S_r.

3.3 Path Evidence Encoder

Given a bag of UG paths $P_r = \{p_1, ..., p_m\}$ connecting an entity pair of interest (e_1, e_2), the Path Evidence Encoder encodes them into a bag-level vector representation $\mathbf{p}_{\text{all}}$. Since we represent a path as a sequence of words (or a long sentence), as shown in Figure 1, analogously to the Sentence Evidence Encoder, we apply a CNN-Max (see Appendix §A.1) to encode each path p_i, namely $\mathbf{p}_i = \text{CNN-Max}(p_i)$. The bag-level path representation $\mathbf{p}_{\text{all}}$ for P_r is then calculated via Equation 5:

$$\mathbf{p}_{\text{all}} = \sum_{i=1}^{m} a'_i \mathbf{p}_i, \qquad (5)$$

$$a'_i = \frac{\exp(\langle \mathbf{r}_{ht}, \mathbf{x}'_i \rangle)}{\sum_{k=1}^{m} \exp(\langle \mathbf{r}_{ht}, \mathbf{x}'_k \rangle)},$$

$$\mathbf{x}'_i = \tanh(\mathbf{W}\mathbf{p}_i + \mathbf{b})$$

where a'_i is the weight for the i-th path in P_r.

3.4 Relation Classification Layer

The conditional probability of (e_1, e_2) having a relation r is formulated via Equation 6:

$$P(e_1, r, e_2 | S_r, P_r, \theta_S, \theta_P) = \frac{\exp([\mathbf{o}]_r)}{\sum_{c=1}^{n_r} \exp([\mathbf{o}]_c)} \qquad (6)$$

where $\mathbf{o} = \mathbf{M}[\mathbf{s}_{\text{all}}; \mathbf{p}_{\text{all}}] + \mathbf{d}$, θ_S, θ_P are the parameters in Sentence Evidence Encoder and Path Evidence Encoder, $\mathbf{M}$ is the representation matrix of relations, $\mathbf{d}$ is a bias vector, $\mathbf{o}$ is the output vector containing the prediction scores of all predefined relations, $[\mathbf{o}]_c$ is the prediction score for the relation c, and n_r is the total number of relations.

Given a training dataset consisting of triplets $\mathcal{O} = \{(e_1^1, r^1, e_2^1), (e_1^2, r^2, e_2^2), ...\}$, we minimize the objective function as follows:

$$J(\theta) = -\frac{1}{|\mathcal{O}|} \sum_{i=1}^{|\mathcal{O}|} \log P(e_1^i, r^i, e_2^i | \theta_{\mathcal{E}}, \theta_{\mathcal{R}}) \qquad (7)$$

$$+ \log P(e_1^i, r^i, e_2^i | S_{r^i}, P_{r^i}, \theta_S, \theta_P)$$

3676

The base model is optimized with Stochastic Gradient Descent (SGD). Following (Han et al., 2018a), we optimize $P(e_1^i, r^i, e_2^i | \theta_{\mathcal{E}}, \theta_{\mathcal{R}})$ and $P(e_1^i, r^i, e_2^i | S_i, P_i, \theta_S, \theta_P)$ in parallel.

4 Proposed Training Method

4.1 Problem of Attention Bias

While extending the base model into our UG setting, we observe that the base model tends to allocate more attention to KG or linguistically simple paths as compared to Textual paths (i.e., the path comes form Text), Hybrid paths (i.e., the path comes from both Text and KG), or linguistically complex ones, as shown in Figure 3a and Figure 3b. We consider that this would be because paths including Textual relations (i.e., Textual and Hybrid paths) or complex paths are comparatively noisier than KG or simple paths, but which does not necessarily mean the former is not useful. For instance, in Figure 1, the complex Hybrid path p_2 is useful for predicting (**Colesevelam_HCl**, *may_treat*, **Type_2_Diabetes**), because p_2 implies a plausible line of reasoning " ***Colesevelam_HCl*** $\underrightarrow{alternative\_to}$ *Colestipol* $\underrightarrow{may\_treat}$ *hyperglyceridemia* $\underrightarrow{strong\_link\_to}$ **Type_2_Diabetes**". However, due to the attention bias mentioned above, the base model allocates low attention ($a_2' \approx 8.0 \times 10^{-36}$) on the informative path, and thus fails to learn from such complex but useful evidences.

To reduce the negative effect of the attention biases and make full use of the UG path, we propose the following two training (or debiasing) strategies: Path Type Adaptive Pretraining (§4.2) and Complexity Ranking Guided Attention (§4.3).

4.2 Path Type Adaptive Pretraining

As shown in Figure 3a, the base model tends to bias toward KG paths. This indicates that the base model mainly relies on KG paths so that it is incapable of capturing informative features from Textual and Hybrid paths. This bias will decrease the flexibility and adaptability of the base model to different types of paths.

To address this issue, we propose a debiasing strategy called Path Type Adaptive Pretraining. In this strategy, we pretrain the base model sequentially using Textual, Hybrid, and KG Paths as path evidences, and then finetune it with all types of paths as illustrated in Figure 4. We hypothesize that this strategy can prevent the reliance on a single type of UG path and improve the capacity of extracting features from the entire UG paths, and thereby increase the performance.

4.3 Complexity Ranking Guided Attention

Similar to the bias towards KG paths, the base model also focuses on linguistically simple paths, as shown in Figure 3b, even though complex ones are informative (e.g. p_2 in Figure 1). We hypothesize that restricting the attention span to the complex (simple) paths can force the model to pay attention to the complex (simple) paths, thereby effectively utilize them. Under this hypothesis, we propose a Complexity Ranking Guided Attention mechanism, as illustrated in Figure 2.

Specifically, given a bag of paths $P_r = \{p_1, ..., p_m\}$, we rank them according to their complexity scores (κ), which are calculated via $\kappa = \lambda_1 \tau_1 + \lambda_2 \tau_2 + ...$, where τ denotes the feature for capturing linguistic complexity (e.g., path length) and λ is a corresponding weight, which is a hyperparameter. Sentence length (i.e., the number of tokens in a sentence) and lexical richness (i.e., the number of token types) are commonly used features for evaluating sentence complexity (Brunato et al., 2018). Therefore, this work adopts them to calculate the complexity for a given path.

Then, we group top j most and least complex paths into a set of complex and simple paths respectively, where j is a hyperparameter[1]. The set level representation is calculated via the Equation 8.

$$\mathbf{p}_{\text{complex or simple}} = \sum_{i \in top\_j \text{ or } i \in last\_j} a_i' \mathbf{p}_i \quad (8)$$

$$a_i' = \frac{\exp(\langle \mathbf{r}_{ht}, \mathbf{x}'_i \rangle)}{\sum_{k \in top\_j \text{ or } k \in last\_j} \exp(\langle \mathbf{r}_{ht}, \mathbf{x}'_k \rangle)},$$
$$\mathbf{x}'_i = \tanh(\mathbf{W}\mathbf{p}_i + \mathbf{b})$$

Finally, we concatenate the resulting representation $\mathbf{s}_{\text{final}}$, $\mathbf{p}_{\text{final}}$, $\mathbf{p}_{\text{simple}}$ and $\mathbf{p}_{\text{complex}}$ as the input to the relation classification layer. The conditional probability $P(e_1, r, e_2 | S_r, P_r, \theta_S, \theta_P)$ is formulated via Equation 9, where $\mathbf{o} = \mathbf{M}[\mathbf{s}_{\text{all}}; \mathbf{p}_{\text{all}}; \mathbf{p}_{\text{complex}}; \mathbf{p}_{\text{simple}}] + \mathbf{d}$.

$$P(e_1, r, e_2 | S_r, P_r, \theta_S, \theta_P) = \frac{\exp([\mathbf{o}]_r)}{\sum_{c=1}^{n_r} \exp([\mathbf{o}]_c)} \quad (9)$$

---

[1]In our experiments, we set j as 30 for NYT10 dataset and 50 for Biomedical dataset.

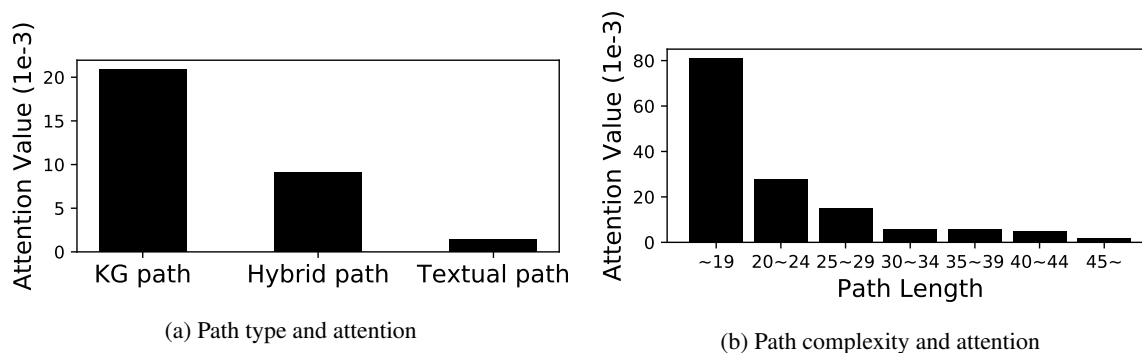(a) Path type and attention   (b) Path complexity and attention

Figure 3: (**Left**) Average attention weights across different path types. (**Right**) Average attention weights across different length range of paths. Here, we use path length as an indicator of path complexity, which denotes the number of words in a path.
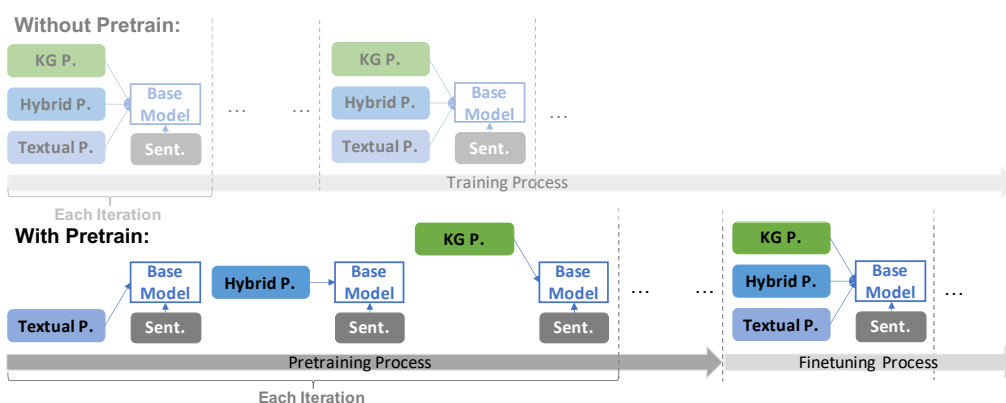

Figure 4: Path Type Adaptive Pretraining strategy, where "Textual/Hybrid/KG P." represent Textual, Hybrid, and KG paths respectively, and "Sent." represents the sentence evidences. In this strategy (**Pretrain**), instead of using all types of paths to train the base model in all iterations, we sequentially train the model with Textual, Hybrid and KG paths, and then finetune it with all types of paths.

5 Experiments

5.1 Data

We evaluate our proposed framework on a biomedical dataset and NYT10 dataset ([Riedel et al., 2010](#)). The statistics of both datasets is summarized in Table 1. We will detail both datasets as follows.

Biomedical Dataset. This datatset is created by linking biomedical KG with biomedical Text. We choose UMLS[2] and Medline corpus as the biomedical KG and Text respectively. UMLS is a frequently used biomedical knowledge base, while Medline corpus is a large collection of biomedical abstracts, both are developed and maintained by the U.S. National Library of Medicine[3]. For identifying UMLS entity mentions in the Medline corpus, we use a state-of-the-art UMLS Named En-

tity Recognizer (NER), ScispaCy ([Neumann et al., 2019](#)). The NER identifies UMLS concepts and annotates them by their corresponding UMLS Concept Unique Identifier (CUI) and entity types.

From the UMLS KG and the entity linked Medline corpus, we extract fact triplets (i.e., (e_1, r, e_2)) and corresponding sentence evidences containing (e_1, e_2) under the restriction that: (1) each entity pair should be connected by a RO (RO stands for "has Relationship Other than synonymous, narrower, or broader") relationship; (2) each entity should belong to the following entity types: Protein, Gene, Disease or Syndrome, Enzyme, Chemical, Sign or Symptom and Pharmacologic Substance. Then we divide the collected triplets and sentence evidences into training and testing set according to the year when the source abstract of sentence evidence was published. The former is aligned to the years until 2008 and the latter to the years 2009 $\sim$ 2018, ensuring the testing set only contains the

	#R	#EP	#Related EP	#Sentence	#UG Path
Biomedical	40	100,549 / 21,081	10,936 / 1,804	165,692 / 28,912	12,854,696 / 2,346,007
NYT10	53	281,270 / 96,678	18,252 / 1,950	522,611 / 172,448	8,967,153 / 2,984,611

Table 1: Statistics of datasets in this work, where **R** and **EP** stand for the target Relation and Entity Pair, $\#_1/\#_2$ represent the number of training and testing data respectively.

unobserved triplets.

To simulate the noise in the real world, besides the "related" triplets, we also extract the "unrelated" triplets and sentence evidences based on a closed world assumption: pairs of entities not listed in a KG are regarded to have NA relation and sentences containing them are considered to be the NA sentence evidences. We divide the NA triplets and NA sentence evidences in the same way mentioned above. We use a subset of UMLS (see Appendix §A.3) and the Medline abstracts published until 2008 as the KG and Text respectively to create the UG for path retrieval. In addition, we use the same subset of UMLS triplets mentioned above to train the KG Encoder introduced in §3.

NYT10. The dataset is created by aligning Freebase relational facts with the New York Times Corpus. Sentence evidences from the year $2005 \sim 2006$ are used for training and the evidences from 2007 are used for testing. NYT10 dataset has been widely used by (Lin et al., 2016; Ji et al., 2017; Du et al., 2018; Jat et al., 2018; Du et al., 2018; Han et al., 2018a,b; Vashishth et al., 2018; Ye and Ling, 2019; Alt et al., 2019). We use Freebase[4] and ClueWeb12 with Freebase entity mention annotations (Gabrilovich et al., 2013) as the KG and Text to create the UG for path searching. In addition, following (Han et al., 2018a), we use FB60K for training the KG Encoder.

UG path search. Given an entity pair (e_1, e_2), the UG path set P_r is obtained by performing random walks over the UG from e_1 till e_2 with maximum step[5].

5.2 Settings

We follow (Lin et al., 2016) and conduct the held-out evaluation, in which the model for DS-RE is evaluated by comparing the fact triplets identified from evidences (i.e., the bag of sentence evidences

---

[4]From the entire Freebase, we only collect the triplets with the relations that are mentioned in NYT10 dataset for UG creation, ensuring not to overlap with testing set.

[5]We manually set the maximum step as 3.

S_r and the bag of UG path evidences P_r) with those in KG. Following the evaluation of previous works, we draw Precision-Recall curves and report the Area Under Curve (AUC) and Precision@N (P@N) metrics, which gives the percentage of correct triplets among top N ranked candidates. The parameter settings of our experiments are detailed in Appendix §A.2.

To demonstrate the effectiveness of our framework, we choose the model proposed by Dai et al. (2019) as the baseline model, because this is the closest model in terms of incorporating multiple paths for DS-RE. Henceforth, "Sent+KG" is the baseline model, which uses both sentence evidences and KG paths. "Sent+UG" represents the base model in §3 which takes UG paths instead of KG paths as path evidences. "Sent+UG+Pretrain" and "Sent+UG+Ranking" denote the base model trained with Path Type Adaptive Pretraining strategy and the base model with Complexity Ranking Guided Attention mechanism, respectively. "Sent+UG+Ranking+Pretrain" means the base model trained with both strategies.

5.3 Results and Discussion

Precision-Recall Curves. The Precision-Recall (PR) curves of each model on the biomedical and NYT10 datasets are shown in Figure 5 and Figure 6, respectively. The results show that: (1) "Sent+UG" does not have obvious advantages than "Sent+KG", illustrating that due to the biases discussed in §4.1, simply applying UG paths on the base model has limited effect on improving the performance of DS-RE. (2) "Sent+UG+Pretrain" and "Sent+UG+Ranking" achieve better overall performance than "Sent+KG" on both datasets, especially when the recall is greater than 0.3, demonstrating that UG has the potential to enhance the performance and the two proposed debiasing strategies are effective for exploiting the potential of UG for DS-RE. (3) "Sent+UG+Ranking+Pretrain" achieves the highest precision over the (almost) entire recall range on both datasets, proving that the two proposed strategies have a mutual complementary relationship on exploiting UG for DS-RE. This is understandable because the two proposed strategies deal with different types of biases, in addition, "Pretrain" helps the base model adapt to UG paths by effectively tuning its weights, while "Ranking" enhances the base model by adjusting its attention mechanism. (4) The consistent improvement on
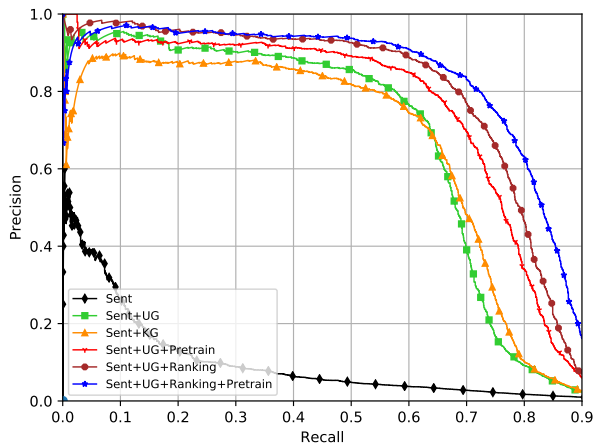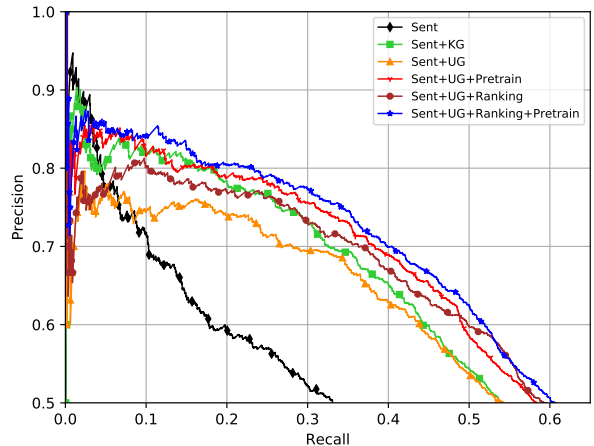
Figure 5: PR curves on Biomedical dataset.

Figure 6: PR curves on NYT10 dataset.

Model	Biomedical dataset					NYT10 dataset					
	AUC	P@0.5k	P@1k	P@1.5k	P@2k	AUC	P@0.1k	P@0.2k	P@0.3k	P@0.5k	P@1k	P@2k
Sent	9.6	30.0	20.8	18.7	16.3	36.6	81.0	73.5	68.3	62.0	53.8	40.2
Sent+KG	62.6	91.4	86.1	74.2	58.5	50.2	80.0	82.0	81.3	77.2	67.9	50.3
Sent+UG	61.0	87.6	83.4	73.8	58.5	48.4	74.0	76.0	74.7	74.0	66.7	50.3
Sent+UG +Pretrain	70.1	95.4	89.7	76.3	60.4	52.7	83.0	82.0	80.3	78.6	70.4	52.6
Sent+UG +Ranking	74.2	95.2	92.2	81.1	62.2	52.1	77.0	80.0	79.3	77.4	70.3	54.4
Sent+UG +Ranking +Pretrain	**77.5**	**95.4**	**93.1**	**83.9**	**64.4**	**55.0**	**86.0**	**84.0**	**83.3**	**80.4**	**71.9**	**54.5**

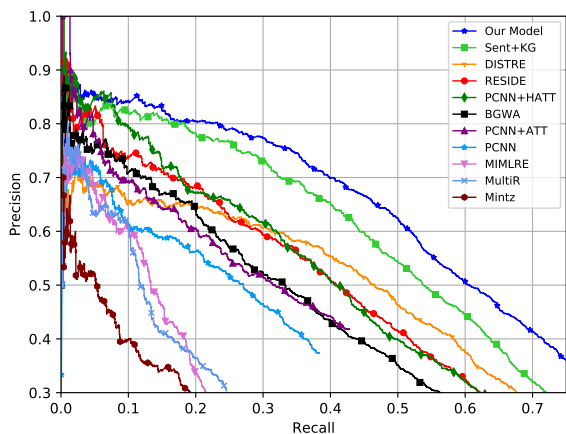Table 2: P@N and AUC on Biomedical and NYT10 dataset (k=1000).

Figure 7: PR curves of previous state-of-the-art methods and our proposed model on NYT10 dataset.

two datasets from different domains further proves the validity of our proposed methods.

AUC and P@N Evaluation. Table 2 further presents the results in terms of AUC and P@N. From them, we have similar observation to the PR curves. We also observe that the effectiveness of UG paths is more pronounced on Biomedical dataset than on NYT10 dataset. We speculate that compared to the generic NYT10 dataset, further Background Knowledge (BK) is needed to identify relations from Biomedical dataset, and UG paths could be utilized as the BK to facilitate the scientific DS-RE.

Comparison with State-of-the-art Baselines on NYT10. To demonstrate the effectiveness of our proposed model, we also compare it against the following baselines on NYT10 dataset: Mintz (Mintz et al., 2009), MultiR (Hoffmann et al., 2011), MIMLRE (Surdeanu et al., 2012), PCNN (Zeng et al., 2015), PCNN+ATT (Lin et al., 2016), BGWA (Jat et al., 2018), PCNN+HATT (Han et al., 2018b), RESIDE (Vashishth et al., 2018), DIS-TRE (Alt et al., 2019) and Sent+KG (Dai et al., 2019). The results shown in Figure 7 and Table 3 indicate that: (1) our selected base model, "Sent+KG", is a strong baseline because it signif-

System	AUC	P@0.1k	P@0.2k	P@0.3k	P@0.5k	P@1k	P@2k
Mintz†	10.7	52.3	50.2	45.0	39.7	33.6	23.4
PCNN+ATT†	34.1	73.0	68.0	67.3	63.6	53.3	40.0
RESIDE†	41.5	81.8	75.4	74.3	69.7	59.3	45.0
PCNN+HATT‡	42.0	81.0	79.5	75.7	68.0	58.6	42.1
DISTRE†	42.2	68.0	67.0	65.3	65.0	60.2	47.9
Sent+KG	50.2	80.0	82.0	81.3	77.2	67.9	50.3
Our Model	**55.0**	**86.0**	**84.0**	**83.3**	**80.4**	**71.9**	**54.5**

Table 3: P@N and AUC from previous state-of-the-art DS-RE models and our proposed model on NYT10 dataset, where †represents that these results are quoted from (Alt et al., 2019) and ‡indicates the results using the pretrained model from (Han et al., 2018b).

icantly outperforms other state-of-the-art models; and (2) our model can effectively take advantage of the rich UG paths for DS-RE because it beats the strong baseline and achieves a new state-of-the-art result on the commonly used DS-RE dataset.

Base	Prop.	**Biomedical Triplet**
✗	✓	(Beta-2...Gene , gene_associated_with_disease, Asthma)
		Multi-hop Path
Low	**High**	hop_1: *"The human Beta-2...Gene is responsible for the binding of endogenous Catecholamine and their ..."* hop_2: *" Catecholamine chemical structure of Epinephrine "* hop_3: *" Epinephrine may treat Asthma "*.

Base	Prop.	**NYT10 Triplet**
✗	✓	(San_Francisco , /location/contains, Noe_Valley)
		Multi-hop Path
Low	**High**	hop_1: *" San_Francisco /location/contains Fort_Point "* hop_2: *"Surf spots and surfing regions include Northern CA, the Bay_Area , San Francisco, Ocean Beach and Fort Point "* hop_3: *" Bay_Area /location/contains Noe_Valley "*

Table 4: Some examples of attention distribution over paths from "Sent+UG" (Base) and "Sent+UG+Ranking+Pretrain" (Prop.), where ✓(or ✗) represents the correct (or incorrect) prediction of the target relation.

Case Study. Table 4 shows the UG path examples that are scored with highest ("High") or lowest (or lower than 1.0×10^{-3}) ("Low") attention by the base model and our proposed framework. The paths in the table generally mean " Beta-2... Gene $\xrightarrow{is\_responsible\_for}$ Catecholamine $\xrightarrow{is\_the\_chemical\_class\_of}$ Epinephrine $\xrightarrow{may\_treat}$ Asthma " and " San_Francisco $\xrightarrow{contains}$ Fort_Point $\xrightarrow{equal\_status}$ Bay_Area $\xrightarrow{contains}$ Noe_Valley ", and thus can be seen as the useful path evidences for identifying *gene_associated_with_disease* and */location/contains* relation respectively. These examples indicate that our proposed training

strategies could help the base model attend such informative UG paths so that it can correctly identify the target relation.

6 Conclusion and Future Work

We have introduced UG paths as extra evidences for the task of DS-RE from text. In order to fully take advantage of the rich UG paths, we have proposed two training (or debiasing) strategies: Path Type Adaptive Pretraining and Complexity Ranking Guided Attention mechanism. We have conducted experiments on both biomedical and NYT10 datasets. The results show that the two proposed methods are effective for exploiting the potential of UG paths for improving the performance of DS-RE.

In the future, we plan to carry out the following steps: (1) we further investigate how the proposed training methods influence the performance via manual analysis so as to better the efficiency; and (2) instead of random walk, we may collect UG paths by adopting more sophisticated mechanisms such as training a path searching agent via reinforcement learning to prevent redundant and noisy paths.

Acknowledgement

References

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. *arXiv preprint arXiv:1906.08646*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Dominique Brunato, Lorenzo De Mattei, Felice Dell'Orletta, Benedetta Iavarone, and Giulia Venturi. 2018. Is this sentence difficult? do you agree? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2690–2699.

Qin Dai, Naoya Inoue, Paul Reisert, Takahashi Ryo, and Kentaro Inui. 2019. Incorporating chains of reasoning over knowledge graph for distantly supervised biomedical knowledge acquisition. In *Proceedings of the 33nd Pacific Asia Conference on Language, Information and Computation (PACLIC33)*, pages 19–28, Hakodate, Japan. Waseda Institute for the Study of Language and Information.

Rajarshi Das, Arvind Neelakantan, David Belanger, and Andrew McCallum. 2017a. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 132–141.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017b. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–365.

Jinhua Du, Jingguang Han, Andy Way, and Dadong Wan. 2018. Multi-level structured self-attentions for distantly supervised relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2216–2225.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0).

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018a. Neural knowledge acquisition via mutual attention between knowledge graph and text. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xu Han, Pengfei Yu, Zhiyuan Liu, Maosong Sun, and Peng Li. 2018b. Hierarchical relation extraction with coarse-to-fine grained attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2245.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Sharmistha Jat, Siddhesh Khandelwal, and Partha Talukdar. 2018. Improving distantly supervised relation extraction using word and entity based attention. *arXiv preprint arXiv:1804.06987*.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. Arnor: attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1399–1408.

Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Neural relation extraction with multi-lingual attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 34–43.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2124–2133.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Arvind Neelakantan, Benjamin Roth, and Andrew McCallum. 2015. Compositional vector space models for knowledge base completion. 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 156–166, Beijing, China. Association for Computational Linguistics.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Huan Sun et al. 2019. Leveraging 2-hop distant supervision from table entity pairs for relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 410–420.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1499–1509.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha 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.

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, Volume 1 (Long and Short Papers)*, pages 2810–2819.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344.

Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating relation paths in neural relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1768–1777.

A   Appendix

A.1   CNN-Max

Convolutional Neural Network with Max pooling layer (CNN-Max) is adopted to derive the sentence representation $\mathbf{s}$ and path representation $\mathbf{p}$. Specifically, vector representation $\mathbf{v}_t$ for each word w_t is calculated via Equation 10, where $\mathbf{W}^w_{emb}$ is a word embedding projection matrix (Mikolov et al., 2013), $\mathbf{W}^{wp}_{emb}$ is a word position embedding projection matrix (Zeng et al., 2014), $\mathbf{x}^w_t$ is a one-hot word representation and $\mathbf{x}^{wp}_t$ is a one-hot word position representation, which indicates the relative distance between the current word and the target entity pair.

$$\mathbf{v}_t = [\mathbf{v}^w_t; \mathbf{v}^{wp1}_t; \mathbf{v}^{wp2}_t], \qquad (10)$$
$$\mathbf{v}^w_t = \mathbf{W}^w_{emb}\mathbf{x}^w_t,$$
$$\mathbf{v}^{wp1}_t = \mathbf{W}^{wp}_{emb}\mathbf{x}^{wp1}_t,$$
$$\mathbf{v}^{wp2}_t = \mathbf{W}^{wp}_{emb}\mathbf{x}^{wp2}_t$$

The sentence representation $\mathbf{s}$ and path representation $\mathbf{p}$ are formulated via the Equation 11, where $\mathbf{W}^{sent}$ (or $\mathbf{W}^{path}$) is the convolution kernal, $\mathbf{b}^{sent}$ (or $\mathbf{b}^{path}$) is the corresponding bias vector, $\mathbf{v}^{sent}_t$ (or $\mathbf{v}^{path}_t$) is the vector for each word w_t in a sentence (or path), $[vec]_i$ is the i-th value of vec, ν is the dimensionality of $\mathbf{s}$ and $\mathbf{p}$, and k is the convolutional window size.

$$[\mathbf{s}]_i = \max_t\{[\mathbf{h}^{sent}_t]_i\}, \ \forall i = 1, ..., \nu$$
$$[\mathbf{p}]_i = \max_t\{[\mathbf{h}^{path}_t]_i\}, \ \forall i = 1, ..., \nu \qquad (11)$$
$$\mathbf{h}^{sent}_t = \tanh(\mathbf{W}^{sent}\mathbf{z}^{sent}_t + \mathbf{b}^{sent}),$$
$$\mathbf{h}^{path}_t = \tanh(\mathbf{W}^{path}\mathbf{z}^{path}_t + \mathbf{b}^{path}),$$
$$\mathbf{z}^{sent}_t = [\mathbf{v}^{sent}_{t-(k-1)/2}; ...; \mathbf{v}^{sent}_{t+(k-1)/2}],$$
$$\mathbf{z}^{path}_t = [\mathbf{v}^{path}_{t-(k-1)/2}; ...; \mathbf{v}^{path}_{t+(k-1)/2}]$$

A.2   Parameter Settings

All of the hyperparameters used in our experiments are listed in Table 5. Most of them follow the hyperparameter setting in (Dai et al., 2019) and (Han et al., 2018a). We use a Word2Vec model[6] to train the word embeddings on the UMLS entity linked corpus for the biomedical dataset, and adopt the word embeddings released by (Lin et al., 2016) for NYT10 dataset. We apply Stochastic Gradient Descent (SGD) to optimize the proposed DS-RE model.

---

[6]Gensim   word2vec   implementation: `https://radimrehurek.com/gensim/models/word2vec.html`

Hyperparameter	Biomedical	NYT10
word embedding dimension	50	50
KG embedding dimension	50	50
position embedding dimension	5	5
CNN window size	3	3
CNN filter number	100	230
dropout rate	0.5	0.5
learning rate (for sentences and paths)	0.02	0.05
learning rate (for KG)	0.05	0.001
batch size	50	160

Table 5: Hyperparameters used in our experiments.

Selected Entity Types
Antibiotic, Biologically Active Substance, Bacterium, Organ, Cell Component, Cell Function, Cell, Clinical Drug, Ion, Eukaryote, Food, Genetic Function Hazardous or Poisonous Substance, Hormone, Immunologic Factor, Inorganic Chemical, Organic Chemical, Pathologic Function, Receptor, Steroid, Virus and Vitamin.

Table 6: List of selected UMLS entity types.

A.3   Subset of UMLS

Besides the 7 entity types mentioned above, we also use other 22 entity types, as listed in Table 6, to collect the UMLS triplets that are connected by RO relationship, ensuring all testing triplets are removed. The main reasons to manually restrict the entity type is because (1) we observe that most of the Medline abstracts discuss the relationship among these entity types; (2) these concrete entities could prevent semantic drift while searching UG paths.