Generating Labeled Data for Relation Extraction: A Meta Learning Approach with Joint GPT-2 Training

Amir Pouran Ben Veyseh[1], **Franck Dernoncourt**[2], **Bonan Min**[3]*, **and Thien Huu Nguyen**[1]
[1] Department of Computer Science, University of Oregon, Eugene, OR, USA
[2] Adobe Research, Seattle, WA, USA
[3] Amazon AWS AI Labs
{apouranb,thien}@cs.uoregon.edu,
dernonco@adobe.com, bonanmin@amazon.com

Abstract

Relation Extraction (RE) is the task of identifying semantic relation between real-world entities mentioned in text. Despite significant progress in RE research, a remaining challenge for RE concerns the lack of training data for data-hungry deep learning models. Cost of annotation and difficulty of the task are among hindrance to collect a large-scale RE dataset in different domains. To address this limitation, we propose a novel framework to automatically generate labeled data for RE. Our framework presents the pre-trained language model GPT-2 for data generation. In addition, to optimize the generated samples for an RE model, we introduce a meta learning approach to allow the GPT-2 model to be updated during the training process for RE. In particular, to leverage the feedback from the RE model to improve the data generation from GPT-2, we propose a novel reward function to update the GPT-2 model with REINFORCE, seeking to promote the similarity of the RE loss function's gradients computed for generated data and a meta development set. We conduct extensive experiments on two benchmark datasets to produce state-of-the-art performance for RE.

1 Introduction

One of the fundamental tasks in Information Extraction (IE) involves Relation Extraction (RE) that aims to identify semantic relations between two entities mentioned in textual data. For instance, in the sentence "*After XZY's decision to move to* ***Europe****, they selected* ***Paris*** *as the final location for their headquarters.*", the semantic relation *PART-Whole* between two entity mentions "*Europe*" and "*Paris*" should be detected. An RE system can be employed to populate a knowledge base with relations among entities, provide information for question answering systems, and present facts for text summerization tools.

Due to the importance of RE, in recent years various methods and models have been proposed for this task. These models can be categorized into feature-based (Zelenko et al., 2003; Zhou et al., 2005; Bunescu and Mooney, 2005; Sun et al., 2011; Chan and Roth, 2010; Nguyen and Grishman, 2014; Nguyen et al., 2015c) and deep learning (Zeng et al., 2014; Nguyen and Grishman, 2015a; dos Santos et al., 2015; Wang et al., 2016; Nguyen and Grishman, 2016; Zhou et al., 2016; Zhang et al., 2017; Nguyen et al., 2019a) models. The existing models provide solutions for RE in various settings including monolingual (Zhang et al., 2018), cross-lingual (Ni et al., 2020), cross-domain (Pouran Ben Veyseh et al., 2020), and joint models (Nguyen et al., 2021, 2022). Despite those progress, one limitation that hinders on-going research for RE is labeled data scarcity. Annotating a large-scale RE dataset is challenging, due to the expensive nature of annotation task and the high requirement for expertise in specific domains. As such, prior methods have resorted to distantly supervised setting (Mintz et al., 2009; Zeng et al., 2015; Ji et al., 2017) or pseudo labeling techniques (Hu et al., 2021b,a) that leverage vast amounts of unlabeled data to address the labeled data scarcity issue for RE. Although these methods are helpful to substantially increase the size of RE datasets, they also introduce massive noisy samples which might hurt the training of an RE model. Consequently, creating cost-efficient large-scale labeled datasets for specific domains remains highly challenging for RE.

To achieve large-scale labeled datasets, in this work we introduce a novel data augmentation method to automatically generate labeled data for RE. In particular, instead of using unlabeled data, we propose to employ the pre-trained language model GPT-2 (Radford et al., 2019) to generate synthetic labeled data for RE. In our method, the GPT-2 model is first fine-tuned on available manually labeled RE datasets. Concretely, the language

* Work done at Raytheon BBN Technologies (prior to joining AWS AI).

Findings of the Association for Computational Linguistics: ACL 2023, pages 11466–11478

model is trained on the label-augmented sentences in which positive and negative RE samples are marked with special tags surrounding two input entity mentions. Next, the fine-tuned GPT-2 model is employed to generate new label-augmented in-domain sentences that can be map back to produce new labeled data for RE. The new labeled data is then combined with the original manually labeled data to train an RE model. However, an issue with this approach involves the separation between the fine-tuning process of GPT-2 and the target RE model that might cause a mismatch between the generated data from GPT-2 and the data expected by the RE model (e.g., the generated data can be noisy or redundant for RE). As such, to improve the effectiveness of the generated data for an RE model, we propose to further optimize GPT-2 parameters during the training of the RE model, thus enabling the interactions between the GPT-2 and RE models to generate optimal/customized data for RE. In particular, we propose a meta learning framework to treat the parameters of the GPT-2 model as meta-parameters for the RE model that will be fine-tuned based on the performance of the RE model on a separate meta development set.

To leverage the performance on meta development set to optimize GPT-2 parameters, one solution is to employ reinforcement learning where the rewards for the generated sentences can be directly based on some performance metric (e.g., F1 score). However, due to the small size of the available data, this reward can lead to unstable training with high variance. To remedy this issue, in this work we propose a novel reward function that instead relies on gradients of the RE model's loss to produce more robust training signals. In particular, our intuition is that a generated sample should have a higher reward if the direction in which the RE model should be updated to perform well on the sample and the development data are similar. To fulfil this objective, in the proposed training procedure, after one iteration of training, we first compute the average gradient of the RE model's loss function over the meta development set. Next, the gradient of the loss of the RE model over a generated sample is computed. Finally, the reward for the generated sample is obtained via the cosine similarity between the gradients from the development set and the generated sample. While this reward is backed up with intuitive objectives, we also provide mathematical derivation of the reward based on bi-level optimization to further demonstrate the advantages of our method. Finally, we evaluate the effectiveness of the proposed method on two benchmark datasets for RE. The experiments show the superiority of the proposed model compared to strong baselines.

2 Model

Task Definition: We study the problem of sentence-level relation extraction. In this setting, the objective is to identify semantic relation between two input entity mentions in a sentence. Formally, the input to our model involve a sentence $T = [w_1, w_2, \ldots, w_n]$ and two indices s and o ($1 \leq s, o \leq n$) to indicate the positions of the subject and object in the relation[1]. Our goal is to predict label y representing semantic relation between the entity mentions w_s and w_o from a predefined relation label set $\mathcal{R}$ ($y \in \mathcal{R}$). Note that if the two entity mentions are not involved in a relation the special label $None$ is employed. Also, for convenience, let $\mathcal{O}_{train}$ be the set of available training data for our RE problem (i.e., $T \in \mathcal{O}_{train}$).

Model Overview: In this work we propose a meta-learning framework to train a deep learning model for relation extraction and a generative language model, i.e., GPT-2, to automatically generating training data for the deep learning RE model. In particular, our approach consists of a base model M_θ to be trained on the combination of original manually labeled RE data and automatically generated data. This base model is finally employed at inference time. Also, our approach involves a pre-trained language model M_ψ that will first be trained on the manually labeled data for RE to prepare it for in-domain synthetic data generation. Afterward, the language model will be jointly optimized with the RE model M_θ to leverage the feedback to each other from the models to improve the effectiveness of the generated data for RE. To realize the second objective, we present a reinforcement learning procedure that employs performance of the RE model M_θ as the reward to update the parameters of the generative model M_ψ. More specifically, a reward function based on agreement of the gradients from a development set and generated data is introduced. In the rest of this section, we first describe the details of the proposed approach. We will then present the derivation of the proposed reward function.

[1]Note that semantic relation between two entity mentions can be directed.

2.1 Base Model

In this work we employ a BERT-based model (Devlin et al., 2019) to implement the base model M_θ for RE (θ involves the learnable parameters for the RE model). Concretely, the input sentence T is provided to BERT$_{base}$ in the form of $[\texttt{[CLS]}, w_1, w_2, \ldots, w_n, \texttt{[SEP]}]$. For each word $w_i \in T$, the corresponding hidden vector e_i in the final layer of the BERT model is employed to represent w_i, leading to the sequence of vectors $E = [e_{\texttt{[CLS]}}, e_1, e_2, \ldots, e_n, e_{\texttt{[SEP]}}]$ for T. Note that if w_i contains multiple word-pieces, we utilize the hidden vector for its first word-piece for e_i. Next, to create an overall representation vector h for the input sentence T with input entity mentions w_s and w_o, we employ the Dynamic Pooling mechanism (Chen et al., 2015): $h = [e_{\texttt{[CLS]}} : f(e_1, \ldots, e_{s-1}) : e_s : f(e_{s+1}, \ldots, e_{o-1}) : e_o : f(e_{o+1}, \ldots, e_n)]$, where ":" indicates vector concatenation and "$f(\cdot)$" is the Max Pooling operation over a set of vectors. Finally, the feature vector h is fed into a network architecture to produce a label distribution $P(y'|T, s, o) = \sigma(FF_C(h))$, where σ is the softmax function and FF_C is a two-layer feed-forward network. To train the base model M_θ, we employ the negative log-likelihood function: $\mathcal{L}_C(T, y; \theta) = -\log P(y|T, s, o)$.

2.2 Generating Labeled Data

This section describes our approach to employ the pre-trained language model GPT-2, i.e., M_ψ, to generate synthetic labeled data for RE (ψ contains the learnable parameters for GPT-2). The training of GPT-2 for this purpose is divided into two stages: (1) Pre-training to generate in-domain labeled data for RE and (2) Fine-tuning to improve the effectiveness of the generated data for the RE model.

Pre-Training: To generate additional labeled data in the same domain as existing manually labeled data, we first train the GPT-2 model on the available RE training samples $\mathcal{O}_{train}$. In particular, we augment each training sentence $T \in \mathcal{O}_{train}$ with special tags surrounding the input entity mentions to imply the existence of a relation. Formally, the label-augmented sentence T' for T is prepared as $T' = [w_1, w_2, \ldots, \texttt{<SUB-l>}w_s\texttt{</SUB-l>}, \ldots, \texttt{<OBJ-l>}w_o\texttt{</OBJ-l>}, \ldots, w_n]$, where `l` is `p` for positive samples (i.e., the subject and object entity mentions are in relation); and `n` otherwise. To train the GPT-2 model M_ψ on the label-augmented sentences T', denoted by $T' = [w'_1, w'_2, \ldots, w'_m]$ with m tokens for convenience, we employ auto-regressive training. In particular, the model M_ψ is trained to predict the the next token w'_i using the left context $[w'_1, \ldots, w'_{i-1}]$. Formally, the following loss function is employed to train M_ψ: $\mathcal{L}_G = -\sum_{i=1}^{m} \log P(w'_i|w'_1, \ldots, w'_{i-1})$.

Once pre-trained, the GPT-2 model M_ψ can be used to generate new label-augmented sentences that can be decoded to obtain new sentences along with markers for entity mention positions and relation labels. This newly generated labeled data can then be combined with the original training data $\mathcal{O}_{train}$ to train the base RE model M_θ. It is noteworthy that our label-augmented sentences T' do not encode actual relation labels (i.e., only the information about the positive or negative examples is included) to simplify the generation task for GPT-2. As such, the new synthetic labeled data can only provide a binary label to indicate the existence of relation. Consequently, to employ the generated data to train the RE model M_θ, we integrate a classification head into the RE base model M_θ in which the overall representation vector h is fed into another feed-forward network with one output to serve as a binary classifier to predict positive/negative examples for the synthetic data. Accordingly, the cross-entropy loss for the binary classifier is computed over generated data for training M_θ (i.e., multi-task learning): $\mathcal{L}_B(T, y_b; \theta) = -[y_b * \log(\delta(FF_B(h))) + (1 - y_b)\log(1 - \delta(FF_B(h)))]$ where δ is the sigmoid function, and y_b is 1 for positive samples and 0 otherwise.

Fine-Tuning: The pre-training of GPT-2 model is helpful to generate in-domain labeled data for RE. However, as this pre-training step is done separately from the RE model M_θ, the generated data from GPT-2 might not be optimal for the RE model. For instance, due to the lack of consultancy with M_θ, the generated data can introduce redundant/noisy information to hinder the training of the RE model. As such, it is necessary to allow the RE model to provide feedback for the training of the GPT-2 model so that the generated data from GPT-2 can be directly optimized/customized for our RE model to improve the model performance. To this end, we propose to further fine-tune the GPT-2 model during the training process of the RE model (i.e., joint training) that facilities the exploitation of training guidance from the RE model

Algorithm 1 Training of the ED model and fine-tuning of the GPT-2 model

Input: $\mathcal{O}_{train}, \mathcal{D}_{meta}$
Output: Optimal Models M_ψ and M_θ
Initialize θ_0 and ψ_0
For $t = 1$ **to** *num_train_steps* **do**
  Sample $|\mathcal{B}_\mathcal{O}|$ data points from $\mathcal{O}_{train}$
  Generate $|\mathcal{B}_\mathcal{G}|$ data points (T_g, y_g) using GPT-2 with T'_g as the label-augmented texts
  $\mathcal{B}_\mathcal{C} \leftarrow \mathcal{B}_\mathcal{O} \cup \mathcal{B}_\mathcal{G}$
  ▷ Optimize θ
  $g_\theta \leftarrow \frac{1}{|\mathcal{B}_\mathcal{C}|}\Sigma_{(T,y)\in\mathcal{B}_\mathcal{C}} \nabla_\theta \mathcal{L}_{base}(T, y; \theta_{t-1})$
  $\theta_t \leftarrow \text{GradientUpdate}(\theta_{t-1}, g_\theta)$
  ▷ Evaluate M_θ on $\mathcal{D}_{meta}$
  $d_\theta \leftarrow \frac{1}{|\mathcal{D}_{meta}|}\Sigma_{(T,y)\in\mathcal{D}_{meta}} \nabla_\theta \mathcal{L}_{base}(T, y; \theta_t)$
  ▷ Optimize ψ
  $r_g \leftarrow d_\theta^\top \cdot \nabla_\theta \mathcal{L}_{base}(T_g, y_g; \theta_{t-1})$
  $g_\psi \leftarrow \frac{1}{|\mathcal{B}_\mathcal{G}|}\Sigma_{g=1}^{|\mathcal{B}_\mathcal{G}|} r_g \cdot \nabla_\psi \log P(T'_g; \psi_{t-1})$
  $\psi_t \leftarrow \text{GradientUpdate}(\psi_{t-1}, g_\psi)$
end

to improve the data generation process in GPT-2.

In particular, we present a meta-learning framework for joint training of the GPT-2 and RE model. At each training iteration t, a batch of training examples $\mathcal{B}_{train}$ is sampled from the original training data $\mathcal{O}_{train}$. The GPT-2 model $M_{\psi_{t-1}}$ at the current iteration is then employed to generate a batch of synthetic data $\mathcal{B}_\mathcal{G}$. The combination of the original and generated data batches $\mathcal{B}_\mathcal{C} = \mathcal{B}_{train} \cup \mathcal{B}_\mathcal{G}$ is next leveraged to update the current base RE model $M_{\theta_{t-1}}$ using the loss functions $\mathcal{L}_C$ and $\mathcal{L}_B$. For convenience, we use $\mathcal{L}_{base}$ to refer to both $\mathcal{L}_C$ and $\mathcal{L}_B$. We can decide which loss to use depending on the type of data, i.e., $\mathcal{L}_C$ for original human-labeled data and $\mathcal{L}_B$ for generated labeled data. Afterward, the current GPT-2 model $M_{\psi_{t-1}}$ is updated using the feedback of the base RE model over the effectiveness of the generated samples $\mathcal{B}_\mathcal{G}$ (i.e., leading to M_{ψ_t}. In this way, the GPT-2 model will be adapted along the training process to be generate effective data for the next training iteration of RE model.

To measure the effectiveness of the generated data batch $\mathcal{B}_\mathcal{G}$ for the RE model for GPT-2 updating, one straightforward solution is to employ the performance (e.g., F1 score) of the updated RE model M_{θ_t} over a separate meta development set $\mathcal{D}_{meta}$ as a reward to update the GPT-2 model $M_{\psi_{t-1}}$ with the REINFORCE algorithm (Williams, 1992) (i.e., to account for the discreteness of generated data). However, as we might not have sufficient labeled data to offer a large meta development set, this approach can have high variance for the reward, thus causing unreliable estimation and limiting the effectiveness of generated data for RE (Du et al., 2018). To address this issue, we propose a novel reward to avoid direct reliance on performance metrics and improve the robustness for the meta learning process. Accordingly, we devise the reward function based on the gradient of the training loss $\mathcal{L}_{base}$ for M_{θ_t} over the meta development set $\mathcal{D}_{meta}$, which captures the the direction to cause largest reduction for the loss function (i.e., the steepest direction). Intuitively, a generated sample T_g is helpful for the RE model M_{θ_t} if the gradient of $\mathcal{L}_{base}$ with this sample aligns with the steepest direction with the development data (i.e., similar gradients from T_g and $\mathcal{D}_{meta}$). Formally, our reward to train GPT-2 is obtained via the dot product: $r_g = d_\theta^\top \cdot \nabla_\theta \mathcal{L}_{base}(T_g, y_g; \theta_{t-1})$, where the d_θ is the average of the gradients of the loss function $\mathcal{L}_{base}$ for the RE model on the development set $\mathcal{D}_{meta}$, i.e., $d_\theta = \frac{1}{|\mathcal{D}_{meta}|}\Sigma_{(T,y)\in\mathcal{D}_{meta}} \nabla_\theta \mathcal{L}_{base}(T, y; \theta_t)$. We use θ_t for d_θ to inform the GPT-2 model with the latest RE model to generate better data in the next iteration. Finally, the parameters of the generative model M_ψ is also updated using REINFORCE algorithm in our framework. The details of the proposed procedure are presented in Algorithm 1.

2.3 Derivation of Gradient-based Reward

This section aims to justify the proposed gradient-based reward with a mathematical foundation to better reveal its effectiveness for updating GPT-2 in our framework for RE. For simplicity, we assume that only one example (T_g, y_g) is generated in an iteration, i.e., $|\mathcal{B}_\mathcal{G}| = 1$. Using the reward r_g for (T_g, y_g), we leverage the REINFORCE algorithm to update ψ_t in the last GradientUpdate(ψ_{t-1}, g_ψ) step of Algorithm 1, leading to the update rule:

$$\psi_t \leftarrow \psi_{t-1} + \gamma r_g \cdot \nabla_\psi \log P(T'_g; \psi_{t-1}) \quad (1)$$

where γ is the learning rate. As such, to justify this update rule, we consider a bi-level optimization problem that starts with (T_g, y_g) sampled from $P(T'_g; \psi_{t-1})$, which is the distribution induced by the GPT-2 model $M_{\psi_{t-1}}$. Next, our first level of optimization aims to optimize the loss function $\mathcal{L}_{base}$ for the RE model using (T_g, y_g), leading to the following update rule with gradient descent: $\theta_t = \theta_{t-1} - \gamma \nabla_\theta \mathcal{L}_{base}(T_g, y_g; \theta_{t-1})$.

Here, θ_t can be seen as a function of ψ due to the dependence on (T_g, y_g), which is in turn computed over ψ_{t-1} (i.e., $\theta_t(\psi)$). For convenience, we also compute the expectation over generated samples for ψ_t, i.e., $\bar{\theta}_t = \mathbb{E}_{T'_g \sim P(T'_g;\psi_{t-1})}[\theta_t] = \theta_{t-1} - \gamma \mathbb{E}_{T'_g \sim P(T'_g;\psi_{t-1})}[\nabla_\theta \mathcal{L}_{base}(T_g, y_g; \theta_{t-1})]$.

Afterward, we estimate the loss function $\mathcal{L}_{base}$ of the new RE model θ_t over the meta development set $\mathcal{D}_{meta}$: $J(\theta_t(\psi), \mathcal{D}_{meta}) = \frac{1}{|\mathcal{D}_{meta}|} \sum_{(T,y)\in\mathcal{D}_{meta}} \mathcal{L}_{base}(T, y; \theta_t)$, serving as a measure for the effectiveness of the generated sample (T_g, y_g) to provide feedback/training signals for the GPT-2 model. To this end, our second level of optimization is to optimize $J(\theta_t(\psi), \mathcal{D}_{meta})$ with respect to ψ to update the GPT-2 model for the next iteration. Using gradient descent, our optimization procedure thus needs to compute the gradient $\nabla_\psi J(\theta_t(\psi), \mathcal{D})$ that can be computed via the chain rule:

$$
\begin{aligned}
&\nabla_\psi J(\theta_t(\psi), \mathcal{D}) = \\
&\quad \nabla_{\bar{\theta}_t} J(\theta_t(\psi), \mathcal{D})^\top \cdot \nabla_\psi \bar{\theta}_t(\psi) \\
&\quad \approx \nabla_\theta J(\theta_t(\psi), \mathcal{D})^\top \cdot \nabla_\psi \bar{\theta}_t(\psi) \\
&\quad \text{(substitute the formula for } \bar{\theta}_t \text{ above)} \\
&\quad = \nabla_\theta J(\theta_t(\psi), \mathcal{D})^\top \cdot \nabla_\psi (\theta_{t-1} - \\
&\quad \gamma \mathbb{E}_{T'_g \sim P(T'_g;\psi_{t-1})}[\nabla_\theta \mathcal{L}_{base}(T_g, y_g; \theta_{t-1})]) \\
&\quad \text{(assume } \nabla_\psi \theta_{t-1} \approx 0 \text{ with Markov assumption)} \\
&\quad \approx -\gamma \nabla_\theta J(\theta_t(\psi), \mathcal{D})^\top \cdot \\
&\quad \nabla_\psi \mathbb{E}_{T'_g \sim P(T'_g;\psi_{t-1})}[\nabla_\theta \mathcal{L}_{base}(T_g, y_g; \theta_{t-1})] \\
&\quad \text{(using the log-gradient trick)} \\
&= -\gamma \mathbb{E}_{T'_g \sim P(T'_g;\psi_{t-1})} \Big[\big(\nabla_\theta J(\theta_t(\psi), \mathcal{D})^\top \\
&\qquad \cdot \nabla_\theta \mathcal{L}_{base}(T_g, y_g; \theta_{t-1}) \big) \cdot \nabla_\psi \log P(T'_g; \psi_{t-1}) \Big]
\end{aligned}
$$

To this end, using one roll-out sample and gradient descent, we can eventually derive the update rule for the GPT-2 parameters ψ in Equation 2.3, thus justifying our gradient-based reward function r_g for REINFORCE to highlight its advantage for labeled data generation for RE.

3 Experiments

3.1 Dataset & Hyper-Parameters

To evaluate the effectiveness of the proposed model, i.e., called Data Generation for Relation Extraction (**DGRE**), we employ two English benchmark datasets for RE, i.e., ACE 2005 (Walker et al., 2006) and SPOUSE (Hancock et al., 2018). For ACE 2005, similar to previous work (Nguyen and Grishman, 2016; Shi et al., 2018; Pouran Ben Veyseh et al., 2020), we use the dataset split and preprocessed by (Yu et al., 2015) for compatible comparison. There are 6 different domains in this dataset setting, i.e., (bc, bn, cts, nw, un, and wl), covering text from news, conversations and web blogs. As such, the union of the domains bn and nw (called news) is used as training data; a half of the documents in bc is reserved for the development data, and the remainder (cts, wl and the other half of bc) serve as the test data. In this way, our data organization presents different domains for the training and test data to focus on cross-domain generalization evaluation of the models (Pouran Ben Veyseh et al., 2020).

In addition, we employ the standard data split for the SPOUSE dataset, involving 22,195 sentences for training data, 2,796 sentences for development data, and 2,697 sentences for test data as done in (Hancock et al., 2018; Pouran Ben Veyseh et al., 2020). Each sentence in SPOUSE[2] contains two marked person names (i.e., the entity mentions) and the goal is to predict whether the two people in the sentence are spouses. For both datasets, we sample 10% of the training data portions to serve as meta development data for our model.

We utilize the development set of ACE 2005 dataset to fine-tune the hyper-parameters for our model. Based on the F1 score on the development set, the following hyper-parameters are selected: 8 for the mini-batch size; 2 layers for the feed-forward networks with 250 hidden dimensions; and $1e$-2 for the learning rate for the GradientUpdate steps in our meta learning framework. Moreover, we use the default hyper-parameter values provided by Huggingface[3] for the pre-training step for the GPT-2 model. Finally, the *num_train_steps* in Algorithm 1 is set to the number of training batches in each dataset.

3.2 Baselines

For experiments on ACE 2005, we compare DGRE with prior models reported on this dataset and also the related data augmentation methods. In particular, we consider the following baselines:

RE Models: (i) Feature based models: These models hand-design linguistic features for RE, i.e., FCM, Hybrid FCM, and LRFCM (Yu et al., 2015; Hendrickx et al., 2010). (ii) Deep learning models: These models employ deep learning architectures for RE, i.e., CNN, Bi-GRU (Nguyen

[2] extracted from news articles

[3] https://huggingface.co/ (Apache License 2.0.)

and Grishman, 2016), CNN+DANN (Fu et al., 2017), GSN (Shi et al., 2018), AGGCN (Attention Guided GCN) (Guo et al., 2019), SACNN (Segment-level Attention-based CNN) (Tran et al., 2019), DRPC (Dependency Relation Prediction and Control model) (Veyseh et al., 2019), EA-BERT (Wang et al., 2019), CEON-LSTM (Pouran Ben Veyseh et al., 2020), MapRE (Dong et al., 2021), and A-GCN (Qin et al., 2021). Note that CEON-LSTM and A-GCN have the best reported performance with different settings over ACE 2005 and SPOUSE.

Data Augmentation Models: These methods employ data augmentation (DA) techniques to address labeled data scarcity for RE or related tasks. In particular, we compare with GradLRE (Hu et al., 2021b) that proposes a Gradient Imitation Reinforcement Learning method to encourage pseudo labeled data to imitate the gradient on labeled data, and MetaSRE (Hu et al., 2021a) that employs pseudo label generation in a self-training procedure. Both methods use existing unlabeled data. In addition, we explore DA methods for IE tasks that exploit GPT-2 for data generation, including Filter-GPT (Anaby-Tavor et al., 2020) that filters the generated data based on confidence scores of a pre-trained RE model before combining them with original data; and Novelty-GPT (Yang et al., 2020a) that computes novelty scores for generated data, in comparison to original training data, to weight the samples in the combined dataset for training.

3.3 Results

The performance for the models on the test set of ACE 2005 is presented at Table 1. This table shows that the proposed method significantly outperforms all the baselines with $p < 0.01$ (except for A-GCN over `cts`)). Specifically, compared to the baselines that employ richer information from the input (e.g., syntactic structures in CEON-LSTM or label semantics in MapRE), the improvement obtained by DRGE is important as it requires only the surface form of the input text. This advantage is helpful in domains and settings that suffer from the lack of rich resources and data. Moreover, compared to the models that employ data augmentation (DA) to address data scarcity, the proposed method achieves significantly better results on all three domains. In particular, compared to "*Filter-GPT*" and "*Novelty-GPT*", which are the most relevant approaches to DRGE, our method can substantially improve the performance by up to 2.6% on the average F1 score. We attribute this improvement to the fact that other DA methods do not interact with the target RE model to guide the labeled data creation for optimal performance. In contrast, our method DRGE embeds the data generation process into the training process for RE to allow direct communication between GPT-2 and the RE model to produce more effective labeled data for the RE models.

In addition, Table 2 reports the performance of the model on test data of the SPOUSE dataset. The table corroborates our findings for the advantages of our labeled data generation method

System	**bc**	**cts**	**wl**	**Avg.**
FCM (2015)	61.90	52.93	50.36	55.06
Hybrid FCM (2015)	63.48	56.12	55.17	58.25
LRFCM (2015)	59.40	-	-	-
CNN (2016)	63.26	55.63	53.91	57.60
Bi-GRU (2016)	63.07	56.47	53.65	57.73
CNN+DANN (2017)	65.16	-	-	-
GSN (2018)	66.38	57.92	56.84	60.38
C-GCN* (2018)	67.02	64.40	58.92	63.44
AGGCN* (2019)	65.29	63.65	60.35	63.09
SACNN* (2019)	68.52	64.21	62.19	64.97
DRPC* (2019)	69.41	65.82	61.65	65.62
EA-BERT* (2019)	69.25	61.70	58.48	63.14
CEON-LSTM* (2020)	71.58	66.92	65.17	67.89
MapRE* (2021)	71.54	69.19	66.13	68.95
A-GCN* (2021)	72.56	70.13	65.07	69.25
GradLRE* (2021b)	71.07	68.92	64.33	68.10
MetaSRE* (2021a)	70.57	69.13	65.22	68.30
Filter-GPT* (2020)	70.77	69.40	64.59	68.25
Novelty-GPT* (2020a)	71.32	68.98	65.33	68.54
DGRE* (ours)	**73.99**	**70.18**	**69.23**	**71.13**

Table 1: F1 scores of the models on the ACE 2005 test set. * designates models that employ BERT.

System	P	R	F1
C-GCN (2018)	71.23	79.59	75.18
AGGCN (2019)	72.45	81.95	76.91
SACNN (2019)	78.89	77.09	77.98
DRPC* (2019)	75.09	83.18	78.93
CEON-LSTM* (2020)	82.33	79.73	81.01
MapRE* (2021)	79.33	81.39	80.35
A-GCN* (2021)	81.40	82.64	82.02
GradLRE* (2021b)	82.77	81.08	81.92
MetaSRE* (2021a)	83.49	77.38	80.32
Filter-GPT* (2020)	80.13	81.84	80.98
Novelty-GPT* (2020a)	82.71	79.80	81.23
DGRE* (ours)	84.15	83.29	**83.72**

Table 2: Model performance on the SPOUSE test set. * designates models that employ BERT.

Model	P	R	F1
DGRE	70.83	72.85	71.83
No GPT-2 Data	69.42	71.05	70.23
Separate Fine-Tuning	70.28	71.51	70.89
Dev Perf. Reward	70.88	69.74	70.31
No Pre-training	70.98	71.42	71.20

Table 3: Performance of models on the development set of ACE 2005

Error	DGRE	No Fine-Tuning
Missing Entity	11%	18%
Wrong Entity	15%	23%
Incorrect Relation	9%	17%
Semantics	11%	14%

Table 4: Frequencies of errors in 100 generated samples by GPT-2 when (1) it is fine-tuned using the proposed reward (i.e., DGRE), or (2) no fine-tuning is employed.

for RE over competitive baselines. Specifically, DRGE is significantly better than all the baselines ($p < 0.01$); the performance improvement over GPT-based baselines is at least 2%, thus suggesting the ability to extend to different datasets and domains for RE of our method.

3.4 Ablation Study

To provide more insight into the performance of DGRE, this section studies the contribution of different components of the model to its final performance. Specifically, we examine the following variants of DGRE: (1) **No GPT-2 Data**: For this variant, we entirely remove the GPT-2 model so that the base RE model is only on original labeled data $\mathcal{O}_{train}$; (2) **Separate Fine-Tuning**: In this baseline, the GPT-2 model is separately fine-tuned on the training set $\mathcal{O}_{train}$ to generate new labeled data, i.e., no information from the RE base model is employed to optimize GPT-2; (3) **Dev Perf. Reward**: To study the importance of the proposed gradient-based reward, we report the performance of the model that replaces the proposed reward in DGRE with direct F1 scores of the RE model on the meta development set (i.e., performance-based reward); and (4) **No Pre-training**: This variant is intended to show the benefit of the initial pre-training step of the GPT-2 model using the original training data $\mathcal{O}_{train}$.

Table 3 shows the performance of the models on the ACE 2005 development data. This table shows that all stages and components in the proposed method are necessary to achieve the best performance for DGRE. In particular, removing GPT-2 hurts the performance the most, demonstrating the importance of augmenting RE models with diverse samples generated by GPT-2. Moreover, replacing the proposed reward with the performance on the meta-development set in REINFORCE algorithm reduces the performance significantly, clearly confirming the advantages of the proposed reward with gradient agreement to train our meta learning framework. Finally, we observe worse performance when the GPT-2 model is optimized separately from the RE model, thus testifying to our proposal of joint training to leverage the interaction between the two models for RE.

3.5 Analysis

Error Analysis: To better understand the effectiveness of the proposed reward to update the parameters of the GPT-2 model for RE, we analyze a sample of generated labeled data from GPT-2. A key insight from our analysis is that the proposed gradient-based reward is able to reduce noises in the generated data from GPT-2, thus better supporting the training of the base model for RE. In particular, we compare the frequencies of errors in the generated samples in two scenarios: (1) GPT-2 is fine-tuned by the proposed reward (i.e., DGRE), and (2) No fine-tuning is applied to the pre-trained GPT-2 (i.e., the GPT-2 is only pre-trained separately from the RE model as discussed in Section 2.2). 100 generated examples are reviewed for each scenario in our study. To this end, we consider the following categories of noises in the generated samples by GPT-2 for the RE model: (1) Missing Entity: In the generated texts, there is no tags for entity mentions, or only the subject or the object mention exists; (2) Wrong Entity: The special tokens "<SUB-1>", "</SUB-1>", "<OBJ-1>", or "<OBJ-1>" do not match or surround correct entity mentions in the generated text; (3) Incorrect Relation: GPT-2 generates samples with correct tags for entity mention spans; however, the relation labels are incorrect (e.g., using the negative tags <SUB-n> and <OBJ-n> for samples with relation and vice versa); (4) Semantics: The semantics of the generated text is not sound (e.g., inconsistent topics, repeated words, etc.).

Table 4 shows the frequency of each noise category in the study. As can be seen, fine-tuning the GPT-2 model using the proposed gradient-based

ID	Sentence
1	The soldiers will destroy all <SUB-p> **cities** </SUB-p> on the <OBJ-p> **earth** </OBJ-p> if they can reach to that point.
2	She mourned <OBJ-p> **her** </OBJ-p> <SUB-p> **son** </SUB-p> for a year.
3	"<SUB-p> **United States** </SUB-p> is closely watching this conflict and is prepared for that", the <OBJ-p> **president** <OBJ-p> said.
4	After <SUB-n> **his** <SUB-n> visit, <OBJ-n>**Arab troops** <OBJ-n> started invading the country.
5	<SUB-n> **Maria** <SUB-n> was informed by the police department that the <OBJ-n> **murderer** </OBJ-n> is released.
6	<OBJ-n> **He** <OBJ-n> must be an idiot to return to his house after that <SUB-n> **accident** <SUB-n>.

Table 5: Sample sentences generated by GPT-2 fine-tuned with DGRE. Tags with p indicates positive samples while negative samples involve tags with n.

reward for RE significantly reduces error rates in all categories. Interestingly, the RE-related errors, i.e., Wrong Entity, Missing Entity and Incorrect Relation, enjoy larger error reduction. This fact corroborates the necessity of integrating the fine-tuning process of the GPT-2 model with the training for the RE model. Moreover, the table shows that among all error categories, Wrong Entity is the major source of noises in the generated samples from GPT-2. Future work can thus explore approaches to integrate entity knowledge into the GPT-2 model to address this major for RE.

Case Study: Finally, to shed more light on the quality of the generated text, we present three positive and three negative samples produced by the GPT-2 model fine-tuned in the final epoch of the proposed training procedure for RE on ACE 2005. The sentences are shown in Table 5, highlighting the diverse nature of the generated samples (e.g, different distances and orders between the subject and object mentions) from GPT-2 for RE models.

4 Related Work

Relation Extraction is one of the fundamental tasks in Information Extraction. Due to its importance, various methods have been proposed for RE, ranging from feature-based and kernel-based techniques (Zelenko et al., 2003; Zhou et al., 2005; Bunescu and Mooney, 2005; Sun et al., 2011; Chan and Roth, 2010; Nguyen and Grishman, 2014; Nguyen et al., 2015c) to recent advanced deep learning models (Zeng et al., 2014; dos Santos et al., 2015; Zhou et al., 2016; Verga et al., 2018; Veyseh et al., 2019). The typical neural architectures for RE include Convolutional Neural Networks (Zeng et al., 2014; Nguyen and Grishman, 2015a; dos Santos et al., 2015; Wang et al., 2016), Recurrent Neural Networks (Nguyen and Grishman, 2016; Zhou et al., 2016; Zhang et al., 2017), and self-attentions in Transformer (Verga et al., 2018).

To address the key challenge of data scarcity for RE, prior work has resorted to distantly supervised methods (Mintz et al., 2009; Zeng et al., 2015; Ji et al., 2017; Chen et al., 2021) or pseudo labeling techniques (Hu et al., 2021b,a). However, such methods suffer from low quality of obtained training data, thus hindering performance for RE. Also, we note that data augmentation based on GPT-2 has also been explored for other tasks, such as event extraction (Pouran Ben Veyseh et al., 2021; Papanikolaou and Pierleoni, 2020; Zhang et al., 2020; Yang et al., 2020b; Madaan et al., 2020). Compared to such prior work, our work features a new meta learning framework to jointly train GPT-2 with the downstream RE model, leveraging gradient agreement-based reward to improve the quality of generated labeled data.

5 Conclusion

We present a novel data augmentation method for RE using the pre-trained language model GPT-2. The language model is fine-tuned over label-augmented texts to generate in-domain and labeled samples for RE. To improve the quality of generated data for RE, the GPT-2 model is further optimized along the training process of a RE model in a novel meta learning framework (i.e., joint training to promote model interaction). Agreement scores between gradients of the RE loss function over generated data and a meta development set are proposed as the reward to update the GPT-2 model. We conduct extensive experiments on two benchmark datasets to demonstrate the benefits of the proposed method for RE. In the future, we will explore the application of the proposed methods to other related tasks in Information Extraction.

Limitations & Risks

Limitations: In this work we present a novel method to address data scarcity issue for Relation Extraction (RE). Although our experiments demonstrate the effectiveness of the proposed method, there are still some limitations that can be improved in future work. First, similar to previous work (dos Santos et al., 2015; Veyseh et al., 2019), the current method assumes golden entity mentions to perform RE that might not be the case in different applications. It is thus helpful to explore the method in a more realistic setting where entity mentions are predicted, e.g., using joint inference models to simultaneously extract entity mentions and relations in an end-to-end fashion. Second, our method is currently evaluated only for sentence-level RE (i.e., entity mentions are in the same sentences). Future work can further explore our method for document-level RE to allow entity mentions to appear in different sentences to better demonstrate its advantage. Finally, our method requires the generative GPT-2 model for data generation. To perform well, GPT-2 needs to be trained on large unlabeled datasets that might not be readily available for low-resource languages. As such, it is important to further evaluate our method on low-resource languages to better reveal its effectiveness.

Risks: In this work, we employ GPT-2 to generate new training samples for the task of RE. Although GPT-2 is publicly available and the datasets employed in this work to fine-tune GPT-2 for RE are also publicly available, a generative language model might produce biased sentences, insulting texts or reveal private information. As such, it is necessary to take further measures before publicly releasing the automatically generated labeled data. To this end, we inspect the data employed for fine-tuning to exclude any offensive text and identity information. The generated data will also be inspected for purpose before publicly releasing the data.

Acknowledgement

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112, the NSF grant CNS-1747798 to the IUCRC Center for Big Learning, and the NSF grant # 2239570. This research is also supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract 2022-22072200003. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *EMNLP*.

Yee S. Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *COLING*.

Tiantian Chen, Nianbin Wang, Hongbin Wang, and Haomin Zhan. 2021. Distant supervision for relation extraction with sentence selection and interaction representation. In *Wireless Communications and Mobile Computing*. Hindawi.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling 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 167–176, Beijing, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Manqing Dong, Chunguang Pan, and Zhipeng Luo. 2021. MapRE: An effective semantic mapping approach for low-resource relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *ACL*.

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. 2018. Adapting auxiliary losses using gradient similarity. In *arXiv preprint arXiv:1812.02224*.

Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. 2017. Domain adaptation for relation extraction with domain adversarial neural network. In *IJCNLP*.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *ACL*.

Braden Hancock, Martin Bringmann, Paroma Varma, Percy Liang, Stephanie Wang, and Christopher Ré. 2018. Training classifiers with natural language explanations. In *ACL*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of SEW-2009*.

Xuming Hu, Chenwei Zhang, Fukun Ma, Chenyao Liu, Lijie Wen, and Philip S. Yu. 2021a. Semi-supervised relation extraction via incremental meta self-training. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.

Xuming Hu, Chenwei Zhang, Yawen Yang, Xiaohe Li, Li Lin, Lijie Wen, and Philip S. Yu. 2021b. Gradient imitation reinforcement learning for low resource relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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 AAAI Conference on Artificial Intelligence*.

Aman Madaan, Dheeraj Rajagopal, Yiming Yang, Abhilasha Ravichander, Eduard Hovy, and Shrimai Prabhumoye. 2020. Eigen: Event influence generation using pre-trained language models. In *arXiv preprint arXiv:2010.11764*.

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*, pages 1003–1011.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 27–38, Online. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Learning cross-task dependencies for joint extraction of entities, events, event arguments, and relations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9349–9360, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *ACL*.

Thien Huu Nguyen and Ralph Grishman. 2015a. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st NAACL Workshop on Vector Space Modeling for NLP (VSM)*.

Thien Huu Nguyen and Ralph Grishman. 2016. Combining neural networks and log-linear models to improve relation extraction. *Proceedings of IJCAI Workshop on Deep Learning for Artificial Intelligence*.

Thien Huu Nguyen, Barbara Plank, and Ralph Grishman. 2015c. Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction. In *ACL-IJCNLP*.

Tuan Ngo Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. 2019a. On the effectiveness of the pooling methods for biomedical relation extraction with deep learning. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*.

Jian Ni, Taesun Moon, Parul Awasthy, and Radu Florian. 2020. Cross-lingual relation extraction with transformers. *arXiv preprint arXiv:2010.08652*.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. In *SciNLP workshop at AKBC*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Exploiting the syntax-model consistency for neural relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Unleash GPT-2 power for event detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.

Han Qin, Yuanhe Tian, and Yan Song. 2021. Relation extraction with word graphs from n-grams. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ge Shi, Chong Feng, Lifu Huang, Boliang Zhang, Heng Ji, Lejian Liao, and Heyan Huang. 2018. Genre separation network with adversarial training for cross-genre relation extraction. In *EMNLP*.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *ACL*.

Van-Hien Tran, Van-Thuy Phi, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Relation classification using segment-level attention-based cnn and dependency-based rnn. In *NAACL-HLT*.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *EMNLP*.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019. Improving cross-domain performance for relation extraction via dependency prediction and information flow control. In *IJCAI*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. In *Technical report, Linguistic Data Consortium*.

Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. 2019. Extracting multiple-relations in one-pass with pre-trained transformers. In *ACL*.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *EMNLP*.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Kluwer Academic*.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020a. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. 2020b. Generative data augmentation for commonsense reasoning. In *Findings of EMNLP 2020*.

Mo Yu, Matthew R Gormley, and Mark Dredze. 2015. Combining word embeddings and feature embeddings for fine-grained relation extraction. In *NAACL-HLT*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3:1083–1106.

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, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING*.

Danqing Zhang, Tao Li, Haiyang Zhang, and Bing Yin. 2020. On data augmentation for extreme multi-label classification. In *arXiv preprint arXiv:2009.10778*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *ACL*.

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 *ACL*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations & Risks

☑ A2. Did you discuss any potential risks of your work?
Limitations & Risks

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

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

B ☑ Did you use or create scientific artifacts?

2.2

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

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The license information is publicly available

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

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Limitations and Risks

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The information is publicly available.

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

C ☑ Did you run computational experiments?

Experiments

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

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

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

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

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

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

Left blank.

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

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

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

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

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