GDA: Generative Data Augmentation Techniques for Relation Extraction Tasks

Xuming Hu[1*], **Aiwei Liu**[1*], **Zeqi Tan**[2], **Xin Zhang**[3], **Chenwei Zhang**[4†], **Irwin King**[5], **Philip S. Yu**[1,6]

[1]Tsinghua University, [2]Zhejiang University, [3]Harbin Institute of Technology (Shenzhen), [4]Amazon, [5]The Chinese University of Hong Kong, [6]University of Illinois at Chicago

[1]{hxm19,liuaw20}@mails.tsinghua.edu.cn

[4]cwzhang@amazon.com

Abstract

Relation extraction (RE) tasks show promising performance in extracting relations from two entities mentioned in sentences, given sufficient annotations available during training. Such annotations would be labor-intensive to obtain in practice. Existing work adopts data augmentation techniques to generate pseudo-annotated sentences beyond limited annotations. These techniques neither preserve the semantic consistency of the original sentences when rule-based augmentations are adopted, nor preserve the syntax structure of sentences when expressing relations using seq2seq models, resulting in less diverse augmentations. In this work, we propose a dedicated augmentation technique for relational texts, named GDA, which uses two complementary modules to preserve both semantic consistency and syntax structures. We adopt a generative formulation and design a multi-tasking solution to achieve synergies. Furthermore, GDA adopts entity hints as the prior knowledge of the generative model to augment diverse sentences. Experimental results in three datasets under a low-resource setting showed that GDA could bring *2.0%* F1 improvements compared with no augmentation technique. Source code and data are available[1].

1 Introduction

Relation Extraction (RE) aims to extract semantic relations between two entities mentioned in sentences and transform massive corpora into triplets in the form of (subject, relation, object). Neural relation extraction models show promising performance when high-quality annotated data is available (Zeng et al., 2017; Zhang et al., 2017; Peng et al., 2020). While in practice, human annotations would be labor-intensive and time-consuming to obtain and hard to scale up to a large number of relations (Hu et al., 2020, 2021a,b; Liu et al., 2022b). This motivates us to solicit data augmentation techniques to generate pseudo annotations.

A classical effort devoted to data augmentation in NLP is adopting rule-based techniques, such as synonym replacement (Zhang et al., 2015; Cai et al., 2020), random deletion (Kobayashi, 2018; Wei and Zou, 2019), random swap (Min et al., 2020) and dependency tree morphing (Şahin and Steedman, 2018). However, these methods generate synthetic sentences without considering their semantic consistencies with the original sentence, and may twist semantics due to the neglection of syntactic structures. Other successful attempts on keeping the semantic consistency of the sentences are model-based techniques. The popular back translation method (Dong et al., 2017; Yu et al., 2018) generates synthetic parallel sentences using a translation model to translate monolingual sentences from the target language to the source language. However, it works exclusively on sentence-level tasks like text classification and translation, which is not designed to handle fine-grained semantics in entity-level tasks like relation extraction. Bayer et al. (2022) design a specific method for RE tasks by fine-tuning GPT-2 to generate sentences for specific relation types. However, it cannot be used in practice because the model generates less diverse sentences – it includes similar entities and identical relational expressions under the same relation.

To keep the generated sentences diverse while semantically consistent with original sentences, we propose a relational text augmentation technique named GDA. As illustrated in Figure 1, we adopt the multi-task learning framework with one shared encoder and two decoders that are complementary with each other: One decoder aims to predict the original sentence by restructuring words in the syntactic structure, which can maintain the semantics of the original sentence and ensure the model has the ability to generate semantically consistent target sentence. However, restructuring the syntactic

[1]https://github.com/THU-BPM/GDA

*Equally Contributed.

†Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 10221–10234

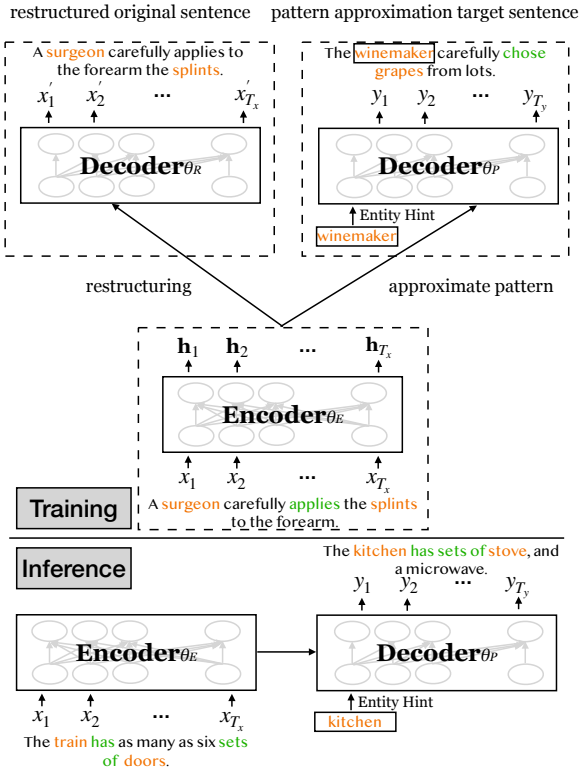


Figure 1: Overview of the proposed relational text augmentation technique with pattern approximation: GDA. We highlight the entities and pattern in the sentences. We define the pattern as the dependency parsing path between two entities.

structure of the original sentence inevitably breaks the coherence. Therefore, another decoder preserves and approximates the syntax patterns of the original sentence by generating the target sentence with a similar syntax structure drawn from the existing data. This decoder can not only keep the target sentences coherent but more importantly, ensure that the model could maintain the original syntax pattern when generating pseudo sentences. Therefore, different patterns under the same relation can be preserved, instead of predicting the same syntax pattern due to relational inductive biases (Sun et al., 2021), thereby increasing the diversity of augmented sentences. We further adopt an entity in the target sentence as a hint to the input of that decoder, which can serve as prior knowledge to control the content of generated sentences. During inference, we could generate diverse sentences by taking a variety of different entity hints and origin sentences with various syntax patterns as input. To summarize, the main contributions of this work are as follows:

- We study the task that focuses on the synergy between syntax and semantic preserving during data augmentation and propose a relational text augmentation technique GDA.
- We adopt GDA which leverages the multi-task learning framework to generate semantically consistent, coherent, and diverse augmented sentences for RE task. Furthermore, entity hints from target sentences are served to guide the generation of diverse sentences.
- We validate the effectiveness of GDA on three public RE datasets and low-resource RE settings compared to other competitive baselines.

2 Related Work

Data augmentation techniques have been widely used to improve the performance of models in the NLP tasks. The existing methods could be divided mainly into three categories: Rule-based techniques, Example interpolation techniques, and Model-based techniques (Feng et al., 2021).

Rule-Based Techniques Rule-based techniques adopt simple transform methods. Wei and Zou (2019) proposes to manipulate some words in the original sentences such as random swap, insertion, and deletion. Şahin and Steedman (2018) proposes to swap or delete children of the same parent in the dependency tree, which could benefit the original sentence with case marking. Chen et al. (2020a) constructs a graph from the original sentence pair labels and augment sentences by directly inferring labels with the transitivity property.

Example Interpolation Techniques Example interpolation techniques such as MixText (Chen et al., 2020b), Ex2 (Lee et al., 2021), and BackMix (Jin et al., 2022) aim to interpolate the embeddings and labels of two or more sentences. Guo et al. (2020) proposes SeqMix for sequence translation tasks in two forms: the hard selection method picks one of the two sequences at each binary mask position, while the soft selection method softly interpolates candidate sequences with a coefficient. Soft selection method also connects to existing techniques such as SwitchOut (Wang et al., 2018) and word dropout (Sennrich et al., 2016).

Model-Based Techniques Model-based techniques such as back translation (Sennrich et al., 2016), which could be used to train a question answering model (Yu et al., 2018) or transfer sequences from a high-resource language to a low-resource language (Xia et al., 2019). Hou et al. (2018) introduce a sequence to sequence model to generate diversely augmented data, which could

Methods		Characteristics		
		Seman.	Coher.	Diver.
Rule Based	Wei and Zou (2019)	✓	–	✓
	Chen et al. (2020a)	✓	–	✓
Example Interpolation	Chen et al. (2020b)	✓	–	–
	Lee et al. (2021)	✓	–	–
	Jin et al. (2022)	✓	–	–
Model Based	Gao et al. (2019)	✓	–	–
	Anaby-Tavor et al. (2020)	✓	✓	–
	Papanikolaou and Pierleoni (2020)	✓	✓	–
	Bayer et al. (2022)	✓	✓	–
	GDA (Ours)	✓	✓	✓

Table 1: Characteristics comparison between different categories of techniques. "Seman." means "semantic consistency", "Coher." means "coherence", and "Diver." means "diversity".

improve the dialogue language understanding task. Kobayashi (2018); Gao et al. (2019) propose the contextualized word replacement method to augment sentences. Anaby-Tavor et al. (2020); Li et al. (2022a); Bayer et al. (2022) adopt language model which is conditioned on sentence-level tags to modify original sentences exclusively for classification tasks. Some techniques try to combine some simple data augmentation methods (Ratner et al., 2017; Ren et al., 2021) or add human-in-the-loop (Kaushik et al., 2019, 2020). Other paraphrasing techniques (Kumar et al., 2020; Huang and Chang, 2021; Gangal et al., 2021) and rationale thinking methods (Hu et al., 2023) also show the effectiveness of data augmentation methods.

Characteristics Comparsion We compare our GDA with other data augmentation techniques from the characteristics of semantic consistency, coherence, and diversity in Table 1. Note that the example interpolation techniques do not generate specific sentences, and only operates at the semantic embedding level. Therefore, we believe that they can only maintain semantic consistency. Compared with other SOTA data augmentation techniques, GDA uses a multi-task learning framework, which leverages two complementary seq2seq models to make the augmented sentences have semantic consistency, coherence, and diversity simultaneously.

3 Proposed data augmentation technique

The proposed data augmentation technique GDA consists of two steps: 1) Train a seq2seq generator. 2) Generate pseudo sentences. As illustrated in Figure 1, the first step adopts T5 (Raffel et al., 2020) consisting of encoder and decoder parts as the seq2seq generator (θ). The generator learns to convert two sentences with the same relation label. Specifically, the encoder part takes a sentence $X = (x_1, x_2, ..., x_{T_x})$ as input where named entities are recognized and marked in advance, and obtains the contextualized token embeddings $H = (\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{T_x})$. The decoder part takes the H as input and generates target sentence $Y = (y_1, y_2, ..., y_{T_y})$ word by word by maximizing the conditional probability distribution of $p(y_i|y_{<i}, H, \theta)$. The second step randomly selects an annotated sentence as input, and leverages the trained generator to generate pseudo sentence with entity marker and same relation label. Now, we introduce the details of each step.

3.1 Train a seq2seq generator

Training a seq2seq generator aims to obtain a generator that could augment annotated sentences to diverse, semantically consistent, and coherent pseudo sentences. In addition, the entities in the augmented pseudo sentences also need to be marked for entity-level relation extraction task. To achieve this goal, the generator must convert two sentences with the same relation label and emphasize contextualized relational signals at the entity level during the generation process. In practice, for each annotated sentence $X = (x_1, x_2, ..., x_{T_x})$, we adopt the labeling scheme introduced in Soares et al. (2019), and augment X with four reserved tokens: $[E_{sub}]$, $[/E_{sub}]$, $[E_{obj}]$, $[/E_{obj}]$ to represent the start and end position of subject and object named entities respectively, and inject them to X. For example, "A $[E_{sub}]$ surgeon $[/E_{sub}]$ carefully applies the $[E_{obj}]$ splints $[/E_{obj}]$ to the forearm.". Then we feed the updated X into the T5 encoder part to obtain contextualized token embeddings H: $H = \text{Encoder}(X)$.

A natural paradigm for the decoder part to generate the target sentence is to select another sentence in the training set that has the same relation as the input sentence. Bayer et al. (2022) fine-tuned GPT-2 to generate sentences for specific relation types. However, it requires too much computational cost to train multiple GPT-2s for each relation type, and we observed no promising results are obtained. We attribute the reason to two aspects: 1) the diversity of generated sentences is not emphasized, resulting in the generation of sentences with similar patterns, and 2) the entity-level relation extraction task is not considered, resulting in missing entity information.

In this paper, we propose to leverage the multi-task learning framework to address the above short-

comings, which performs two tasks: original sentence restructuring and original sentence pattern approximation. In practice, our framework consists of two seq2seq models that share the same encoder part, but employ two decoder parts to complete the two tasks, respectively.

Original sentence restructuring. The original sentence restructuring task aims to improve the ability of the model to generate semantically consistent sentences. As illustrated in Figure 1, the target generated sentence is just the restructured original sentence $X^{'} = (x_1^{'}, x_2^{'}, ..., x_{T_x}^{'})$ that has the same length and words as the original sentence. We adopt the pre-ordering rules proposed by Wang et al. (2007) in machine translation. These rules could modify the syntactic parse tree obtained from the original sentence and permutate the words by modifying the parsed tree. The target sentence is closer to the expression order of words without changing the semantics of the original sentence Furthermore, since the entities are not changed, it is easy to mark the position of the entities, e.g.:

> Original: A [*Esub*] surgeon [*/Esub*] carefully applies the [*Eobj*] splints [*/Eobj*] to the forearm.
> Target: A [*Esub*] surgeon [*/Esub*] carefully applies to the forearm [*Eobj*] splints [*/Eobj*].

In this way, the decoder network is employed to predict the restructured original sentence by maximizing $p(X^{'}|H, \theta_R)$:

$$\mathcal{L}_{\theta_R} = \sum_{m=1}^{M} \sum_{i=1}^{T_x} \log p\left(X_i^{'(m)} \mid X_{<i}^{'(m)}, H^{(m)}, \theta_R\right), \quad (1)$$

where θ_R denotes the parameters of the decoder part for original sentence restructuring. M is the number of the training data.

Original sentence pattern approximation. The restructured sentence inevitably breaks the coherence of the original sentence. Therefore, we employ another seq2seq model to auxiliary predict unmodified sentences. When we directly adopt seq2seq model for sentence generation (Bayer et al., 2022), The seq2seq model usually generates stereotyped sentences with the same pattern (Battaglia et al., 2018). For example, for relation `Component-Whole`, using generative methods such as T5 will always tend to generate pattern "*consist of*" with high frequency, rather than rare "*is opened by*" and "*has sets of*", which will limit the performance improvement of augmented sentences. In this paper, we introduce the original sentence pattern approximation task to force the original and target sentences to have approximate patterns, so that the augmented sentences could maintain the pattern of the original sentences and enhance the sentence diversity. In practice, we define the **pattern** as the dependency parsing path between two entities. We assume this parsing path is sufficient to express relational signals (Peng et al., 2020).

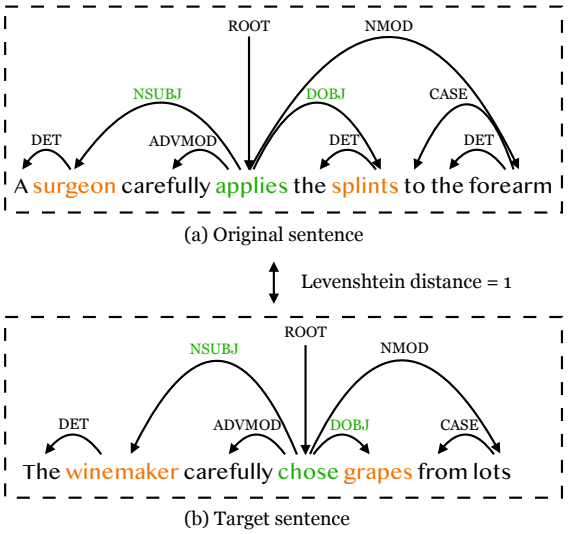


Figure 2: Overview of the original sentence approximation task. We highlight the entities and pattern in the original and target sentence.

As illustrated in Figure 2, we first parse the original sentence (Chen and Manning, 2014) and obtain the path "NSUBJ-applies-DOBJ" between two entities "surgeon" and "splints" as pattern. To force the generative model to learn this pattern, we employ the Levenshtein (Lev) distance (Yujian and Bo, 2007) to find the target pattern that is closest to the original pattern and has the same relation with the original sentence, then the corresponding sentence will be chosen as the output during training. The Lev distance is a measure of the surface form similarity between two sequences, which is defined as the minimum number of operations (inserting, deleting, and replacing) required to convert the original sequence to the target sequence. We give a pseudo code implementation in the Appendix A.

For example, in Figure 2, the Lev distance between the pattern "NSUBJ-applies-DOBJ" and the pattern "NSUBJ-chosen-DOBJ" is 1, so the target output sentence is:

> Original: A [*Esub*] surgeon [*/Esub*] carefully ***applies*** the [*Eobj*] splints [*/Eobj*] to the forearm.
> Target: The [*Esub*] winemaker [*/Esub*] carefully ***chose*** [*Eobj*] grapes [*/Eobj*] from lots.

In practice, we choose the target sentence with pattern less than λ (e.g. $\lambda = 3$) from the original

sentence's pattern, where λ is a hyperparameter.

In this way, the decoder network is employed to predict the pattern approximation target sentence $Y = (y_1, y_2, ..., y_{T_y})$ by maximizing $p(Y|H, \theta_P)$:

$$\mathcal{L}_{\theta_P} = \sum_{n=1}^{N} \sum_{i=1}^{T_y} \log p \left(y_i^{(n)} \mid y_{<i}^{(n)}, H^{(n)}, \theta_P \right), \quad (2)$$

where θ_P denotes the parameters of the decoder part for the original sentence pattern approximation. N is the number of sentences for all outputs that satisfy the Lev distance less than 3.

Entity-level sentence generation. Furthermore, to generate more controllable entity-level sentences and help the generator to better mark entities in the augmented sentences, we add a subject or object **Entity** (E) from the target output sentence to the input embedding of the decoder as a hint.

For example, in Figure 1, we add **winemaker** or **grapes** to the input of the decoder part θ_P, which helps derive entity-oriented controllable sentence and increase the diversity of generated sentences by adopting different entity hints. Therefore, we finalize the loss function of Eq. 2 as:

$$\mathcal{L}_{\theta_P} = \sum_{n=1}^{N} \sum_{i=1}^{T_y} \log p \left(y_i^{(n)} \mid y_{<i}^{(n)}, E^{(n)}, H^{(n)}, \theta_P \right). \quad (3)$$

The overall loss function of multi-task learning is the sum of log probabilities of original sentence restructuring and pattern approximation tasks:

$$\begin{aligned} \mathcal{L}_{\theta} = & \sum_{n=1}^{N} \sum_{i=1}^{T_y} \log p \left(Y_i^{(n)} \mid Y_{<i}^{(n)}, E^{(n)}, H^{(n)}, \theta \right) \\ & + \sum_{m=1}^{M} \sum_{i=1}^{T_x} \log p \left(X_i^{'(m)} \mid X_{<i}^{'(m)}, H^{(m)}, \theta \right), \end{aligned} \quad (4)$$

where $\theta = (\theta_E, \theta_R, \theta_P)$. θ_E is the parameter of encoder part. In practice, we adopt an iterative strategy to train two complementary tasks. For each iteration, we first optimize the (θ_E, θ_R) framework in the restructuring task for five epochs. The optimized θ_E will be employed as the initial θ_E of the pattern approximation task. Next, we optimize the (θ_E, θ_P) framework for five epochs in the pattern approximation task, and the updated θ_E will be used in the next iteration. Finally, θ_E and θ_P will be adopted to generate augmented sentences.

3.2 Generate pseudo sentences

After we obtain the trained seq2seq generator T5 (θ_E, θ_P), which focuses on the reconstruction of diverse, semantically consistent, and coherent relational signals. We leverage the generator to generate entity-oriented pseudo sentences as augmented data. In practice, we randomly select an annotated sentence X and one marked subject or object entity E under the relation label to which the X belongs from the annotated data. Then we obtain the augmented sentence by $(X, E, \theta_E, \theta_P)$, where subject and object entities (one of them is E) have been marked during the generation process.

The augmented sentences have the same relation label as the original sentences and have enough diversity with different sentences and entity hints randomly sampled from the annotated data.

4 Experiments

We conduct extensive experiments on three public datasets and low-resource RE setting to show the effectiveness of GDA and give a detailed analysis to show its advantages.

4.1 Base Models and Baseline Introduction

We adopt two SOTA base models:

(1) **SURE** (Lu et al., 2022) creates ways for converting sentences and relations that effectively fill the gap between the formulation of summarization and RE tasks.

(2) **RE-DMP** (Tian et al., 2022) leverages syntactic information to improve relation extraction by training a syntax-induced encoder on auto-parsed data through dependency masking.

We adopt three types of baseline models.

(1) **Rule-Based Techniques**: **EDA** (Wei and Zou, 2019) adopts synonym replacement, random insertion, random swap, and random deletion to augment the original sentences. **Paraphrase Graph** (Chen et al., 2020a) constructs a graph from the annotated sentences and creates augmented sentences by inferring labels from the original sentences using a transitivity property.

(2) **Example Interpolation Techniques**: Inspired by Mixup (Zhang et al., 2018), **MixText** (Chen et al., 2020b) and **Ex2** (Lee et al., 2021) aim to interpolate the embeddings and labels of two or more sentences. **BackMix** (Jin et al., 2022) proposes a back-translation based method which softly mixes the multilingual augmented samples.

(3) **Model-Based Techniques**: **Soft DA** (Gao et al., 2019) replaces the one-hot representation of a word by a distribution over the vocabulary and calculates it based on contextual information.

Methods / Datasets	PLMs	Para.	SemEval				TACRED				TACREV				AVG.	Δ
			10%	25%	50%	100%	10%	25%	50%	100%	10%	25%	50%	100%		
Base (SURE)†	BART-Large	406M	77.2	81.5	83.9	86.3	67.9	70.4	71.9	73.3	72.3	75.1	77.4	79.2	76.4	–
+EDA†	–	–	77.9	82.0	84.4	86.7	68.6	71.0	72.2	73.8	72.7	76.0	77.9	79.6	76.9	0.5 ↑
+Paraphrase Graph†	–	–	77.8	81.8	84.4	86.6	68.4	70.9	72.3	73.7	72.9	75.7	77.7	79.4	76.8	0.4 ↑
+MixText†	BERT-Base	110M	78.6	82.6	85.0	87.2	69.0	71.7	72.9	74.1	73.6	76.4	78.4	80.0	77.5	1.1 ↑
+Ex2†	T5-Base	220M	79.1	83.0	85.5	87.5	69.6	72.1	73.2	74.3	74.2	76.7	78.8	80.3	77.8	1.4 ↑
+BackMix‡	mBART-Base	140M	78.7	82.5	85.2	87.2	69.2	71.8	72.8	74.0	73.9	76.3	78.2	80.0	77.5	1.1 ↑
+Soft DA†	BERT-Base	110M	78.5	82.4	85.1	87.0	68.9	71.7	72.8	74.0	73.5	76.5	78.4	79.9	77.4	1.0 ↑
+LAMBADA†	GPT-2	117M	78.4	82.4	85.0	87.1	69.1	71.6	72.9	73.8	73.6	76.5	78.3	79.8	77.4	1.0 ↑
+DARE‡	GPT-2	117M	78.7	82.6	85.3	87.2	69.2	71.7	72.9	74.1	73.8	76.5	78.4	80.1	77.5	1.1 ↑
+Text Gen‡	GPT-2-Medium	345M	79.0	83.2	85.7	87.7	69.7	71.9	73.4	74.4	74.2	76.8	78.6	80.4	77.8	1.4 ↑
+GDA (T5-Base)	T5-Base	220M	**79.7**	**83.6**	**85.9**	**88.0**	**70.4**	**72.6**	**73.8**	**74.9**	**74.8**	**77.2**	**79.1**	**80.8**	**78.3**	**1.9** ↑
w/o Approximation	T5-Base	220M	78.8	82.7	85.2	87.4	69.2	71.9	73.0	74.5	74.1	76.9	78.6	80.4	77.7	–
w/o Restructuring	T5-Base	220M	79.1	82.9	85.4	87.5	69.4	71.9	73.2	74.6	74.4	77.0	78.8	80.5	77.9	–
w/o Two Tasks	T5-Base	220M	78.3	82.3	84.7	86.9	68.7	71.2	72.5	74.0	73.2	76.2	78.1	79.6	77.1	–
+GDA (BART-Base)	BART-Base	140M	79.2	83.2	85.6	87.8	69.7	72.3	73.3	74.5	74.4	76.8	78.7	80.5	78.0	1.6 ↑
+GDA (T5-Small)	T5-Small	60M	79.0	82.9	85.4	87.6	69.5	72.1	72.9	74.1	74.1	76.5	78.4	80.2	77.7	1.3 ↑
Base (RE-DMP)†	BERT-Large	340M	76.4	81.1	83.4	85.9	67.3	70.0	71.1	72.4	71.5	74.7	77.0	78.5	75.8	–
+EDA†	–	–	76.9	81.5	83.7	86.1	67.9	70.4	71.5	72.6	72.1	75.1	77.3	78.7	76.2	0.4 ↑
+Paraphrase Graph†	–	–	77.0	81.5	83.8	86.2	68.0	70.4	71.4	72.7	72.1	75.0	77.2	78.7	76.2	0.4 ↑
+MixText†	BERT-Base	110M	77.8	82.0	84.0	86.5	68.6	70.8	71.9	73.1	72.6	75.6	77.7	79.2	76.7	0.9 ↑
+Ex2†	T5-Base	220M	78.3	82.6	84.6	87.0	69.1	71.4	72.6	73.5	73.1	76.2	78.0	79.5	77.2	1.4 ↑
+BackMix‡	mBART-Base	140M	77.9	82.2	84.3	86.6	68.5	71.0	71.9	73.2	73.0	75.7	77.6	79.3	76.8	1.0 ↑
+Soft DA†	BERT-Base	110M	78.0	82.1	83.9	86.4	68.3	70.9	71.8	72.9	72.8	75.6	77.5	79.1	76.6	0.8 ↑
+LAMBADA†	GPT-2	117M	77.7	82.1	83.9	86.4	68.5	70.7	71.7	73.0	72.9	75.5	77.6	79.2	76.8	1.0 ↑
+DARE‡	GPT-2	117M	77.8	82.3	84.2	86.5	68.7	71.0	72.0	73.1	73.1	75.8	77.7	79.2	76.8	1.0 ↑
+Text Gen‡	GPT-2-Medium	345M	78.5	82.6	84.8	87.1	69.0	71.6	72.5	73.6	73.3	76.2	78.1	79.7	77.3	1.5 ↑
+GDA (T5-Base)	T5-Base	220M	**79.0**	**83.1**	**85.4**	**87.8**	**69.7**	**72.2**	**73.1**	**74.1**	**74.0**	**76.7**	**78.2**	**80.1**	**77.9**	**2.1** ↑
w/o Approximation	T5-Base	220M	78.0	82.4	84.6	87.0	68.9	71.4	72.2	73.6	73.2	76.0	78.1	79.4	77.1	–
w/o Restructuring	T5-Base	220M	78.2	82.5	84.8	87.1	69.1	71.5	72.4	73.9	73.5	76.2	78.0	79.7	77.2	–
w/o Two Tasks	T5-Base	220M	77.3	81.8	84.0	86.3	68.3	70.7	71.7	72.9	72.4	75.3	77.5	79.0	76.4	–
+GDA (BART-Base)	BART-Base	140M	78.6	82.5	85.0	87.4	69.1	71.7	72.6	73.4	73.4	76.3	78.2	79.7	77.3	1.5 ↑
+GDA (T5-Small)	T5-Small	60M	78.2	82.0	84.6	86.8	68.5	71.2	72.0	73.0	73.1	75.9	77.8	79.3	76.9	1.1 ↑

Table 2: Average micro F1 results over three runs in three RE datasets. † means we rerun the open source code and adopt the given parameters, ‡ means we product the code with the given parameters. We underline the best results among the baseline models. PLMs and Para. mean the pre-trained models and the corresponding parameters used.

LAMBADA (Anaby-Tavor et al., 2020) and **DARE** (Papanikolaou and Pierleoni, 2020) fine-tune multiple generative models for each relation type to generate augmentations. **Text Gen** (Bayer et al., 2022) proposes a sophisticated generation-based method that generates augmented data by incorporating new linguistic patterns.

4.2 Datasets and Experimental Settings

Three classical datasets are used to evaluate our technique: the SemEval 2010 Task 8 (**SemEval**) (Hendrickx et al., 2010), the TAC Relation Extraction Dataset (**TACRED**) (Zhang et al., 2017), and the revisited TAC Relation Extraction Dataset (**TACREV**) (Alt et al., 2020). SemEval is a classical benchmark dataset for relation extraction task which consists of 19 relation types, with 7199/800/1864 relation mentions in training/validation/test sets, respectively.

TACRED is a large-scale crowd-source relation extraction benchmark dataset which is collected from all the prior TAC KBP shared tasks. TACREV found that the TACRED dataset contains about 6.62% noisily-labeled relation mentions and relabeled the validation and test set. TACRED and TACREV consist of 42 relation types, with 75049/25763/18659 relation mentions in training/-validation/test sets.

We train the T5-base (Raffel et al., 2020) with the initial parameter on the annotated data and utilize the T5 default tokenizer with max-length as 512 to preprocess data. We use AdamW with $5e{-}5$ learning rate to optimize cross-entropy loss. Both GDA and all baseline augmentation methods augment the annotated data by **3x** for fair comparison. For the low-resource RE setting, We randomly sample 10%, 25%, and 50% of the training data and use them for all data augmentation techniques. All augmented techniques can only be trained and augmented on these sampled data.

4.3 Main Results

Table 2 shows the average micro F1 results over three runs in three RE datasets. All base models could gain F1 performance improvements from the augmented data when compared with the models that only adopt original training data, which demonstrates the effectiveness of data augmentation tech-

niques in the RE task. For three RE datasets, Text Gen is considered the previous SOTA data augmentation technique. The proposed GDA technique consistently outperforms all baseline data augmentation techniques in F1 performance (with student's T test $p < 0.05$). More specifically, compared to the previous SOTA: Text Gen, GDA on average achieves 0.5% higher F1 in SemEval, 0.5% higher F1 in TACRED, and 0.4% higher F1 in TACREV across various annotated data and base models.

Considering the low-resource relation extraction setting when annotated data are limited, e.g. 50% for SemEval, TACRED and TACREV, GDA could achieve an average boost of 0.5% F1 compared to Text Gen. When less labeled data is available, 10% for SemEval, TACRED, and TACREV, the average F1 improvement is consistent, and surprisingly increased to 0.8%. We attribute the consistent improvement of GDA to the diverse and semantically consistent generated sentences that are exploited: we bootstrap the relational signals of the augmented data via multi-task learning, which could help generate entity-oriented sentences for relation extraction tasks.

To demonstrate the impact of different pre-trained language models (PLMs) on the quality of augmented data, we present the PLMs adopted by GDA and baseline augmentation techniques and their corresponding parameters in Table 2. An exciting conclusion is that compared to Text Gen, although we use PLMs with fewer parameters (345M vs. 220M), our augmentation effect is still improved by an astonishing 0.6% compared to Text Gen, and a new SOTA for the RE task has been achieved. Even though we adopt T5-Small (60M) in GDA, which has fewer parameters than BERT-Base and GPT-2 ($\approx$ 110M), the augmented data can still bring competitive F1 improvement. More specifically, GDA (T5-Small) can achieve F1 improvement of 0.9% and 1.1% on SURE and RE-DMP, respectively, which illustrates the effectiveness of GDA for data augmentation in RE task.

4.4 Ablation Study

We conduct an ablation study to show the effectiveness of different modules of GDA on test set. GDA *w/o Restructuring* is the proposed technique without the decoder part θ_R and only uses the original sentence pattern approximation task to train the T5. GDA *w/o Approximation* is the proposed technique without the decoder part θ_P and entity hint from the target sentence, and we use θ_R for generation during both training/inference. GDA *w/o Two Tasks* directly fine-tunes T5 on the training data, only requiring that the input sentence to be from the same relation as the target sentence.

Methods / Datasets	PLMs	SemEval	TACRED	TACREV
SURE	BART-Large	86.3	73.3	79.2
+EDA	–	86.7	73.8	79.6
+Paraphrase Graph	–	86.6	73.7	79.4
+Ex2	T5-Base	87.5	74.3	80.3
+Text Gen	GPT-2-Medium	87.7	74.4	80.4
+Restructured	–	87.0	74.0	79.8
+Pattern	–	87.2	74.1	80.0
+**GDA**	T5-Base	**88.0**	**74.9**	**80.8**

Table 3: We adopt SURE as the base model and use 100% training data over three datasets. We report F1 results on the test sets. "Restructured" and "Pattern" mean to directly use restructured original sentences and pattern approximation target sentences as augmentations.

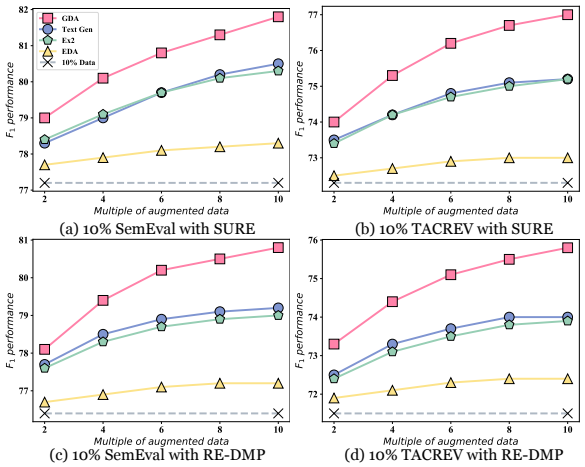
A general conclusion from the ablation results in Table 2 is that all modules contribute positively to GDA. More specifically, without multi-task learning framework, GDA *w/o Two Tasks* brings 1.3% less F1 performance averaged over three datasets. Similarly, compared with the restructuring task, the pattern approximation task can bring more average improvement in F1 performance (0.6% vs. 0.8%), which also means that we need to focus more on the pattern approximation task when training T5.

4.5 Generative Model Ablation Study

We additionally study the effect of removing the generative model on the augmentation effect, that is, we directly use restructured original sentences and pattern approximation target sentences as augmented sentences. From Table 3, we found that directly using restructured sentences and pattern approximation sentences as augmented data results in a 1.3% drop in F1 performance compared to GDA, which indicates the necessity of using T5-Base to train augmented sentences. These two augmented sentences are also rule-based techniques. Compared with other rule-based data augmentation techniques (EDA and Paraphrase Graph), they can bring an average F1 improvement of 0.4%, which additionally illustrates the effectiveness of our modification of the original sentences on the RE tasks.

4.6 Performance on Various Augmentation Multiples

We vary the multiple of augmented data from 2x to 10x the 10% training set to study the influence of data augmentation techniques for the base models under low-resource scenarios. We choose the



(a) 10% SemEval with SURE (b) 10% TACREV with SURE

(c) 10% SemEval with RE-DMP (d) 10% TACREV with RE-DMP

Figure 3: F1 results of the base model: SURE and RE-DMP with various multiples of the augmented data.

10% SemEval and 10% TACREV training datasets and the base models: SURE and RE-DMP, then represent the results on the test set in Figure 3.

We observe that two base models have more performance gains with ever-increasing augmented data and GDA achieves consistently better F1 performance, with a clear margin, when compared with baseline data augmentation techniques under various multiples of the augmented data. Especially for 10% TACREV, GDA brings an incredible 3% improvement in F1 performance with only 4x augmented data, which is even 0.2% better than adopting 25% (2.5x) of the training data directly.

4.7 Diversity Evaluation

We measure the diversity of augmented sentences through automatic and manual metrics. For automatic metric, we introduce the Type-Token Ratio (TTR) (Tweedie and Baayen, 1998) to measure the ratio of the number of different words to the total number of words in the dependency path between two entities for each relation type. Higher TTR (%) indicates more diversity in sentences. Besides that, we ask 5 annotators to give a score for the degree of diversity of the 30 generated sentences for each relation type, with score range of 1~5. According to the annotation guideline in Appendix C, a higher score indicates the method can generate more diverse and grammatically correct sentences. We present the average scores for all relation types on three datasets in Table 5. Since the example interpolation techniques do not generate the sentences shown, they are ignored. As a model-based augmentation technique, GDA could obtain 11.4% TTR and 0.4 diversity performance boost in average compared to Text Gen, and can even have a diversity capability similar to the rule-based methods. Furthermore, we give the detailed hyperparameter analysis in Appendix.

Original: The meeting was opened by the welcome speech of the Mayor of Komotini.
Relation Label: **Component-Whole**
Entity Hint: **program**
GDA:The program was opened by the host, who was a former member of the Congressional Black Caucus.
GDA *w/o Pattern Approximation*: The program consists of a seminar and a workshop.
GDA *w/o Entity Hint*: The ricotta mixture was the best part of this dish.
Original: This train has as many as six sets of doors.
Relation Label: **Component-Whole**
Entity Hint: **kitchen**
GDA: The kitchen has sets of stove, and a microwave.
GDA *w/o Pattern Approximation*: The kitchen contents include a refrigerator.
GDA *w/o Entity Hint*: The ricotta mixture was the best part of this dish.

Table 4: Case study. We highlight the entities and pattern in the original and generated sentences.

Methods / Datasets	SemEval		TACRED		TACREV	
	TTR	Diver.	TTR	Diver.	TTR	Diver.
EDA	82.4	3.1	**84.7**	3.4	83.2	3.3
Paraphrase Graph	85.9	3.6	84.1	3.9	84.6	3.5
LAMBADA	72.6	2.3	76.2	2.2	78.4	2.2
DARE	75.3	2.4	75.8	2.7	74.7	2.5
Text Gen	74.8	3.5	72.1	3.7	76.3	3.6
GDA (T5-Base)	**86.4**	**4.0**	84.1	**4.1**	**86.9**	**3.9**
GDA *w/o Approximation*	74.7	3.2	75.2	3.2	72.8	3.1
GDA *w/o Restructuring*	80.3	3.8	81.3	3.7	82.0	3.8

Table 5: Diversity Evaluation on three datasets.

4.8 Case Study

We give two cases in Table 4. GDA adopts the entity hint: "program" and input sentence to generate a controllable target sentence, while retaining the original pattern: "was opened by" without changing the semantics. GDA *w/o Pattern Approximation* converts the rare pattern "was opened by" to the high frequency pattern "consists of" due to the inductive bias, which will affect the diversity of augmented sentences. GDA *w/o Entity Hint* will generate uncontrollable entities, resulting in the same sentence generated by the same relation, which affects the diversity of generated sentences.

4.9 Coherence Analysis

Compared to rule-based augmentation techniques, GDA conditionally generates pseudo sentences with entity hints, providing more coherent and reasonable sentences. We analyze the coherence of the augmented sentences through perplexity based on GPT-2 (Radford et al., 2019). Note that the exam-

Methods / Datasets	SemEval	TACRED	TACREV
EDA	8.24	9.18	8.33
Paraphrase Graph	7.44	7.88	7.01
LAMBADA	4.21	4.38	4.11
DARE	4.28	4.46	4.22
Text Gen	4.02	4.24	4.11
GDA (T5-Base)	**3.97**	**4.21**	**4.05**
Original	3.88	4.09	3.91

Table 6: Perplexity of the augmented sentences in three datasets. Original means the original sentences. Lower perplexity is better.

ple interpolation techniques interpolate the embeddings and labels of two or more sentences without the generation of specific sentences, so we did not compare these methods.

From Table 6, GDA could obtain the lowest average perplexity. Although Text Gen is also based on generative models, the augmented sentences are still not coherence enough due to the neglect of entity-level relational signals (entity hint) during the training process. Therefore, Text Gen is not so natural in generating augmented sentences with entity annotations.

4.10 Semantic Consistency Analysis

Unlike rule-based data augmentation techniques, which will change the semantics of the original sentence, GDA can better exploit relational signals: the target sentence during the training process comes from the restructured original sentence with the same relation label, so GDA can generate semantically consistent augmented sentences.

In our tasks, we first train SURE on the 100% training datasets and then apply GDA to the test set to obtain augmented sentences. We feed the 100 original sentences and 100 augmented sentences with the same relation labels into the trained SURE, and obtain the output representations from the last dense layer. We apply t-SNE (Van Der Maaten, 2014) to these embeddings and plot the visualization of the 2D latent space. From Figure 4, we observed that the latent space representations of the augmented sentences closely surrounded those of the original sentences with the same relation labels, indicating that GDA could augment sentences semantically consistently. Conversely, sentences augmented with rule-based method: EDA appear outliers, indicating semantic changes.

5 Conclusions and Future works

In this paper, we propose a relational text augmentation technique: GDA for RE tasks. Unlike conventional data augmentation techniques, our technique adopts the multi-task learning framework to generate diverse, coherent, and semantic consistent augmented sentences. We further adopt entity hints as prior knowledge for diverse generation. Experiments on three public datasets and low-resource settings could show the effectiveness of GDA. For future research directions, we can try to explore more efficient pre-ordering and parsing methods, and apply our data augmentation methods to more NLP applications, such as semantic parsing (Liu et al., 2022a, 2023), natural language inference (Li et al., 2023, 2022b).

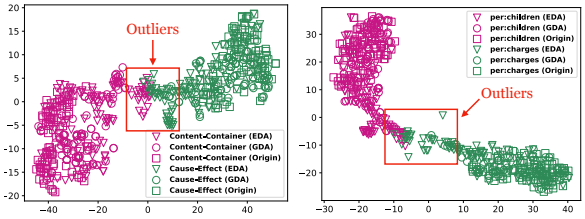


Figure 4: Latent space visualization of original and augmented sentences in the SemEval (left) and TACRED (right). The same relation labels use the same color.

6 Limitations

We would like to claim our limitations from two perspectives: application-wise and technical-wise.

Application-wise: GDA needs annotations to fine-tune T5, which requires more computing resources and manual labeling costs than the rule-based techniques.

Technical-wise: Our "original sentence restructuring" and "original sentence pattern approximation" tasks rely on the efficiency and accuracy of pre-ordering rules (Wang et al., 2007) and parsing methods (Chen and Manning, 2014). Although current GDA show effectiveness, we still need to find more efficient pre-ordering and parsing methods.

7 Acknowledgement

We thank the reviewers for their valuable comments. The work described here was partially supported by grants from the National Key Research and Development Program of China (No. 2018AAA0100204) and from the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14222922, RGC GRF, No. 2151185), NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Tacred revisited: A thorough evaluation of the tacred relation extraction task. In *Proc. of ACL*, pages 1558–1569.

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 *Proc. of AAAI*, volume 34, pages 7383–7390.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Markus Bayer, Marc-André Kaufhold, Björn Buchhold, Marcel Keller, Jörg Dallmeyer, and Christian Reuter. 2022. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International Journal of Machine Learning and Cybernetics*, pages 1–16.

Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Proc. of ACL*, pages 6334–6343.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proc. of EMNLP*, pages 740–750.

Hannah Chen, Yangfeng Ji, and David K Evans. 2020a. Finding friends and flipping frenemies: Automatic paraphrase dataset augmentation using graph theory. In *Proc. of EMNLP: Findings*, pages 4741–4751.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020b. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proc. of ACL*, pages 2147–2157.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proc. of EMNLP*, pages 875–886.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. In *Proc. of ACL-IJCNLP: Findings*, pages 968–988.

Varun Gangal, Steven Y Feng, Malihe Alikhani, Teruko Mitamura, and Eduard Hovy. 2021. Nareor: The narrative reordering problem. *arXiv preprint arXiv:2104.06669*.

Fei Gao, Jinhua Zhu, Lijun Wu, Yingce Xia, Tao Qin, Xueqi Cheng, Wengang Zhou, and Tie-Yan Liu. 2019. Soft contextual data augmentation for neural machine translation. In *Proc. of ACL*, pages 5539–5544.

Demi Guo, Yoon Kim, and Alexander M Rush. 2020. Sequence-level mixed sample data augmentation. In *Proc. of EMNLP*, pages 5547–5552.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O 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 *Proc. of SemEval*, pages 33–38.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proc. of COLING*, pages 1234–1245.

Xuming Hu, Zhaochen Hong, Chenwei Zhang, Irwin King, and Philip S Yu. 2023. Think rationally about what you see: Continuous rationale extraction for relation extraction. *arXiv preprint arXiv:2305.03503*.

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and Philip S. Yu. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *Proc. of EMNLP*, pages 3673–3682.

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 EMNLP*, pages 487–496.

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 *Proc. of EMNLP*, pages 2737–2746.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proc. of EACL*, pages 1022–1033.

Chang Jin, Shigui Qiu, Nini Xiao, and Hao Jia. 2022. Admix: A mixed sample data augmentation method for neural machine translation. *Proc. of IJCAI*, pages 4171–4177.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*.

Divyansh Kaushik, Amrith Setlur, Eduard H Hovy, and Zachary Chase Lipton. 2020. Explaining the efficacy of counterfactually augmented data. In *ICLR*.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proc. of NAACL-HLT*, pages 452–457.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *TACL*, 8:329–345.

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. 2021. Neural data augmentation via example extrapolation. *arXiv preprint arXiv:2102.01335*.

Bohan Li, Yutai Hou, and Wanxiang Che. 2022a. Data augmentation approaches in natural language processing: A survey. *AI Open*.

Shu'ang Li, Xuming Hu, Li Lin, Aiwei Liu, Lijie Wen, and Philip S. Yu. 2023. A multi-level supervised contrastive learning framework for low-resource natural language inference. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1771–1783.

Shu'ang Li, Xuming Hu, Li Lin, and Lijie Wen. 2022b. Pair-level supervised contrastive learning for natural language inference. *arXiv preprint arXiv:2201.10927*.

Aiwei Liu, Xuming Hu, Li Lin, and Lijie Wen. 2022a. Semantic enhanced text-to-sql parsing via iteratively learning schema linking graph. In *Proc. of KDD*, pages 1021–1030.

Aiwei Liu, Xuming Hu, Lijie Wen, and Philip S Yu. 2023. A comprehensive evaluation of chatgpt's zero-shot text-to-sql capability. *arXiv preprint arXiv:2303.13547*.

Shuliang Liu, Xuming Hu, Chenwei Zhang, Shu'ang Li, Lijie Wen, and Philip S. Yu. 2022b. Hiure: Hierarchical exemplar contrastive learning for unsupervised relation extraction. In *Proc. of NAACL-HLT*, pages 5970–5980.

Keming Lu, I Hsu, Wenxuan Zhou, Mingyu Derek Ma, Muhao Chen, et al. 2022. Summarization as indirect supervision for relation extraction. In *Proc. of EMNLP*.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proc. of ACL*, pages 2339–2352.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. *arXiv preprint arXiv:2004.13845*.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? an empirical study on neural relation extraction. In *Proc. of EMNLP*, pages 3661–3672.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. 2017. Learning to compose domain-specific transformations for data augmentation. *NeurIPS*, 30.

Shuhuai Ren, Jinchao Zhang, Lei Li, Xu Sun, and Jie Zhou. 2021. Text autoaugment: Learning compositional augmentation policy for text classification. In *Proc. of EMNLP*, pages 9029–9043.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In *Proc. of EMNLP*, pages 5004–5009.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proc. of ACL*, pages 86–96.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proc. of ACL*, pages 2895–2905.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2021. Progressive multitask learning with controlled information flow for joint entity and relation extraction. *Proc. of AAAI*.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Improving relation extraction through syntax-induced pretraining with dependency masking. In *Proc. of ACL: Findings*, pages 1875–1886.

Fiona J Tweedie and R Harald Baayen. 1998. How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, 32(5):323–352.

Laurens Van Der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *The journal of machine learning research*, 15(1):3221–3245.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proc. of EMNLP-CoNLL*, pages 737–745.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. In *Proc. of EMNLP*, pages 856–861.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proc. of EMNLP-IJCNLP*, pages 6382–6388.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proc. of ACL*, pages 5786–5796.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *ICLR*.

Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE TPAMI*, 29(6):1091–1095.

```
func LevenshteinDistance(char s[1..m], char t[1..n]){
  // for all i and j, d[i,j] will hold the Levenshtein distance between
  // the first i characters of s and the first j characters of t
  declare int d[0..m, 0..n]

  set each element in d to zero

  // source prefixes can be transformed into empty string by
  // dropping all characters
  for i from 1 to m:
    d[i, 0] := i

  // target prefixes can be reached from empty source prefix
  // by inserting every character
  for j from 1 to n:
    d[0, j] := j

  for j from 1 to n:
    for i from 1 to m:
      if s[i] = t[j]:
        substitutionCost := 0
      else:
        substitutionCost := 1

      d[i, j] := minimum(d[i-1, j] + 1,
                         // deletion
                         d[i, j-1] + 1,
                         // insertion
                         d[i-1, j-1] +
      substitutionCost)  // substitution

  return d[m, n]
}
```


Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating relation paths in neural relation extraction. In *Proc. of EMNLP*, pages 1768–1777.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *ICLR*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *NeurIPS*, 28:649–657.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proc. of EMNLP*, pages 35–45.

A Levenshtein Distance Pseudo Code Implementation

We give a pseudo code implementation of Levenshtein Distance algorithm.

B Hyperparameter Analysis

We study the hyperparameter λ in the original sentence pattern approximation task, which represents the similarity between the pattern of the target sentence and the input sentence. An appropriate λ can bring diverse generated sentences during generation process. We vary the λ from 1 to 5 and represent the F1 results on the test set using the SURE model with 100% training data in Table 7.

Datasets / λ	1	2	3	4	5
SemEval	87.4	87.8	**88.0**	87.9	87.2
TACRED	73.8	74.7	**74.9**	74.5	74.0
TACREV	80.0	80.3	80.8	**81.0**	80.4

Table 7: F1 performance under different λ.

With no more than 1% F1 fluctuating results among three datasets, GDA is robust enough to λ. Besides, the results indicate that both approximation and coverage of target sentences will impact performance. Using a high λ will introduce target sentences with low pattern approximation, causing the inductive bias problem. Low λ will cause the low coverage of target sentences, affecting the training effect on T5.

C Annotation Guideline

Each annotator needs to carefully read each augmented sentence, compare it with the original sentence, and give a score according to the following criteria. Note that all augmented sentences for a relation label are given an average score, and the final score is the average of all relation labels.

- Score:1. The augmented sentences under the same relation are almost the same.
- Score:2. The augmented sentences under the same relation are slightly different, with serious grammatical errors.
- Score:3. The augmented sentences under the same relation are slightly different, and there are almost no grammatical errors.
- Score:4. The augmented sentences under the same relation are diverse, with serious grammatical errors.
- Score:5. The augmented sentences under the same relation are diverse, and there are almost no grammatical errors.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

☑ A2. Did you discuss any potential risks of your work?
Section 6

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

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

B ☑ Did you use or create scientific artifacts?

Section 3, Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 3, Section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3, Section 4

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3, Section 4

☑ 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?
Section 4.2, Appendix B

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.2, Section 5, Appendix B

☑ 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.
Section 4.2, Appendix B

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4.2

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

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

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4.3

☑ 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.)?
Section 4.2, Section 4.3

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

Section 4.7, Appendix C

☑ 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.?
Appendix C

☑ 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)?
Section 4.7

☑ 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?
Section 4.7, Appendix C

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.