Learning with Silver Standard Data for Zero-shot Relation Extraction

Tianyin Wang[1,2], **Jianwei Wang**[1,2], **Ziqian Zeng**[1]
[1]Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, China
[2]School of Computer Science & Engineering, South China University of Technology, China
{cstianyinwang,201930420160}@mail.scut.edu.cn, zqzeng@scut.edu.cn

Abstract

The superior performance of supervised relation extraction (RE) methods heavily relies on a large amount of gold standard data. Recent zero-shot relation extraction methods converted the RE task to other NLP tasks and used off-the-shelf models of these NLP tasks to directly perform inference on the test data without using a large amount of RE annotation data. A potentially valuable by-product of these methods is the large-scale silver standard data. However, there is no further investigation on the use of potentially valuable silver standard data. In this paper, we propose to first detect a small amount of clean data from silver standard data and then use the selected clean data to finetune the pretrained model. We then use the finetuned model to infer relation types. We also propose a class-aware clean data detection module to consider class information when selecting clean data. The experimental results show that our method can outperform the baseline by 12% and 11% on TACRED and Wiki80 dataset in the zero-shot RE task. By using extra silver standard data of different distributions, the performance can be further improved.

1 Introduction

Relation extraction (RE) is a fundamental problem in information extraction. It aims to identify the semantic relation between two entities in unstructured texts. The predominant approaches to solve the relation extraction task are supervised learning methods (Kambhatla, 2004; Zhou et al., 2005; Zeng et al., 2014; Soares et al., 2019; Yamada et al., 2020; Zhong and Chen, 2021; Lyu and Chen, 2021). Supervised learning methods require a large amount of gold standard data, which restricts their applications to real-world scenarios where large-scale annotated data are not available.

Recent works (Sainz et al., 2021; Lu et al., 2022) attempt to convert the RE task to other NLP tasks and used off-the-shelf models of these tasks to infer the relation types without using a large amount of RE annotated data. LaVeEntail (Sainz et al., 2021) used a well-trained textual entailment (TE) model to directly infer relation types on the RE test data by converting a RE task to a TE task. SURE (Lu et al., 2022) formulated a RE task to a summarization task, and used a small amount of RE annotated data to finetune a well-trained summarization model thus it can perform inference on the RE test data. We term the TE model and summarization model in above methods as pretrained models. The concept of pretraining derives from transfer learning (Pan and Yang, 2009). A model is first pretrained on the source task, i.e., textual entailment or summarization, and then finetuned on the target task, i.e., relation extraction.

Since pretrained models can directly infer the relation types of unlabeled data, they can serve as low-cost annotators, producing large-scale silver standard data. However, in above works, silver standard data are not well-exploited. The straightforward way to utilize them is to directly train a traditional supervised RE system on silver standard data. But the performance is normally unsatisfactory due to the noisy nature of silver standard data. Learning with noisy labels has been well studied in the literature (Frénay and Verleysen, 2013; Algan and Ulusoy, 2021; Han et al., 2020). One direction is to develop noise-robust losses that can mitigate the effect of noisy labels (Ghosh et al., 2017; Zhang and Sabuncu, 2018; Charoenphakdee et al., 2019; Kim et al., 2019; Lyu and Tsang, 2019; Menon et al., 2020; Thulasidasan et al., 2019). Another direction is to identify noisy data or clean data and deal with them separately either by re-weighting or converting to a semi-supervised learning task. (Han et al., 2018a; Jiang et al., 2018; Arazo et al., 2019; Kim et al., 2019; Shu et al., 2019; Yao et al., 2019; Li et al., 2020). The setting of traditional noisy labels learning does not consider the existence of

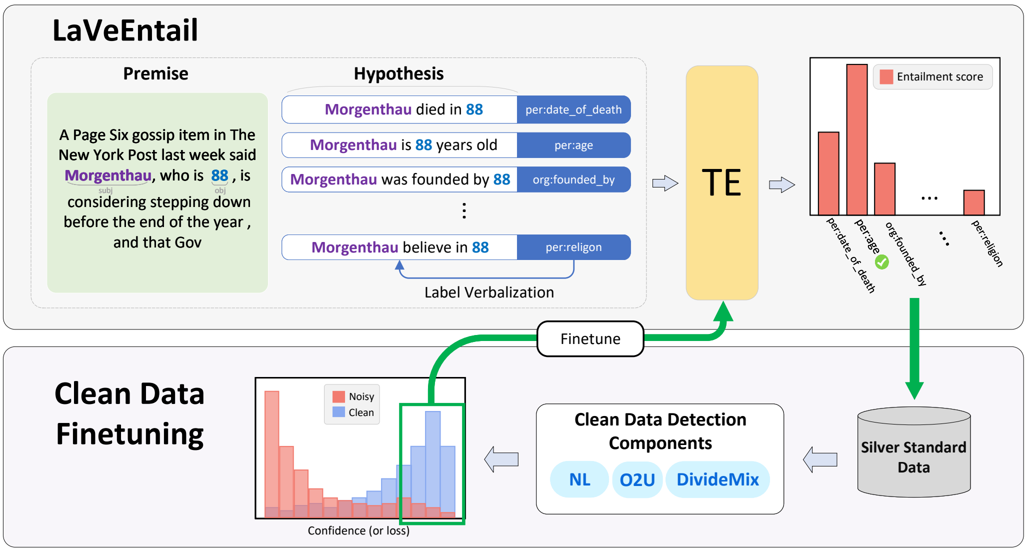


Figure 1: The diagram shows the procedure of our method. First, we apply LaVeEntail on an unlabeled dataset, obtaining standard silver data. The clean data detection module uses confidence scores to distinguish the clean and noisy samples. The selected clean data are used to finetune the TE model. Finally, we use this TE model to infer relation types on the relation extraction test set.

a pretrained model. Is there a better way to utilize silver standard data when a pretrained model is available? According to our best knowledge, there is no further investigation on the use of potentially valuable silver standard data when there exists a pretrained model.

In this paper, we propose to first detect a small amount of clean data from silver standard data and then use the selected clean data to finetune the pretrained model. The procedure is shown in Figure 1. In the clean data detection module, we used a noise indicative metric, i.e, confidence scores, to select clean data. However, the clean data detection module selects clean data without considering class information. Samples in some classes can yield very high confidence scores. Large quantities of samples in those classes are selected. But some classes even do not have clean data. It can harm performance severely. Hence, we develop a class-aware clean data detection module that selects some clean data from each class.

To obtain silver standard data, we use the pretrained model to annotate the unlabeled RE training data. The distribution of silver standard data is the same as the RE test data. However, unlabeled data of the same distribution with test data are still scarce while unlabeled data of different distribution are common. Hence, we use the finetuned model to annotate some unlabeled data of different distributions, i.e., only partial classes are overlapped with the test data. The experimental results show that our method can utilize the silver standard data of different distribution to further improve the performance.

Our contributions are summarized as follows,

• We propose to first detect a small amount of clean data which are later used to finetune the pretrained model. We then use the finetuned model to infer the relation types on the RE test data.

• We propose a class-aware clean data detection module that can consider class information.

• The experimental results show that our method can outperform the baseline by a large margin on both datasets. By using additional silver standard data of different distributions, the performance can be further improved.

2 Related Work

Supervised Relation Extraction. The predominant approaches to solve the relation classification task are supervised learning methods (Kambhatla, 2004; Zhou et al., 2005; Zeng et al., 2014; Wu and He, 2019; Yamada et al., 2020; Zhong and Chen, 2021; Lyu and Chen, 2021). Before the era of pretrained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019), supervised learning methods require a large amount of annotated data to train models from scratch (Kambhatla, 2004; Zhou et al., 2005; Zeng et al., 2014). Later, finetuning on PLMs methods (Wu and He, 2019; Zhong and

Chen, 2021; Lyu and Chen, 2021) outperformed traditional methods.

Zero-shot and Few-shot Relation Extraction. Recent works attempt to address RE in a low data regime, namely zero-shot or few-shot relation extraction (Han et al., 2018b; Soares et al., 2019; Chen and Li, 2021) which require zero or few examples for each relation type during the test phase. However, they still require a large amount of annotated relation classification data for training. In zero-shot and few-shot settings, the classes in the test phase are never seen in the training phase. Levy et al. (2017) reformulated the zero-shot slot filling task to a reading comprehension task. The slot filling task aims to predict the entity given a relation type and another entity. (Obamuyide and Vlachos, 2018) reduced the zero-shot and few-shot relation classification task to a textual entailment problem. The premise is the sentence containing two entities. The hypothesis is a relation description template instantiated by two entities.

In the modern zero-shot and few-shot frameworks, the training annotations are optional. They obtained supervision from other available resources such as language models, relation description, and other NLP tasks. (Goswami et al., 2020) reformulated the zero-shot slot filling task to a cloze question answering problem. LaVeEntail (Sainz et al., 2021) utilized an off-the-shelf textual entailment model to directly infer RE data. (Tran et al., 2021) compared the similarity of the embeddings of manually created relation exemplars and the input sentence to infer the relation types. SURE (Lu et al., 2022) converted the few-shot RE task to a summarization task. The input sentence in RE is the long document input. The ground truth relation description is the shortened summary. LaVeEntail(Sainz et al., 2021) is the state-of-the-art method in zero-shot RE task.

Learning with Noisy Labels. One direction of learning with noisy labels is to develop noise-robust loss. The widely-used cross entropy (CE) loss in classification tasks has been shown to be not robust against label noise (Ghosh et al., 2017). Several noise-robust losses have been proposed for training models with noisy labels (Reed et al., 2015; Zhang and Sabuncu, 2018; Wang et al., 2019; Ma et al., 2020; Menon et al., 2020; Jin et al., 2021; Zhou and Chen, 2021), which were shown to be more robust than CE. However, since current deep networks have a large number of parameters, these methods can still memorize the noisy labels given sufficient training time (Zhang et al., 2017a).

Another direction is to identify noisy data or clean data and cope with them separately either by re-weighting them or converting the problem to a semi-supervised learning task. (Arpit et al., 2017; Charoenphakdee et al., 2019) found out the memorization effect which is stated as although deep networks can memorize noise data, they tend to learn simple patterns first. Based on the memorization effect (Arpit et al., 2017; Zhang et al., 2021), many methods separate clean and noisy samples by using loss value (Han et al., 2018a; Jiang et al., 2018; Arazo et al., 2019; Shu et al., 2019; Yao et al., 2019; Li et al., 2020) and number of disagreement or forgetting events (Malach and Shalev-Shwartz, 2017; Yu et al., 2019). The re-weighting methods (Ren et al., 2018; Shu et al., 2019) learned optimal weights for different samples by using meta-learning (Hospedales et al., 2020). The semi-supervised learning methods (Kim et al., 2019; Huang et al., 2019; Li et al., 2020) divide the training data into a labeled set with clean samples and an unlabeled set with noisy samples, and trains the model on both the labeled and unlabeled data in a semi-supervised manner. The setting of traditional noisy labels learning does not consider the existence of a pretrained model.

3 Method

LaVeEntail(Sainz et al., 2021) is the state-of-the-art method in the zero-shot relation extraction task, hence we use LaVeEntail as the backbone to obtain silver standard data. We will first introduce the LaVeEntail method in Section 3.1, the clean data detection module in Section 3.2, the class-aware clean data detection module in Section 3.3, and the finetuning and inference in Section 3.4.

3.1 LaVeEntail

LaVeEntail (Sainz et al., 2021) includes two processes, i.e., label verbalization and textual entailment model inference.

3.1.1 Label Verbalization

The label verbalization process creates templates of relation types and then uses them to generate hypotheses. The templates can be easily created because relation labels naturally implicate such verbalization templates. For example, the relation `per:schools_attended` can be verbalized as {subj} studied in {obj}. Given an input sentence x with subject entity being x_{subj} and objective entity being x_{obj}, the hypothesis is generated by substituting x_{subj} and x_{obj} to corresponding placeholders, i.e., {subj} and {obj}.

3.1.2 Textual Entailment Model Inference

Textual Entailment (TE) is the task of predicting whether, for a premise-hypothesis pair, the facts in the premise necessarily imply the facts in the hypothesis.

For each input sentence, LaVeEntail constructed hypotheses that are generated by verbalization templates of all relation types, and fed them to a TE model, and obtained entailment scores of all hypotheses. LaVeEntail inferred that the predicted relation type of the input sentence is the relation type whose hypothesis yields the highest entailment score. Figure 1 shows the inference procedure.

Entity type information is helpful to infer relation types (Tran et al., 2020). A relation naturally indicates entity types of subject and object. For instance, the relation per:city_of_death implicates that the entity type of subject and object should be PERSON and CITY respectively. In the inference stage, when the entity type information is given, we could rule out some relation types which are impossible to be ground truth. For example, in TACRED, given that the subject entity type is PERSON and the object entity type is CITY, possibly correct relation types are per:city_of_death, per:city_of_birth, and per:cities_of_residence. Other relation types such as org:founded are impossible to be the ground truth. LaVeEntail created entity type constraint(s) for each relation according to the meaning of the relation. If the entity types in the input sentence do not match the entity type constraints of a relation, then the entailment score(s) of all hypotheses related to this relation is set to zero.

In the case where there is no relation between two entities, a threshold-based approach is used to detect no_relation. If the entailment scores of all hypotheses are less than a threshold, the prediction is no_relation.

3.2 Clean Data Detection

The clean data detection module utilizes the values of noise indicative metrics such as confidence scores to distinguish the clean and noisy samples. After training with a clean data detection algorithm, we can obtain the confidence score of each sample, we then select clean data D_{clean} from D_{silver} based on the metric. D_{silver} is the silver standard data set obtain by LaVeEntail.

The clean data detection module is adapted from a noisy labels learning method in the Computer Vision area, i.e., Negative Learning for Noisy Labels (NLNL) (Kim et al., 2019). We only adopt the Negative Learning (NL) method in NLNL as the clean data detection component.

Different from positive learning loss (e.g., CE loss) which tells the model what is correct, the negative learning loss provides the model with the complementary label(s), telling what is not correct, e.g., the input image is not a dog. The complementary label is randomly selected from the label space excluding the input label (possibly noisy). For noisy data, the probability of selecting the ground truth as the complementary label is low. Hence, using negative learning loss can decrease the risk of overfitting noisy labels. The formula of NL loss is shown as follows,

$$\mathcal{L}_{neg} = -\sum_{d \in D} \sum_{i=1}^{|\mathcal{Y}|} \widehat{\mathbf{y}}_i^d \log(1 - \mathbf{p}_i^d), \quad (1)$$

where d is a sample in the dataset D, $|\mathcal{Y}|$ is the number of relation types, $\widehat{\mathbf{y}}^d$ is a one-hot vector with the complementary label being one, $\widehat{\mathbf{y}}_i^d$ is the i-th element of $\widehat{\mathbf{y}}^d$, $\widehat{\mathbf{p}}^d$ is the output probability distribution of a smaple d, and $\widehat{\mathbf{p}}_i^d$ is the i-th element of $\widehat{\mathbf{p}}^d$.

After training a classifier (the architecture is shown in Appendix B) using negative learning loss, we sort whole data by their confidence scores. We select a fixed proportion η of whole data as the clean data set. Samplers with higher confidence scores have higher priority to be selected. Given that $\mathcal{S}(D_s)$ is the total confidence scores of all samples in D_s, and η is a hyperparameter representing proportion, the clean data set D_{clean} are selected as follows,

$$D_{clean} = \arg\max_{D_s:|D_s|=\eta \cdot |D_{silver}|} \mathcal{S}(D_s). \quad (2)$$

3.3 Class-aware Clean Data Detection

The clean data detection module selects clean data according to their confidence scores. It does not consider class information. Samples in some classes can yield very high confidence scores.

Algorithm 1 Class-aware Clean Data Detection

Input: silver standard data set D_{silver}, proportion η, diversity number m, the set of classes $\mathcal{C}$, the total confidence scores function $\mathcal{S}(\cdot)$.

$D_{clean} = \varnothing$.
Obtain D_{clean} using Eq. 2 by setting the proportion to η.
$D_{rest} = D_{silver} - D_{clean}$, divide D_{rest} into $|\mathcal{C}|$ subsets according to class predictions. The subset for class c is denoted as D^c.
for c in $\mathcal{C}$ **do**
  $D^c_{clean} = \arg\max_{D_s:|D_s|=\frac{|D^c|}{|D_{rest}|}\cdot m} \mathcal{S}(D_s)$
  $D_{clean} = D_{clean} \cup D^c_{clean}$
end for

Output: clean data set D_{clean}.

Large quantities of samples in those classes are selected in D_{clean}. But some classes do not have any clean data in D_{clean}. It can harm performance severely.

We propose a class-aware clean data detection algorithm that considers confidence scores as well as class information. First, we select a proportion η of data with high confidence scores. This step can ensure that samples with low noise levels are selected. Next we select m more samples to encourage diversity. For each class, we select some samples with high confidence scores in this class. The number of selected samples for each class is proportional to the number of the class in the prediction. Ideally, we should select samples according to the true class distribution. But it is unknown, we estimate it based on the class distribution in prediction. This step aims to involve samples from more classes. The class-aware clean data detection module is presented in Algorithm 1.

The number of selected samples of a class is dynamic, which is determined by the number of samples of a class in the candidate pool. The counterpart of the dynamic strategy is selecting a fixed number of samples for each class. We compare dynamic and fixed methods in the experiment.

3.4 Finetuning and Inference

After running clean data detection algorithms, we obtain D_{clean} which consists of the input sentence and its relation type pairs. Since the input forms of the TE and RE task are different, we need to convert D_{clean} to premise-hypothesis pairs so that we can

Algorithm 2 Zero-shot RE Utilizing Silver Standard Data

Input: silver standard data set D_{silver}, test set D_{test}, textual entailment model $\mathcal{M}$.

Obtain D_{clean} using Eq. 2 or Algorithm 1.
Generate premise hypothesis pairs dataset $D^{'}_{clean}$ based on D_{clean}.
Finetune $\mathcal{M}$ using $D^{'}_{clean}$, and obtain finetuned model $\mathcal{M}'$.
Use $\mathcal{M}'$ to infer relation types on D_{test}.

Output: relation types of samples on D_{test}.

use D_{clean} to finetune the TE model. The premise-hypothesis construction procedure is shown in Appendix C.

We use premise hypothesis pairs constructed from D_{clean} to finetune the off-the-shelf TE model. Finally, we use the finetuned TE model to infer relation types on the test set. The complete algorithm is presented in Algorithm 2.

4 Experiment

4.1 Experiemtal Settings

We conduct experiments on TACRED (Zhang et al., 2017b) and Wiki80 (Han et al., 2019). The statistics of two datasets are shown in Appendix D.1. We use an off-the-shelf TE model to annotate the training set as the silver standard data.

For each dataset, we manually created verbalization templates and the entity type constraints, which are shown in the Appendix D.8 and Appendix D.9 respectively. There is no entity type information on Wiki80. We describe how to generate entity types for Wiki80 in Appendix D.2. For LaVeEntail and our method, we only use 1% of the development set to select hyper-parameters.

4.2 Compared Methods

To demonstrate the effectiveness of our method, we compare our model with the following baselines:

LaVeEntail (Sainz et al., 2021) utilized off-the-shelf textual entailment model to directly infer on the relation extraction test data. **Labeled Data Finetune** randomly select a proportion of labeled training data to finetune the off-the-shelf TE model.

We consider the following baselines using different loss functions: training a supervised relation classification model on silver standard data using **CE** (Cross Entropy loss) and different noise-robust

Method	TACRED			Wiki80		
	Pr.	Rec.	F1	Pr.	Rec.	F1
LaVeEntail (Sainz et al., 2021)	58.02	44.73	52.18±0.00	49.09	41.16	41.16±0.00
Labeled Data Finetune (1%)	49.39	56.60	56.61±1.29	56.60	47.39	47.39±0.33
Labeled Data Finetune (5%)	58.76	52.06	63.72±1.03	61.10	53.89	53.89±0.46
CE	50.47	33.25	45.35±0.58	51.15	40.76	40.76±0.29
BSH (Reed et al., 2015)	50.23	31.23	46.20±1.94	51.18	40.86	40.86±0.52
GCE (Zhang and Sabuncu, 2018)	50.14	31.61	45.93±0.67	49.96	41.28	41.28±0.61
SCE (Wang et al., 2019)	50.32	31.86	45.82±0.92	50.70	41.12	41.12±0.24
ER-GCE (Jin et al., 2021)	55.78	31.85	44.90±1.26	49.61	40.93	40.93±0.41
Co-Regularization (Zhou and Chen, 2021)	66.69	38.58	48.86±0.34	28.50	28.48	28.48±0.42
Self-training (Yarowsky, 1995)	21.99	10.85	30.59±0.22	42.60	37.13	37.13±0.12
O2U (Huang et al., 2019)	58.23	34.21	47.52±0.81	53.57	42.62	42.62±0.03
NLNL (Kim et al., 2019)	46.55	34.38	48.17±0.86	48.00	41.71	41.71±0.34
DivideMix (Li et al., 2020)	37.08	49.41	49.78±0.80	48.99	45.52	45.52±0.26
All Data Finetune	56.88	46.01	54.67±0.58	55.13	44.57	44.57±0.31
Clean Data Finetune	53.92	53.93	62.08±0.89	57.50	49.94	49.94±0.23
Class-aware Clean Data Finetune (Fixed)	48.02	60.52	56.41±1.82	57.35	52.34	52.34±0.38
Class-aware Clean Data Finetune (Dynamic)	56.26	54.03	63.36±1.03	57.47	51.53	51.53±0.53
+ Extra Data						
Clean Data Finetune	54.79	56.72	62.91±0.75	57.72	52.65	**52.65**±0.39
Class-aware Clean Data Finetune (Fixed)	52.27	57.38	59.55±0.98	57.67	52.52	52.52±0.21
Class-aware Clean Data Finetune (Dynamic)	55.81	55.25	**64.33**±1.22	56.84	52.41	52.41±0.14

Table 1: Results for the zero-shot relation extraction task on TACRED and Wiki80[1]. We report the average macro precision, macro recall, and micro F1 scores in three runs. The best F1 scores are marked in **bold**.

losses including **BSH** (Bootstrap Hard loss) (Reed et al., 2015), **GCE** (Generalized Cross Entropy loss) (Zhang and Sabuncu, 2018), **SCE** (Symmetric Cross Entropy loss) (Wang et al., 2019), **ER-GCE** (Entropy Regularized Generalized Cross Entropy loss) (Jin et al., 2021), and **Co-Regularization** (Zhou and Chen, 2021). The briefed introduction of those losses are shown in Appendix D.3.

Self-training (Yarowsky, 1995) first trained a classifier on a small amount of labeled data in a supervised manner, and then used this classifier to annotate more samples to train this classifier again. The initial labeled data are annotated by an off-the-shelf TE model. Only samples with a confidence score greater than a threshold are selected into the initial labeled data set. The threshold is selected on the development set.

O2U (Overfiting to Underfitting) (Huang et al., 2019) changed the model status from underfitting to overfitting repeatedly, and then used the loss to detect and remove noisy data, and finally trained the model using clean data. We directly apply it to silver standard data.

NLNL (Negative Learning for Noisy Labels) (Kim et al., 2019) first trained a classifier by using NL loss, then and selected partial data and trained them using NL as well as positive learning loss, and finally selected some clean data as labeled data and trained a classifier in a semi-supervised manner. We directly apply it to silver standard data.

DivideMix (Li et al., 2020) used a Gaussian Mixture Model (GMM) to divide the training data into a labeled set with clean samples and an unlabeled set with noisy samples, and trained the model on both the labeled and unlabeled data in a semi-supervised manner. It is a competitive method in noise labels learning area. We directly apply it to silver standard data.

All Data Finetune used all silver standard data to finetune an off-the-shelf TE model.

Clean Data Finetune used selected clean data D_{clean} to finetune a off-the-shelf TE model. We obtain D_{clean} using Eq. 2.

Class-aware Clean Data Finetune (Fixed) used selected clean data to finetune an off-the-shelf TE model. For each predicted class, we select a fixed number of samples. Higher confidence scores are selected first.

[1]Note that in a multi-class setting, for a balanced dataset such as Wiki80, macro recall is equal to micro F1.

Class-aware Clean Data Finetune (Dynamic) used selected clean data D_{clean} to finetune a off-the-shelf TE model. We obtain D_{clean} using Algorithm 1.

4.3 Result Analysis

Table 1 shows our method outperforms LaVeEntail by **12**% and **11**% on TACRED and Wiki80 respectively. In the first block of Table 1, we show results of using 1% and 5% labeled training data to finetune a TE model. Our method is comparable to the supervised method with 5% labeled data.

As shown in the second last block of Table 1, the All Data Finetune method can outperform LaVeEntail by 2%-3%, which shows that, to a certain extent, it is beneficial to finetune the pretrained model regardless of the quality of silver standard data. These findings inspire us to explore the direction of finetuning pretrained models.

The Clean Data Finetune method outperforms the All Data Finetune method by 8% and 5% on TACRED and Wiki80 respectively, which shows the clean data detection module is effective. As shown in Appendix D.1 Table 7, the noise level of clean data is lower than that of the silver data.

The Class-aware Clean Data Finetune methods outperform the Clean Data Finetune method which is class agnostic by 1% - 2%, which shows that it is better to consider class information when selecting clean data. In TACRED, the fixed method is worse than the Clean Data Finetune. Since the TACRED is skewed, if we force the distribution of training data to be uniform, it is possible to harm the performance. In Wiki80, the results are the opposite. Since Wiki80 is a balanced data, so using training data with the same distribution as test data is likely to produce good results. However, in real-world scenarios, the distribution of test data is always unknown. The dynamic method always yields a good result regardless of the distribution.

As shown in the second block of Table 1, noise-robust loss based methods cannot outperform LaVeEntail in TACRED and are comparable with LaVeEntail in Wiki80. The possible barriers to good performance are training a classifier from scratch and the high noise ratio of training data. In the third block, the semi-supervised based noisy labels learning methods are better than noise-robust loss based method. The best performance is DivideMix. In TACRED, the semi-supervised learning based methods still cannot outperform LaVeEntail. In Wiki80, the semi-supervised based methods except the self-training method can outperform or be comparable with LaVeEntail. The possible reason is TACRED has large intra-class differences compared with Wiki80. We show some instances in Appendix D.4. For Wiki80, as long as these semi-supervised based methods can identify a few clean samples for each class, they can fully utilize the semi-supervised learning assumption (Chapelle et al., 2009) (points that are close to each other are more likely to share a label) to achieve good results.

4.4 Extra Silver Standard Data

We investigate whether using extra data can further improve the performance. Extra data come from WikiFact(Goodrich et al., 2019) which is a large-scale relation extraction dataset with 923 relation types. The class distribution of WikiFact is different from TACRED and Wiki80. There is no entity type information in WikiFact data. We predict the entity types for WikiFact. Details are shown in the Appendix D.2. The statistics of WikiFact are shown in Appendix D.1.

We randomly select a fraction of the WikiFact dataset as extra data (45K instances). The whole dataset contains 3.4 million instances. By applying different clean data selection strategies (Clean Data Finetune / Class-aware Fixed / Dynamic Finetune) on silver standard data (TACRED / Wiki80), we can obtain different finetuned TE models. We do not access labels of WikiFact data. We use the corresponding finetuned TE models to annotate WikiFact data and get clean WikiFact data. We combine the clean WikiFact data and clean TACRED/Wiki80 data to finetune the TE model and infer on the relation types on TACRED and Wiki80 test set.

As shown in the last block of Table 1, compared to without extra data methods, using extra data can improve 1% - 3%, which shows using extra data can further improve the performance even if the distribution of extra data is different from the test data.

4.5 Few Shot Relation Extraction

We evaluate the effectiveness of our methods under the condition that only a few labeled examples can be provided. We first use a few labeled examples to finetune the TE model. The finetuned TE model is then used to annotate the training set as silver standard data. We randomly select k examples per

		Pr.	Rec.	F1
TACRED	NL	53.92	53.93	62.08±0.89
	O2U	59.83	43.65	58.67±0.40
	DivideMix	58.04	44.55	56.79±0.56
Wiki80	NL	57.50	49.94	49.94±0.23
	O2U	55.76	46.38	46.38±0.15
	DivideMix	55.80	48.57	48.57±0.63

Table 2: The results of using different clean data detection components (NL, O2U, and DivideMix) in our Clean Data Finetune method in the zero-shot setting.

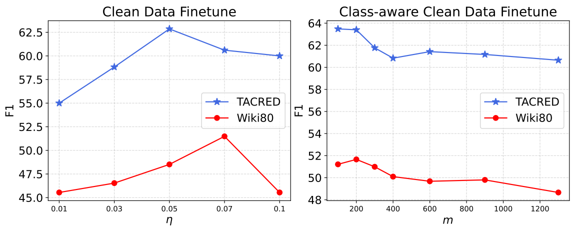


Figure 2: Results of the Clean Data Finetune method under different η and Class-aware Clean Data Finetune method under different m on 1% development set.

relation, totally $k \cdot n$ samples, where $k \in \{1, 2, 3\}$ and n is the number of relations in the dataset. As shown in the Table 3, our method outperforms LaVeEntail by 5% - 8% on TACRED and 7% - 11% on Wiki80 under different few-shot settings.

4.6 Hyper-parameter Analysis

Clean Data Proportion η. We evaluate the effects of hyper-parameter η which controls the number of clean data used to finetune TE. We select $\eta \cdot |D_{silver}|$ data to finetune the TE model. The search range for η is $[0.01, 0.03, \cdots, 0.1]$. As shown in Figure 2 (left), with the increase of parameter η, the performance of the Clean Data Finetune method increases first and then decreases. When η is too small, although D_{clean} has a low noise level, it only contains a few samples and classes, thus the model performance is barely satisfactory. When η is too large, it easily involves too many noisy samples, thus deteriorating performance. For Class-aware Clean Data Finetune (Dynamic) method, we do not tune η and use the optimal η on both datasets.

Diversity Number m. We evaluate the effects of hyper-parameter m which controls the number of samples from diverse classes. The search range for m is $[100, 200, \cdots, 1300]$. As shown in Figure 2 (right), with the increase of m, the performance of the dynamic method generally decreases. When m is too large, it easily involves too many noisy samples, thus deteriorating performance.

Shot	Method	TACRED Pr.	TACRED Rec.	TACRED F1	Wiki80 Pr.	Wiki80 Rec.	Wiki80 F1
k=0	LaVeEntail	58.02	44.73	52.18	49.09	41.16	41.16
	Ours	53.92	53.93	62.08	57.50	49.94	49.94
k=1	LaVeEntail	62.12	44.04	54.43	51.90	42.20	42.20
	Ours	52.43	52.84	62.58	57.73	50.59	50.59
k=2	LaVeEntail	61.65	44.56	55.60	54.23	43.67	43.67
	Ours	61.52	48.53	63.37	59.05	52.86	52.86
k=3	LaVeEntail	58.15	49.89	59.30	55.07	43.70	43.70
	Ours	53.49	56.08	64.10	58.10	54.80	54.80

Table 3: Results of ours (Clean Data Finetune) and LaVeEntail under different few-shot settings on TACRED and Wiki80.

4.7 Clean Data Detection Module Analysis

We also report the results of using different clean data detection algorithms in Table 2. The details of using O2U and DivideMix to detect clean data are shown in Appendix A. As shown in Table 2, NLNL consistently outperforms O2U and DivideMix. We use NLNL as our clean data detection module since it yields better performance. In Appendix D.5, we also plot the histogram of confidence scores or losses on silver standard data to visualize the ability to detect clean data for different modules.

4.8 Visualization and Implementation

Due to page limit, we put visualization of confusion matrices in Appendix D.6 and implementation details in Appendix D.7. We also provide codes in submission materials.

5 Conclusion

When a pretrained model and large-scale silver standard data exist for the zero-shot relation extraction task, we propose to first detect a small amount of clean data from silver standard data and then use them to finetune pretrained model. To further improve the performance, we propose a class-aware detection algorithm to select clean data because the number of samples per class and the number of classes are important for finetuning. The experimental results show the effectiveness of our proposed method. Finally, by using extra silver standard data of different distributions, the performance can be further improved.

Limitations

- The performance of our proposed method relies on the quality of silver standard data. Underperforming pretrained models will lead to noisy silver standard data and deteriorate the performance.

- It is time-consuming to annotate unlabeled data using an off-the-shelf TE model. It takes 3.5 hours and 2.2 hours to annotate the training set of TACRED and Wiki80 respectively. The training time of our method is 1.5 hours.

References

Görkem Algan and Ilkay Ulusoy. 2021. Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge Based System*, 215:106771.

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *Proceedings of ICML*, pages 312–321.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *Proceedings of ICML*, pages 233–242.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Proceedings of NeurIPS*, pages 5050 – 5060.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Journal of web semantics*, 7(3):154–165.

George EP Box and David R Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society*, 26(2):211–243.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. 2019. On symmetric losses for learning from corrupted labels. In *Proceedings of ICML*, pages 961–970.

Chih-Yao Chen and Cheng-Te Li. 2021. Zs-bert: Towards zero-shot relation extraction with attribute representation learning. In *Proceedings of NAACL-HLT*, pages 3470–3479.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI*, pages 1919–1925.

Ben Goodrich, Vinay Rao, Peter J. Liu, and Mohammad Saleh. 2019. Assessing the factual accuracy of generated text. In *Proceedings of KDD*.

Ankur Goswami, Akshata Bhat, Hadar Ohana, and Theodoros Rekatsinas. 2020. Unsupervised relation extraction from language models using constrained cloze completion. In *Findings of EMNLP*, pages 1263–1276.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *CoRR*.

Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W. Tsang, James T. Kwok, and Masashi Sugiyama. 2020. A survey of label-noise representation learning: Past, present and future. *arXiv preprint*.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018a. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proceedings of NeurIPS*, pages 8536–8546.

Xu Han, Tianyu Gao, Yuan Yao, Demin Ye, Zhiyuan Liu, and Maosong Sun. 2019. Opennre: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP-IJCNLP*, pages 169–174.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018b. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*, pages 4803–4809.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *Proceedings of ICLR*.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*.

Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. 2019. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of ICCV*, pages 3326–3334.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of ICML*, pages 2304–2313.

Lifeng Jin, Linfeng Song, Kun Xu, and Dong Yu. 2021. Instance-adaptive training with noise-robust losses against noisy labels. In *Proceedings of EMNLP*, pages 5647–5663.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the ACL*, pages 178–181.

Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. 2019. Nlnl: Negative learning for noisy labels. In *Proceedings of ICCV*, pages 101–110.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of CoNLL*, pages 333–342.

Junnan Li, Richard Socher, and Steven CH Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proceedings of ICLR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Keming Lu, I Hsu, Wenxuan Zhou, Mingyu Derek Ma, Muhao Chen, et al. 2022. Summarization as indirect supervision for relation extraction. *arXiv preprint arXiv:2205.09837*.

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. In *Proceedings of ACL*, pages 390–395.

Yueming Lyu and Ivor W Tsang. 2019. Curriculum loss: Robust learning and generalization against label corruption. In *Proceedings of ICLR*.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah M. Erfani, and James Bailey. 2020. Normalized loss functions for deep learning with noisy labels. In *Proceedings of ICML*, volume 119, pages 6543–6553.

Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". In *Proceedings of NeurIPS*, pages 960–970.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. 2020. Can gradient clipping mitigate label noise? In *Proceedings of ICLR*.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the FEVER*, pages 72–78.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, pages 1345–1359.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *Proceedings of ICLR Workshop*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *Proceedings of ICML*, pages 4334–4343.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero-and few-shot relation extraction. In *Proceedings of EMNLP*, pages 1199–1212.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Proceedings of NeurIPS*, pages 1917–1928.

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

Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. 2019. Combating label noise in deep learning using abstention. *arXiv preprint arXiv:1905.10964*.

Thy Thy Tran, Phong Le, and Sophia Ananiadou. 2020. Revisiting unsupervised relation extraction. In *Proceedings of ACL*, pages 7498–7505.

Thy Thy Tran, Phong Le, and Sophia Ananiadou. 2021. One-shot to weakly-supervised relation classification using language models. In *Proceedings of AKBC*.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of ICCV*, pages 322–330.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of CIKM*, pages 2361–2364.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: deep contextualized entity representations with entity-aware self-attention. In *Proceedings of EMNLP*, pages 6442–6454.

Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Kwok. 2019. Searching to exploit memorization effect in learning from corrupted labels. *arXiv preprint arXiv:1911.02377*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the ACL*, pages 189–196.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption? In *Proceedings of ICML*, pages 7164–7173.

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

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017a. Understanding deep learning requires rethinking generalization. In *Proceedings of ICLR*.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

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

Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of NeurIPS*, pages 8792–8802.

Zexuan Zhong and Danqi Chen. 2020. A frustratingly easy approach for entity and relation extraction. In *Proceedings of NAACL*, pages 50–61.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of NAACL*, pages 50–61.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the ACL*, pages 427–434.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of EMNLP*, pages 5381–5392.

Algorithm 3 Clean Data Detection using **NL**

Input: silver data D_{silver} , total epoch T, number of complementary labels n_c.

1: **for** $t \leftarrow 1$ **to** T **do**
2: Generate n_r complementary labels for each input label.
3: Minimize Eq. 1.
4: Calculate the confidence score of each sample in D_{silver}.
5: **end for**
6: Obtain clean data set D_{clean} by Eq. 2.

Ouput: clean data set D_{clean}.

A Clean Data Detection Components

This section presents how we adapt three noisy labels training algorithms, i.e., NLNL, O2U and DivideMix, to clean data detection modules. These modules use the same classifier, as shown in B.

A.1 Negative Learning (NL) Method

In NL, the complementary label is randomly selected from the label space excluding the input label (possibly noisy). As shown in Eq.1, the negative learning loss provides the model with the complementary label(s), telling what is incorrect and optimizing the output probability corresponding to the complementary label to be close to zero.

The whole clean data detection process by using NL is presented in Algorithm 3. In our experiment, n_c is equaled to the number of relation types.

A.2 Clean label detection by Overfitting to Underfitting (O2U)

O2U (Huang et al., 2019) exploited the loss of each sample to detect clean data in a special setting where the status of the model transfers from overfitting to underfitting cyclically. The status change is implemented by changing the learning rate cyclically. The intuition of O2U is the memorization effect (Arpit et al., 2017) which is stated as although deep networks can memorize noise data, they tend to learn simple patterns first. The clean samples can be quickly learned by deep networks, hence their losses maintain small once they are learned. Clean samples tend to have smaller losses in the whole training procedure. The hard samples and noisy samples are not memorized until it reaches the overfitting stage, hence their losses are large at the underfitting stage. Hence, by transferring the status from underfitting to overfitting and collecting the statistics of losses, it is possible to detect clean data. In O2U-based clean data detection module, a classifier (see Appendix B) is trained in two phases, i.e., pretraining and cyclical training.

Pretraining. The network is pre-trained on the silver standard data with a constant learning rate.

Cyclical Training. A cyclical learning rate is applied to train the classifier. During this process, the learning rate is changed from maximum to minimum repeatedly. In a training epoch, suppose the maximum learning rate is r_{max} and the minimum learning rate is r_{min}, a linear learning rate decrease function $r(t)$ is adopted to adjust the learning rate as follows,

$$r(t) = r_{max} - \frac{t}{E} \times (r_{max} - r_{min}), \quad (3)$$

where t is the t-th epoch of a cyclical training round, E is the total number of epochs in each cyclical round, and $r(t)$ is the learning rate applied at epoch t in a cyclical training round.

After training the classifier in the cyclical setting, we sort silver standard data by their summation losses of several rounds of underfitting to overfitting procedure. We select a fixed proportion η of the silver data with smaller loss as clean data. Given that $\mathcal{L}(D_s)$ is total loss of each sample in the selected data set D_s , the clean data set D_{clean} are selected as follows,

$$D_{clean} = \arg\min_{D_s:|D_s|=\eta\cdot|D_{silver}|}\mathcal{L}(D_s). \quad (4)$$

A.3 Clean label detection by DivideMix

DivideMix (Li et al., 2020) used a Gaussian Mixture Model to dynamically divide the noisy data into a labeled set with clean samples and an unlabeled set with noisy samples. DivideMix trained a classifier on the labeled set as well as unlabeled set in a semi-supervised manner. Two models are simultaneously trained for co-dividing and co-guessing to reduce confirmation bias. DivideMix can efficiently maintain small losses for clean samples but keep large losses for noisy samples.

When we apply DivideMix to clean data detection, we have modifications in generating data augmentation and applying MixMatch (Berthelot et al., 2019) on the augmented labeled data and augmented unlabeled data. For data augmentation, we randomly replace the subject entity or object entity with other entities in the dataset with the same entity type. When applying MixMacth, DivideMix linearly interpolated inputs of random samples. However, text cannot be directly interpolated, while interpolation is straightforward for image pixels. Thus, as proposed by (Guo et al., 2019), we interpolate the text embedding of random samples.

We use the cross-entropy loss to detect clean data. We select D_{clean} using Eq. 4

B Classifier

We use the relation model in the PURE system (Zhong and Chen, 2020) as the relation classifier. The input text is inserted with text markers to highlight the subject and object and their positions. Given a input text x, the subject span SUBJECT, the object span OBJECT, the subject entity type t_{subj}, and object entity type t_{obj}, Text markers are defined as $\langle$S:$t_{subj}\rangle$, $\langle$/S:$t_{subj}\rangle$, $\langle$O:$t_{obj}\rangle$, and $\langle$/O:$t_{obj}\rangle$. We insert them into the input text before and after the subject and object span. Let $\hat{x}$ denote the modified sentence with text markers inserted:

$$\widehat{x} = \ldots \langle \text{S:}e_{subj} \rangle \; \textsc{Subject} \; \langle \text{/S:}e_{subj} \rangle \\ \ldots \langle \text{O:}e_{obj} \rangle \; \textsc{Object} \; \langle \text{/O:}e_{obj} \rangle \ldots .$$

We concatenate the hidden state embeddings of the final layer in BERT (Devlin et al., 2019) at the subject start marker position and object start marker position as the contextual representations of $\hat{x}$,

$$\mathbf{h}_r(\hat{x}) = [\widehat{\mathbf{x}}_{\widehat{\text{START}}_{subj}}; \widehat{\mathbf{x}}_{\widehat{\text{START}}_{obj}}],$$

where $\widehat{\text{START}}_{subj}$ and $\widehat{\text{START}}_{obj}$ are the position indices of $\langle$S:$e_{subj}\rangle$ and $\langle$O:$e_{obj}\rangle$ in $\hat{x}$, and $\widehat{\mathbf{x}}$ is a list of hidden state embeddings of all words in $\widehat{x}$. Finally, the representation $\mathbf{h}_r(\hat{x})$ will be fed into a feedforward network to obtain the probability distribution of the relation type.

C Premise-Hypothesis Pairs Construction

We have different generation strategies when generating premise-hypothesis pairs for the positive relation and the negative relation.

1. **Positive Relation.** The positive relation means there is a relation between subject and object. For the positive relation, a

contradiction hypothesis is generated using no_relation verbalization template "{subj} and {obj} are not related", a **neutral** hypothesis is generated by randomly select a template that does not describe the ground truth relation, and a **entailment** hypothesis is generated with the templates that describes the ground truth relation.

2. **Negative Relation.** The negative relation means subject and object are not related. For the negative relation, a **contradiction** hypothesis is generated using has-relation template `There is a relation between {subj} and {obj}` , a **neutral** hypothesis is generated by randomly select a positive relation template, and a **entailment** hypothesis is generated by the `no-relation` verbalization template mentioned above.

D Experiments

D.1 Dataset Statistics

The dataset statistics are shown in Table 4. TACRED consists of 42 relation labels including `no_relation` and its relation distribution is skewed. TACRED provides entity type information. Wiki80 contains 80 relation labels and its relation distribution is uniform. Since the test set of Wiki80 is not provided, we used the development set for testing. We take 20% of the training data as the development set.

The noise ratios of silver standard data are 16.67% and 58.88% on TACRED and Wiki80 respectively. In TACRED, `no_relation` is a major class, accounting for 85.75% of whole data. We also provide the noise ratio of only positive relation data on TACRED, i.e., 42.01%.

The statistics of WikiFact are shown in Table 4. We randomly select 45K instances from the whole WikiFact dataset as the extra data candidate pool as the whole dataset is very large. After applying different clean data detection algorithms, we finally select 800/1500 samples in WikiFact as extra data for TACRED/Wiki80 datasets.

Dataset	Relation Types	Entity Types	Distribution	Instances		
				Train	Dev	Test
TACRED	42	17	Skewed	68124	22631	15509
Wiki80	80	29	Uniform	40320	10080	5600
WikiFact	923	126	Skewed	2236367	276967	279699

Table 4: The statistics of TACRED and Wiki80 datasets. Each instance is a sentence with two entities and their entity types.

D.2 Entity Types Generation

To obtain entity types in Wiki80, we finetune a pretrained language model (Devlin et al., 2019) using prompt learning paradigm on DBpedia dataset (Bizer et al., 2009) to predict the entity type. The prompt template is designed as "{entity} is a `[MASK]`". DBpedia describes more than 2.6 million entities. Each text describes one entity and has a class label. We treat the class label as the entity type. At the inference phase, the prediction for the `[MASK]` token is used as the entity type for the entity on Wiki80.

D.3 Compared Methods

We briefly introduce baselines using different losses.

CE (Cross Entropy loss) has been widely used as optimization loss. We consider it as a baseline.

BSH (Bootstrap Hard loss) (Reed et al., 2015) consider neural network predictions are possible to be correct. BSH modified the CE loss and used a weighted combination of predicted and input labels (possibly noisy) as the correct labels. Hard labels are used as they have better performance. The hard label is the one-hot vector after taking $\arg\max$ operation on the prediction distribution vector.

GCE (Generalized Cross Entropy loss) (Zhang and Sabuncu, 2018) combined the CE loss and mean absolute error (MAE) loss via the negative Box-Cox transformation (Box and Cox, 1964). The MAE loss is proved to be noise-robust.

SCE (Symmetric Cross Entropy loss) (Wang et al., 2019) combined the CE loss and a noise-robust counterpart Reverse Cross Entropy (RCE) to deal with a weakness of CE. CE tends to overfit noisy labels on "easy" classes and underfit on "hard" classes.

ER-GCE (Entropy Regularized Generalized Cross Entropy loss) (Jin et al., 2021) improved GCE by interpolating the CE loss with an entropy regularizer. It has a tighter bound than GCE.

Co-Regularization (Zhou and Chen, 2021) trained several classifiers with the same structures but different parameter initialization, and regularized all models to generate similar predictions rather than overfit the input (possibly noisy) labels.

D.4 Dataset Samples

As shown in Table 1, In TACRED, the semi-supervised based methods still cannot outperform LaVeEntail. In Wiki80, the semi-supervised based methods except the self-training method can outperform or be comparable with LaVeEntail. The possible reason is that although the noise ratio of TACRED is lower than that of Wiki80, TACRED is a more challenging dataset than Wiki80. Table 5 shows that the sentences in TACRED have a more complex context, while texts are straightforward in Wiki80. Also, TACRED has large intra-class differences compared with Wiki80. Table 6 shows that instances in TACRED have large intra-class differences while instances in Wiki80 have similar structures. For Wiki80, as long as these semi-supervised based methods can identify a few clean samples for each class, they can fully utilize the semi-supervised learning assumption (Chapelle et al., 2009) (i.e., points that are close to each other are more likely to share a label) to achieve good results. But in TACRED, data that share a label might be different in the input space.

D.5 Clean Data Detection Module Analysis

We also plot the histogram of confidence scores or losses on silver standard data. As shown in the Figure 3, NLNL has a better ability to distinguish clean and noisy data. The clean and noisy data are well separated by confidence scores.

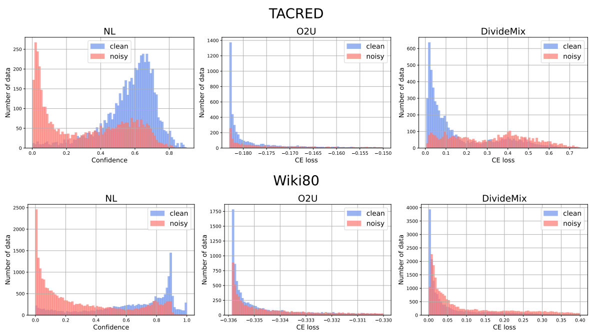


Figure 3: Histogram showing the confidence scores or losses distribution of silver standard data on TACRED and Wiki80, Blue indicates clean data, whereas red indicates noisy data.

D.6 Visualization

We show confusion matrices of the test data on both datasets in Figure 4. As shown in the figures, most of the relations are classified correctly in Clean Data Finetune method.

D.7 Implementation Details

All of our experiments are performed on a single NVIDIA RTX 3090 GPU. Both the Clean Data Finetune method and Class-aware Clean Data Finetune method need 1.5 hours for training. In zero-shot and few-shot settings, the pre-trained TE model we used is microsoft/deberta-v2-xlarge-mnli (He et al., 2021). We also report the performance of Clean Data Finetune Method using microsoft/deberta-v2-xxlarge-mnli (He et al., 2021) in Table 8. The results show that xlarge model can outperform or be comparable with xxlarge model. We use the light-weighted model to save training time. For all the clean data detection algorithms, we adopt bert-base-uncased (Devlin et al., 2019) as the backbone model of the classifier. We use AdamW optimizer with weight decay 1e-2.

Hyper-parameters Settings. As shown in Figure 2 (left), for the Clean Data Finetune method, $\eta = 0.05$ for TACRED and $\eta = 0.07$ for Wiki80 have the best performance on 1% development set. Thus we set $\eta = 0.05$ for TACRED and $\eta = 0.07$ for Wiki80 in the method of Clean Data Finetune. For Class-aware Clean Data Finetune (Dynamic) method, we do not tune η and we use the optimal η as the Clean Data Finetune method uses on both datasets.

As shown in Figure 2 (right), we use the expansion ratios $\delta = 100$ for TACRED and $\delta = 200$ for Wiki80 in the method of Class-aware Clean Data Finetune (Dynamic).

Clean Data Detection Module Settings. 1) For O2U: In the pre-training step, the constant learning rate is 5e-6. In cyclical training, the cyclical learning rate is linearly adjusted from 5e-6 to 1e-7 in a cycle round. The cycle length is five epochs in a cycle round, and we adopt one cycle round. 2) For NLNL: The learning rate is 4e-7. The number of complementary labels on a single input label is the same as the classification number. We run ten epochs. 3) For DivideMix: The sharpening temperature T is 0.5, the parameter for Beta is 4, the weight for unsupervised loss is 25, and clean probability threshold is 0.5. The learning rate is 4e-7.

Finetuning Settings. We use 80% of silver standard data to finetune TE model, while the remaining 20% data serves as the development set. The learning rate is warmed up linearly to 4e-7 and then it decreases following the values of the cosine function between 4e-7 to zero.

Relation	Instance in TACRED	Instance in Wiki80
religion	Iran 's supreme leader Ayatollah Ali Khamenei on Wednesday condemned Israel 's works near the flashpoint mosque compound in Jerusalem, urging Muslim countries to make the Jewish state regret the move.	Angelo Scola (born 7 November 1941) is an Italian Cardinal of the Catholic Church , philosopher and theologian.
	Though not a household name, Wildmon has considerable clout; his group has a vast mailing list and a proven ability to mobilize Christian conservatives by the hundreds of thousands.	Vincenzo Maria Sarnelli (5 April 1835–7 January 1898) was an Italian Catholic archbishop.
	Carson 's grandmother raised him in a Baptist church and enrolled him at an inner-city Catholic school, where he entertained the idea of becoming a priest.	Giovanni Arcimboldi (died 1488) (called the Cardinal of Novara or the Cardinal of Milan) was an Italian Roman Catholic bishop and cardinal.
	Chalabi, Mahdi and Solagh all represent the Iraq National Alliance, the main Shiite religious list .	Vazgen I , head of the Armenian Apostolic Church , sent Pope Paul VI a letter mourning Agagianian 's death.
	Note: My thinking he is the worst has little to do with him being Muslim , since I think the other Muslim Congressman, Andre Carson is a pretty good guy.	There are Sámi Christians who believe in Laestadianism that use Ipmil for God.

Table 5: Some instances on TACRED and Wiki80. The subject is marked in blue, and the object is marked in red.

Instances of "per:cities_of_residence" on **TACRED**	Instances of "taxon rank" on **Wiki80**
On the July morning in 1944 when she boarded a Greyhound bus in Gloucester bound for Baltimore , Kirkaldy was not thinking about tackling racial segregation.	Culeolus is a genus of ascidian tunicates in the family Pyuridae .
She stayed at her home in Wasilla , located 40 miles to the north, but was expected in her office on Friday, spokesman Bill McAllister said.	It is the only recognized extant genus in the family Equidae .
In Vienna , Austria, on Monday , International Atomic Energy Agency chief Mohamed ElBaradei lamented a " stalemate " in efforts to begin talks over Iran 's nuclear program.	Polyozellus is a fungal genus in the family Thelephoraceae , a grouping of mushrooms known collectively as the leathery earthfans.
At her death , she was assistant clinical professor emeritus of psychiatry at Albert Einstein College of Medicine of Yeshiva University in the Bronx .	Megalaria is a genus of lichenized fungi in the family Megalariaceae .
His death was confirmed by Hazel McCallion , mayor of Mississauga , Ontario , the Toronto suburb where Peterson lived.	Leucothoe is a genus of amphipods in the family Leucothoidae .

Table 6: Some instance on TACRED and Wiki80. The subject is marked in blue, and the object is marked in red.

	Silver data	Clean data
TACRED	57.50%	80.42%
Wiki80	48.96%	89.42%

Table 7: The accuracies of silver data and clean data. Higher accuracy means lower noise level.

Model	TACRED			Wiki80		
	Pr.	Rec.	F1	Pr.	Rec.	F1
microsoft/deberta-v2-xlarge-mnli	53.92	53.93	62.08$\pm$0.89	57.50	49.94	49.94$\pm$0.23
microsoft/deberta-v2-xxlarge-mmli	48.62	57.23	59.12$\pm$2.29	55.44	50.04	50.04$\pm$1.16

Table 8: F1 scores of Clean Data Finetune method on TACRED and Wiki80 using different sizes of textual entailment model.

D.8 Verbalization Templates

D.8.1 TACRED

We show verbalization templates of all relation types on TACRED in Table 9.

D.8.2 Wiki80

We show verbalization templates of all relation types on Wiki80 in Table 10.

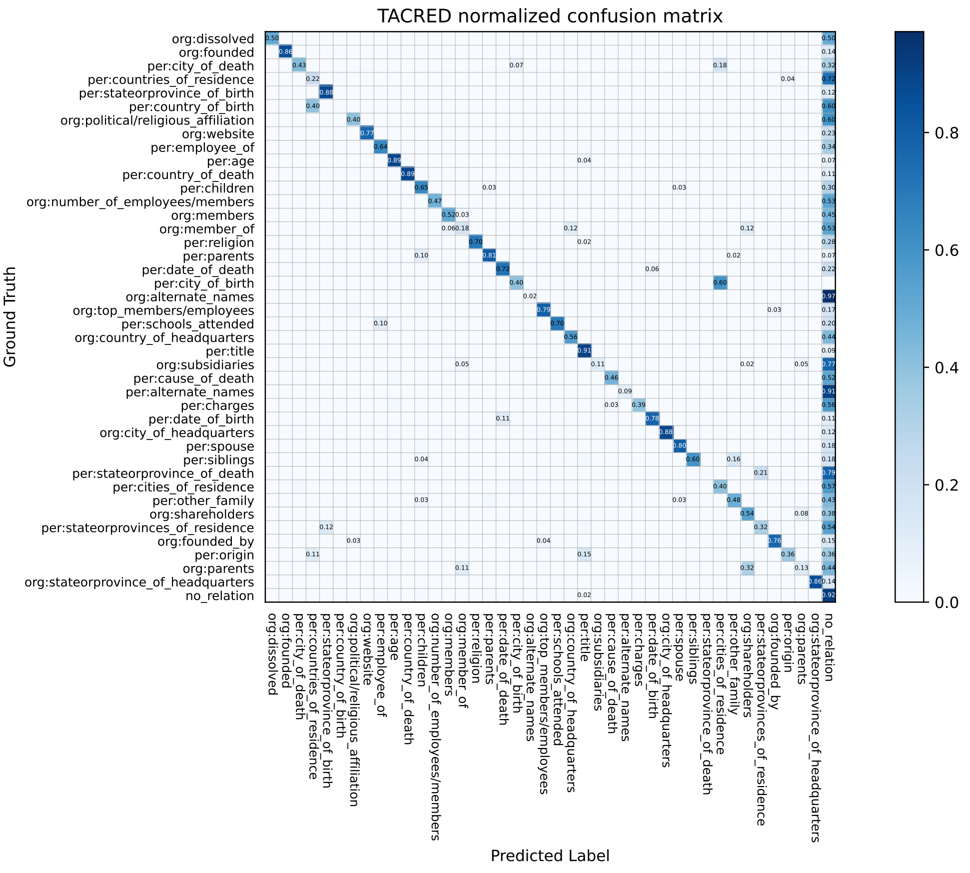
D.9 Entity Type Constraints

D.9.1 TACRED

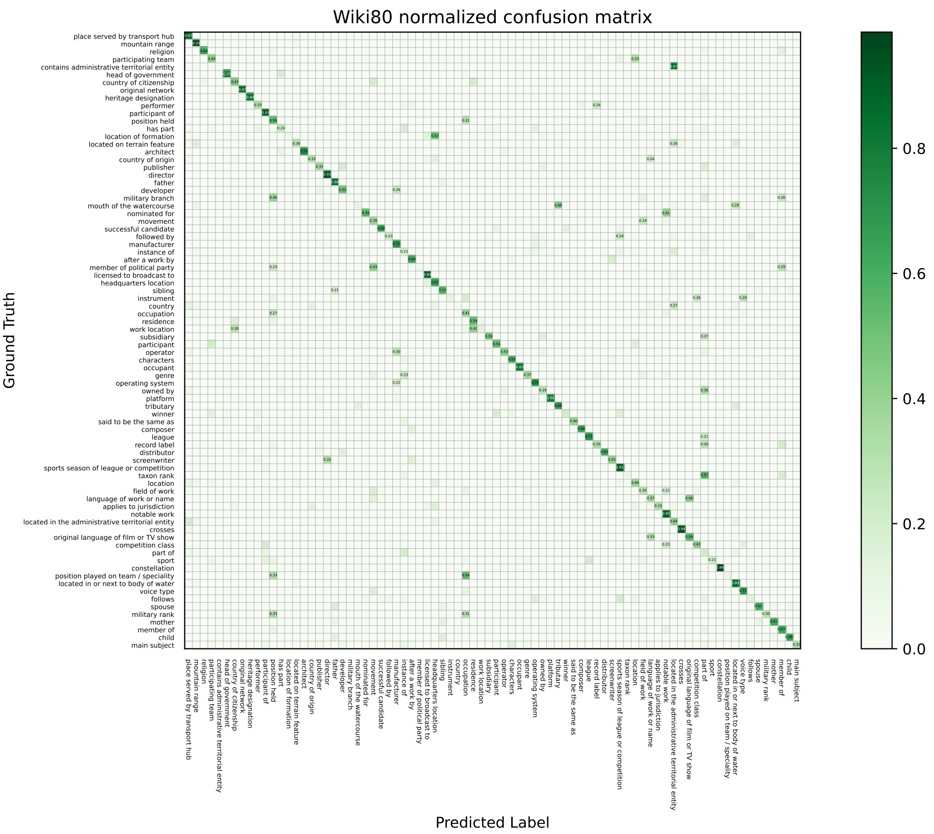
We present the entity type constraints of relation types on TACRED in Table 11. The constraints are different from that of LaVeEntail. We delete the constraints which leak the information of the ground truth. For example, there is only one relation type that has the constraint where the subject entity type is PERSON and the object entity type is TITLE. The sentence that satisfies this constraint has a very large probability to be inferred as `per:title` relation because other relations are ruled out. (Tran et al., 2020) showed that entity types are a strong inductive bias. But in LaVeEntail, the inductive bias is not learned by the algorithm itself but by manually designed type constraints. It leads to artificially inflated performance, so we deleted those type constraints.

D.9.2 Wiki80

We present the entity type constraints of relation types on Wiki80 in Table 12.



(a) TACRED



(b) Wiki80

Figure 4: Confusion matrix (rowise normalized) of Clean Data Finetune on the test set of TACRED and Wiki80.

Relation	Template	Relation	Template
no_relation	{subj} and {obj} are not related	per:alternate_names	{subj} is also known as {obj}
per:date_of_birth	{subj} 's birthday is on {obj} {subj} was born in {obj}	per:age	{subj} is {obj} years old
per:country_of_birth	{subj} was born in {obj}	per:stateorprovince_of_birth	{subj} was born in {obj}
per:city_of_birth	{subj} was born in {obj}	per:origin	{obj} is the nationality of {subj}
per:date_of_death	{subj} died in {obj}	per:country_of_death	{subj} died in {obj}
per:stateorprovince_of_death	{subj} died in {obj}	per:city_of_death	{subj} died in {obj}
per:cause_of_death	{obj} is the cause of {subj}'s death	per:countries_of_residence	{subj} lives in {obj} {subj} has a legal order to stay in {obj}
per:stateorprovinces_of_residence	{subj} lives in {obj} {subj} has a legal order to stay in {obj}	per:cities_of_residence	{subj} lives in {obj} {subj} has a legal order to stay in {obj}
per:schools_attended	{subj} studied in {obj} {subj} graduated from {obj}	per:title	{subj} is a {obj}
per:employee_of	{subj} is member of {obj} {subj} is an employee of {obj}	per:religion	{subj} belongs to {obj} religion {obj} is the religion of {subj} {subj} believe in {obj}
per:spouse	{subj} is the spouse of {obj} {subj} is the wife of {obj} {subj} is the husband of {obj}	per:parents	{obj} is the parent of {subj} {obj} is the mother of {subj} {obj} is the father of {subj} {subj} is the son of {obj} {subj} is the daughter of {obj}
per:children	{subj} is the parent of {obj} {subj} is the mother of {obj} {subj} is the father of {obj} {obj} is the son of {subj} {obj} is the daughter of {subj}	per:siblings	{subj} and {obj} are siblings {subj} is brother of {obj} {subj} is sister of {obj}
per:other_family	{subj} and {obj} are family {subj} is a brother in law of {obj} {subj} is a sister in law of {obj} {subj} is the cousin of {obj} {subj} is the uncle of {obj} {subj} is the aunt of {obj} {subj} is the grandparent of {obj} {subj} is the grandmother of {obj} {subj} is the grandson of {obj} {subj} is the granddaughter of {obj}	per:charges	{subj} was convicted of {obj} {obj} are the charges of {subj}
org:alternate_names	{subj} is also known as {obj}	org:political/religious_affiliation	{subj} has political affiliation with {obj} {subj} has religious affiliation with {obj}
org:top_members/employees	{obj} is a high level member of {subj} {obj} is chairman of {subj} {obj} is president of {subj} {obj} is director of {subj}	org:number_of_employees/members	{subj} employs nearly {obj} people {subj} has about {obj} employees
org:members	{obj} is member of {subj} {obj} joined {subj}	org:member_of	{subj} is member of {obj} {subj} joined {obj}
org:subsidiaries	{obj} is a subsidiary of {subj} {obj} is a branch of {subj}	org:parents	{subj} is a subsidiary of {obj} {subj} is a branch of {obj}
org:founded_by	{subj} was founded by {obj} {obj} founded {subj}	org:founded	{subj} was founded in {obj} {subj} was formed in {obj}
org:dissolved	{subj} existed until {obj} {subj} disbanded in {obj} {subj} dissolved in {obj}	org:country_of_headquarters	{subj} has its headquarters in {obj} {subj} is located in {obj}
org:stateorprovince_of_headquarters	{subj} has its headquarters in {obj} {subj} is located in {obj}	org:city_of_headquarters	{subj} has its headquarters in {obj} {subj} is located in {obj}
org:shareholders	{obj} holds shares in {subj}	org:website	{obj} is the URL of {subj} {obj} is the website of {subj}

Table 9: Verbalization templates on TACRED.

Relation	Template	Relation	Template
place served by transport hub	{subj} is the place that served by a transport hub in {obj}.	mountain range	{subj} mountain range is in the {obj}. {subj} mountain range is on the {obj}. {subj} mountain range is part of the {obj}.
religion	{obj} is {subj}'s religion.	participating team	{obj} team participated in {subj}. {obj} rival participated in {subj}.
contains administrative territorial entity	{obj} place is the terrioty of {subj}.	head of government	{obj} is the government head of {subj}.
country of citizenship	{obj} country does {subj} has a citizenship of.	original network	{obj} is the original network of {subj}.
heritage designation	{subj} heritage designation is listed on the {obj}.	performer	{obj} are performers of " {subj} ".
participant of	{subj} participated in {obj}. {obj} event did {subj} participate in.	position held	{obj} position is held by {subj}.
has part	{subj} does {obj} belong to.	location of formation	{obj} is {subj} formed.
located on terrain feature	{obj} is the terrain feature {subj} located in.	architect	{obj} is the architect of {subj}.
country of origin	{obj} is {subj}'s country of origin.	publisher	{obj} is the publisher of " {subj} ".
director	{obj} is the director of " {subj} ".	father	{obj} is {subj}'s father.
developer	{obj} is the developer of " {subj} ".	military branch	{obj} military branch does {subj} work for.
mouth of the watercourse	{subj} is the mouth of the watercourse {obj}.	nominated for	{obj} are " {subj} " nominated for. {subj} is the nominee of {obj}.
movement	{obj} is movement of {subj}.	successful candidate	{obj} is the successful candidate of {subj}.
followed by	{subj} is before " {obj} ". {subj} is followed by " {obj} ".	manufacturer	{obj} is the manufacturer of {subj}.
instance of	{subj} is an instance of {obj}. {obj} is the {subj}.	after a work by	{subj} is created by " {obj} ". {subj} is based on " {obj} ".
member of political party	{obj} political party does {subj} belong to.	licensed to broadcast to	{subj} is licensed to {obj}.
headquarters location	{obj} is the headquarter of {subj}.	sibling	{obj} are {subj}'siblings. {subj} are {obj}'s siblings.
instrument	{obj} instruments does {subj} play.	country	{obj} country does {subj} belong to.
occupation	{obj} is {subj}'s occupation.	residence	{obj} does {subj} live in.
work location	{obj} does {subj} work in.	subsidiary	{obj} organization is the subsidiary of {subj}.
participant	{obj} are participants of {subj}.	operator	{obj} are operators of {subj}.
characters	{obj} are the characters of {subj}.	occupant	{obj} teams are occupants of {subj}.
genre	{obj} is the genre of " {subj} ".	operating system	{obj} are operating systems of {subj}.
owned by	{obj} own {subj}.	platform	{subj} are platforms of {obj}.
tributary	{obj} are tributaries of {subj}.	winner	{obj} are the winners of {subj}.
said to be the same as	{obj} are said to be the same as {subj}.	composer	{obj} are composers of {subj}.
league	{obj} is the league of {subj}.	record label	{obj} is the record label of {subj}.
distributor	{obj} are distributors of {subj}.	screenwriter	{obj} are screenwriters of {subj}.
sports season of league or competition	{subj} seasons of {obj} are mentioned.	taxon rank	{obj} is taxon rank of {subj}.
location	{obj} did {subj} held.	field of work	{obj} are {subj}'s fields of work.
language of work or name	{obj} is the language of the work " {subj} ". {obj} is the language of the name " {subj} ".	applies to jurisdiction	{obj} is the jurisdiction of {subj} applied to.
notable work	{obj} are notable works of {subj}.	located in the administrative territorial entity	{obj} is the administrative territorial entity {subj} located in.
crosses	{subj} cross {obj}.	original language of film or TV show	{obj} is the original language of the film " {subj} ". {obj} is the original language of the TV show " {subj} ".
competition class	{obj} is the competition class of {subj}.	part of	{subj} is a part of {obj}.
sport	{obj} sports does {subj} play.	constellation	{subj} are in the constellation of " {obj} ".
position played on team / speciality	{obj} position does {subj} play on the team.	located in or next to body of water	{obj} body of water is {subj} located in.
voice type	{obj} is the voice type of {subj}.	follows	{subj} is after " {obj} ". {subj} follows " {obj} ".
spouse	{obj} is {subj}'s spouse.	military rank	{obj} is the military rank of {subj}.
mother	{obj} is {subj}'s mother.	member of	{subj} is a member of {obj}.
child	{obj} are {subj}'s children.	main subject	{obj} is the main subject of " {subj} ".

Table 10: Verbalization templates on Wiki80.

Relation	Constraint	Relation	Constraint
per:alternate_names	PERSON:PERSON	per:date_of_birth	PERSON:DATE
per:age	PERSON:TITLE , PERSON:CITY , PERSON:STATE_OR_PROVINCE PERSON:ORGANIZATION , PERSON:RELIGION , PERSON:DURATION PERSON:NUMBER , PERSON:LOCATION , PERSON:DATE PERSON:NATIONALITY , PERSON:IDEOLOGY , PERSON:PERSON PERSON:MISC , PERSON:COUNTRY , PERSON:CAUSE_OF_DEATH PERSON:URL , PERSON:CRIMINAL_CHARGE	per:country_of_birth	PERSON:COUNTRY
per:stateorprovince_of_birth	PERSON:STATE_OR_PROVINCE	per:city_of_birth	PERSON:CITY
per:origin	PERSON:NATIONALITY , PERSON:COUNTRY , PERSON:LOCATION	per:date_of_death	PERSON:DATE
per:country_of_death	PERSON:COUNTRY	per:stateorprovince_of_death	PERSON:STATE_OR_PROVICE
per:city_of_death	PERSON:CITY	per:cause_of_death	PERSON:TITLE , PERSON:CITY , PERSON:STATE_OR_PROVINCE PERSON:ORGANIZATION , PERSON:RELIGION , PERSON:DURATION PERSON:NUMBER , PERSON:LOCATION , PERSON:DATE PERSON:NATIONALITY , PERSON:IDEOLOGY , PERSON:PERSON PERSON:MISC , PERSON:COUNTRY , PERSON:CAUSE_OF_DEATH PERSON:URL , PERSON:CRIMINAL_CHARGE
per:countries_of_residence	PERSON:COUNTRY , PERSON:NATIONALITY	per:stateorprovinces_of_residence	PERSON:STATE_OR_PROVINCE
per:cities_of_residence	PERSON:CITY	per:schools_attended	PERSON:ORGANIZATION
per:title	PERSON:TITLE , PERSON:CITY , PERSON:STATE_OR_PROVINCE PERSON:ORGANIZATION , PERSON:RELIGION , PERSON:DURATION PERSON:NUMBER , PERSON:LOCATION , PERSON:DATE PERSON:NATIONALITY , PERSON:IDEOLOGY , PERSON:PERSON PERSON:MISC , PERSON:COUNTRY , PERSON:CAUSE_OF_DEATH PERSON:URL , PERSON:CRIMINAL_CHARGE	per:employee_of	PERSON:ORGANIZATION
per:religion	PERSON:TITLE , PERSON:CITY , PERSON:STATE_OR_PROVINCE PERSON:ORGANIZATION , PERSON:RELIGION , PERSON:DURATION PERSON:NUMBER , PERSON:LOCATION , PERSON:DATE PERSON:NATIONALITY , PERSON:IDEOLOGY , PERSON:PERSON PERSON:MISC , PERSON:COUNTRY , PERSON:CAUSE_OF_DEATH PERSON:URL , PERSON:CRIMINAL_CHARGE	per:spouse	PERSON:PERSON
per:parents	PERSON:PERSON	per:children	PERSON:PERSON
per:siblings	PERSON:PERSON	per:other_family	PERSON:PERSON
per:charges	PERSON:TITLE , PERSON:CITY , PERSON:STATE_OR_PROVINCE PERSON:ORGANIZATION , PERSON:RELIGION , PERSON:DURATION PERSON:NUMBER , PERSON:LOCATION , PERSON:DATE PERSON:NATIONALITY , PERSON:IDEOLOGY , PERSON:PERSON PERSON:MISC , PERSON:COUNTRY , PERSON:CAUSE_OF_DEATH PERSON:URL , PERSON:CRIMINAL_CHARGE	org:alternate_names	ORGANIZATION:ORGANIZATION
org:political/religious_affiliation	ORGANIZATION:TITLE , ORGANIZATION:CITY , ORGANIZATION:STATE_OR_PROVINCE ORGANIZATION:ORGANIZATION , ORGANIZATION:RELIGION , ORGANIZATION:DURATION ORGANIZATION:NUMBER , ORGANIZATION:LOCATION , ORGANIZATION:DATE ORGANIZATION:NATIONALITY , ORGANIZATION:IDEOLOGY , ORGANIZATION:PERSON ORGANIZATION:MISC , ORGANIZATION:COUNTRY , ORGANIZATION:CAUSE_OF_DEATH ORGANIZATION:URL , ORGANIZATION:CRIMINAL_CHARGE	org:top_members/employees	ORGANIZATION:PERSON
org:number_of_employees/members	ORGANIZATION:TITLE , ORGANIZATION:CITY , ORGANIZATION:STATE_OR_PROVINCE ORGANIZATION:ORGANIZATION , ORGANIZATION:RELIGION , ORGANIZATION:DURATION ORGANIZATION:NUMBER , ORGANIZATION:LOCATION , ORGANIZATION:DATE ORGANIZATION:NATIONALITY , ORGANIZATION:IDEOLOGY , ORGANIZATION:PERSON ORGANIZATION:MISC , ORGANIZATION:COUNTRY , ORGANIZATION:CAUSE_OF_DEATH ORGANIZATION:URL , ORGANIZATION:CRIMINAL_CHARGE	org:members	ORGANIZATION:ORGANIZATION
org:member_of	ORGANIZATION:ORGANIZATION , ORGANIZATION:COUNTRY , ORGANIZATION:LOCATION ORGANIZATION:STATE_OR_PROVINCE	org:subsidiaries	ORGANIZATION:ORGANIZATION
org:parents	ORGANIZATION:ORGANIZATION	org:founded_by	ORGANIZATION:PERSON
org:founded	ORGANIZATION:DATE	org:dissolved	ORGANIZATION:DATE
org:country_of_headquarters	ORGANIZATION:COUNTRY	org:stateorprovince_of_headquarters	ORGANIZATION:STATE_OR_PROVINCE
org:city_of_headquarters	ORGANIZATION:CITY	org:shareholders	ORGANIZATION:PERSON , ORGANIZATION:ORGANIZATION
org:website	ORGANIZATION:TITLE , ORGANIZATION:CITY , ORGANIZATION:STATE_OR_PROVINCE ORGANIZATION:ORGANIZATION , ORGANIZATION:RELIGION , ORGANIZATION:DURATION ORGANIZATION:NUMBER , ORGANIZATION:LOCATION , ORGANIZATION:DATE ORGANIZATION:NATIONALITY , ORGANIZATION:IDEOLOGY , ORGANIZATION:PERSON ORGANIZATION:MISC , ORGANIZATION:COUNTRY , ORGANIZATION:CAUSE_OF_DEATH ORGANIZATION:URL , ORGANIZATION:CRIMINAL_CHARGE		

Table 11: Entity type constraints on TACRED.

Relation	Constraint	Relation	Constraint
place served by transport hub	FAC:GPE	mountain range	MOUNTAIN:MOUNTAIN , MOUNTAIN:GLACIER , GLACIER:MOUNTAIN GLACIER:GLACIER
religion	LOC:NORP , GPE:NORP , ORG:NORP	participating team	EVENT:GPE
contains administrative territorial entity	GPE:GPE	head of government	GPE:PERSON
country of citizenship	PERSON:GPE	original network	BROADCASTER:ORG , NETWORK:ORG
heritage designation	WORK_OF_ART:LOC	performer	WORK_OF_ART:PERSON
participant of	PERSON:EVENT	position held	LOC:EVENT
has part	ORG:PERSON	location of formation	ORG:GPE
located on terrain feature	GPE:LOC , GPE:GPE	architect	FAC:PERSON
country of origin	PERSON:GPE	publisher	WORK_OF_ART:ORG
director	WORK_OF_ART:PERSON	father	PERSON:PERSON
developer	GAME:ORG , SEQUEL:ORG , WEBSITE:ORG	military branch	PERSON:ORG
mouth of the watercourse	RIVER:RIVER , RIVER:LAKE , RIVER:STREAM RIVER:TRIBUTARY , LAKE:RIVER , LAKE:LAKE LAKE:STREAM , LAKE:TRIBUTARY , STREAM:RIVER STREAM:LAKE , STREAM:STREAM , STREAM:TRIBUTARY TRIBUTARY:RIVER , TRIBUTARY:LAKE , TRIBUTARY:STREAM TRIBUTARY:TRIBUTARY	nominated for	WORK_OF_ART:WORK_OF_ART
movement	PERSON:NORP , PERSON:ORG	successful candidate	DATE:PERSON
followed by	WORK_OF_ART:WORK_OF_ART	manufacturer	MODEL:ORG
instance of	DATE:EVENT	after a work by	WORK_OF_ART:WORK_OF_ART , WORK_OF_ART:PERSON
member of political party	PERSON:POLITICAL PARTY	licensed to broadcast to	ORG:GPE
headquarters location	COMPANY:GPE , CONGLOMERATE:GPE , SUBSIDIARY:GPE	sibling	PERSON:PERSON
instrument	PERSON:FAC	country	PERSON:ORG , PERSON:GPE
occupation	PERSON:PERSON	residence	PERSON:GPE
work location	PERSON:GPE	subsidiary	ORG:ORG
participant	EVENT:PERSON	operator	PRODUCT:PERSON
characters	PERSON:PERSON	occupant	FAC:ORG

Table 12: Entity type constraints on Wiki80.