

Corpus-Based Relation Extraction by Identifying and Refining Relation Patterns

Sizhe Zhou^{1,*}, Suyu Ge^{2,*}, Jiaming Shen³, and Jiawei Han² (✉)

¹ Shanghai Jiao Tong University, Shanghai, China
sizhezhou@sjtu.edu.cn

² University of Illinois Urbana-Champaign, Urbana, USA
{suyuge2, hanj}@illinois.edu

³ Google Research, New York City, USA
jmshen@google.com

Abstract. Automated relation extraction without extensive human-annotated data is a crucial yet challenging task in text mining. Existing studies typically use lexical patterns to label a small set of high-precision relation triples and then employ distributional methods to enhance detection recall. This *precision-first* approach works well for common relation types but struggles with unconventional and infrequent ones. In this work, we propose a *recall-first* approach that first leverages high-recall patterns (e.g., a `per:siblings` relation normally requires both the head and tail entities in the person type) to provide initial candidate relation triples with weak labels and then clusters these candidate relation triples in a latent spherical space to extract high-quality weak supervisions. Specifically, we present a novel framework, RCLUS, where each relation triple is represented by its head/tail entity type and the shortest dependency path between the entity mentions. RCLUS first applies high-recall patterns to narrow down each relation type’s candidate space. Then, it embeds candidate relation triples in a latent space and conducts spherical clustering to further filter out noisy candidates and identify high-quality weakly-labeled triples. Finally, RCLUS leverages the above-obtained triples to prompt-tune a pre-trained language model and utilizes it for improved extraction coverage. We conduct extensive experiments on three public datasets and demonstrate that RCLUS outperforms the weakly-supervised baselines by a large margin and achieves generally better performance than fully-supervised methods in low-resource settings.

Keywords: Relation Extraction · Weak Supervision · Latent Space Clustering.

1 Introduction

Relation extraction, which aims to extract semantic relationships between the head and tail entities as shown in Figure 1, is crucial to various downstream tasks including hypernymy detection [34], knowledge base construction [26], and question answering [35,41,39]. A common practice of relation extraction is to fine-tune pre-trained language models with massive human annotations as full supervisions. As human annotations are

* Equal contribution.

Relation Extraction Task Setting:

Sentence: *It's a meeting of L.C.K., a civil rights organization founded by Shawn.*

Head Entity: L.C.K.

Tail Entity: Shawn



Relation between Head & Tail Entities:
org:founded_by

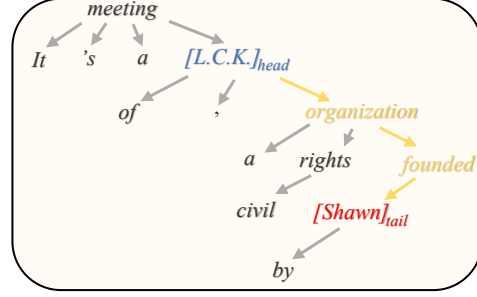
Dependency Parsing Tree & Dependency Path:

Fig. 1. Sentence's relation is explicitly contained in the dependency path. Head entities are indicated in blue while tail entities are indicated in red. The shortest dependency path connecting each pair of head entity and tail entity is indicated in light yellow.

expensive to acquire, potentially outdated or even noisy, such supervised methods are unable to scale. Instead of relying on massive human annotations, weakly-supervised relation extraction has been explored to tackle the data scarcity issue [25,27,49]. To improve the efficiency and minimize the expense of obtaining annotations, weakly-supervised relation extraction leverages only an incomplete set of pre-defined patterns to automatically annotate a portion of the corpus with weak labels as supervision [15,28].

In general, weakly-supervised relation extraction methods can be divided into two types: alignment-based and distributional. Alignment-based approaches obtain weak labels by exactly aligning pre-defined lexical patterns (e.g., certain tokens between entities or entity co-occurrence) with unlabeled examples from the corpus [25,21,29,15]. However, due to such context-agnostic hard matching process, the labels annotated by alignment-based approaches are noisy and suffer from limited recall and semantic drift [8]. Distributional approaches try to tackle such issues by encoding textual patterns with neural models so that the pattern matching can be conducted in a soft matching way [5,49,27]. Typically, distributional approaches utilize the alignment-based weak supervision or scarce human annotations at the initial stage to train neural encoder models [34]. However, such dependence introduces the severe problem of initial noise propagation [47,46]. Besides, the dependence on the initial alignment-based weak supervision along with the noise propagation also causes such distributional approaches to suffer from semantic drift and generalization problems.

To tackle the above mentioned high precision but low recall issue, we propose a novel recall-first framework RCLUS for weakly-supervised relation extraction which takes the sentence, head entity and tail entity as input and return the extracted relations as output (see Figure 1 for an example). Instead of sticking to the traditional precision-first philosophy for weak supervision, RCLUS starts with initial weak supervisions with high recall and then further refines the weak supervision. Our RCLUS framework features three key designs as follows.

First, instead of relying on annotated data, RCLUS utilizes pre-defined patterns to obtain weak labels. To maximize the recall of weak supervision, RCLUS uses entity types along with relation-indicative words as relation identifiers for weak supervision. The head and tail entity types are usually fixed for a specific relation type. For example, the relation `org:founded_by` generally specifies the head entity as an organization and the tail entity as a person. Utilizing the entity requirements along with occurrence of relation-indicative words, such as “founder” and “establish”, maximizes the recall of the weak supervision.

Second, based on the maximized recall, RCLUS tries to compensate the precision by presenting a novel representation of relation triples and conducting clustering on the representations. As utilizing relation-indicative words for weak supervision ignores the complete semantics for the relation expression, RCLUS adopts the shortest dependency path as the relation-related context within which the relation-indicative words will be searched. For example, the shortest dependency path in Figure 1 helps neglecting irrelevant information including *It’s a meeting of* and *civil rights*. The shortest dependency path is adopted as it retains the most relevant information to the target relation which is hence beneficial to the precision [44,36,13,7]. Furthermore, as the above alignment-based weakly-supervised extraction only focuses on local indicative words in the relation-related contexts, the assigned weak labels still suffer from noise. For example, *William talks with the founder team of the company D.M..* will give $\langle \text{org., company} - \text{founder_team} - \text{talks, person} \rangle$ which satisfies the entity requirements and contains the indicated word “founder” of relation `org:founded_by`. However, based on the complete semantics from the sentence expression, it’s unclear whether *D.M.* is founded by *William* or not. To prevent such noisy extractions, RCLUS proposes to cluster on a latent space which accommodates the objective to highlight the salient relation-related contexts across the corpus to isolate noisy contexts.

Third, in order to generalize to implicit and other varied expression patterns of relations to further improve the recall of the whole system, RCLUS prompt-tunes a pre-trained language model based on the limited but quality samples selected from the clustering space. To consolidate the pre-defined rules as the foundation for generalization, RCLUS selects quality samples from the clustering for tuning as these samples are noise-reduced and well represent the pre-defined patterns for relations. Meanwhile, RCLUS aggregates sub-prompts to extract relation-related components from the entire sentence and to improve context understanding. Compared with fine-tuning, prompt-tuning has a closer objective to the pre-training objective of language models. Thus, RCLUS can more efficiently distill the knowledge acquired from pre-training for generalizing the relation patterns under low resource setting.

To summarize, our main contributions are as follows: (1) We have proposed a weakly-supervised relation extraction framework based on the novel recall-first philosophy of weak supervision construction and then improve precision to tackle the data scarcity issue, (2) we have designed the relation triple representation extraction and the latent space clustering to mitigate the noisy labeling and noise propagation issues and we have incorporated prompt-tuning to mitigate the generalization issues, and (3) we have conducted extensive experiments on three relation extraction datasets to verify the effectiveness of our framework.

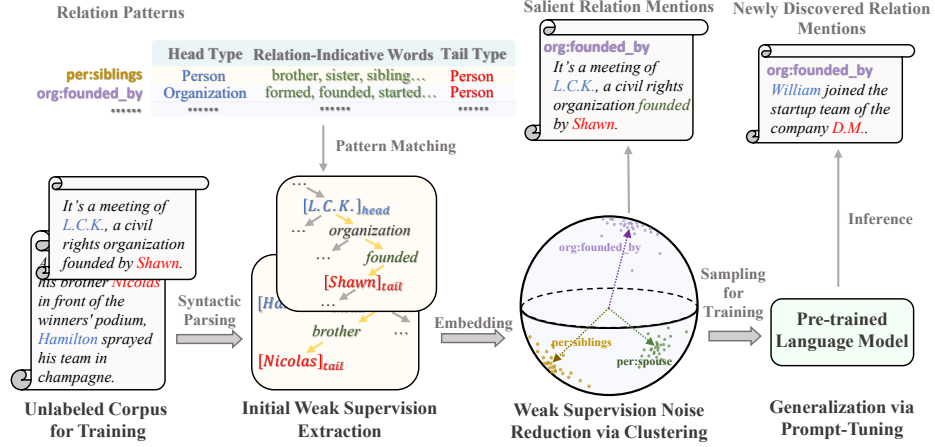


Fig. 2. Framework overview. Our model mainly consists of three steps: (1) relation triple extraction, (2) relation triple clustering, and (3) prompt-tuning with sub-prompts.

2 Problem Formulation

Let corpus $\mathcal{S} := \{S_1, \dots, S_N\}$ be a set of sentences with each sentence S_i consists of a word sequence $[w_{i,1}, \dots, w_{i,n_i}]$. For relation extraction, it is assumed that for each sentence S_i , a head entity $W_{h,i}$ and a tail entity $W_{t,i}$ are given, and both of them are represented by a sub-sequence of the sentence. Given S_i , $W_{h,i}$ and $W_{t,i}$, the goal of relation extraction is to predict the relation $y_i \in \mathcal{Y}$ between $W_{h,i}$ and $W_{t,i}$ which is the most appropriate based on the sentence, where $\mathcal{Y}$ is a set of pre-defined relations.

3 Methodology

In Figure 2, we outline our framework that extracts relations from corpus in three major steps: (1) *initial weak supervision extraction* which matches the extracted representations of relation triple with pre-defined patterns to obtain weak labels with high recall (Section 3.2), (2) *weak supervision noise reduction via clustering* in a latent spherical space which mines salient relation-related contexts to filter the noisy weak labels for improving precision (Section 3.3), and (3) *generalization via prompt-tuning* which leverages salient samples from the clustering space to recall implicit and varied relation expressions (Section 3.4).

3.1 Representation of Relation Triple

We first introduce the concept of representations of relation triples which is fundamental for our initial weak supervision extraction. Then we introduce the method to construct the corresponding embeddings for representations of relation triple which will be used for latent space clustering.

Representation of Relation Triple The relation triple is defined to be in the form of $\langle head\ entity, relation, tail\ entity \rangle$. Based on the definition of relation triples, we further define the representation of relation triple which is the example-specific triple containing the essential relation-related information. The formulation of relation triple representations is aimed to automatically annotate examples with most suitable weak labels while maximally reducing the noise under the low resource setting. To assign weak relation labels with maximal suitability, the head and tail entity types along with the relation-indicative words serve as strong relation identifiers. For example, instead of using only the entity mentions *L.C.K.* and *Shawn* in Figure 1, the entity types *organization* and *person* together with the word *founded* also indicate the relation label *founded_by*. However, directly matching the relation-indicative words (e.g. *founded* above) in the whole sentence will likely get distracted by the noise from parts of sentence which are irrelevant to the target entities’ relationship. Previous studies suggest that shortest dependency paths between head and tail entities retain the most relevant information to their relation [44,36,13,7] which makes it perfect to isolate noise from irrelevant contexts. As shown in Figure 1, the semantics of *founded* in the shortest dependency path between *L.C.K.* and *Shawn* is clearly relevant to entities’ semantic relationship. Meanwhile, other parts of the sentence beyond the shortest dependency path such as “a civil right organization” is not relevant to the semantic relationship. Therefore the use of shortest dependency paths further avoids noise from directly matching with relation-indicative words.

Based on the above intuitions, we define a representation of relation triple K as $\langle h, r, t \rangle$ where h indicates the head entity type, t indicates the tail entity type, and r indicates the shortest dependency path starting from head entity mention to tail entity mention. Each valid representation of relation triple K is associated with a relation y . For the sentence in Figure 1, a representation of relation triple would be $\langle org., organization-founded, person \rangle$ associated with relation $org: founded_by$.

Embedding for Representation of Relation Triple Suppose relation triple representation extraction gives M representations of relation triples $\{K_1, \dots, K_M\}$. For each relation triple representation $K_i = \langle h_i, r_i, t_i \rangle$, we acquire its initial features in the form of relation triple representation embeddings $\langle \vec{h}_{h_i}, \vec{h}_{r_i}, \vec{h}_{t_i} \rangle$ which includes: head entity embedding $\vec{h}_{h_i}$, dependency path embedding $\vec{h}_{r_i}$ and tail entity embedding $\vec{h}_{t_i}$.

Head/Tail Entity Embedding: We derive the embedding for head or tail entity based on their entity type surface names. Namely, head entity embedding $\vec{h}_{h_i} \in \mathbf{H}_h$ is obtained by retrieving and averaging pre-trained token embeddings⁴ of the head entity type surface name h_i . The tail entity embedding $\vec{h}_{t_i} \in \mathbf{H}_t$ is constructed likewise. Here $\mathbf{H}_h$ and $\mathbf{H}_t$ denote the semantic spaces for head and tail entities respectively.

Dependency Path Embedding: To capture the complete semantics of the dependency information, we construct the contextualized embedding $\vec{h}_{r_i}^{cont}$ as one component of the dependency path embedding $\vec{h}_{r_i}$. To accommodate the word choice variation of the dependency path (e.g., “founded” and “established” alternatively for the same relation

⁴ For simplicity in feature acquisitions, we adopts BERT-Large [10] as the pre-trained language model for all the encoding.

org:founded_by), we construct masked language modeling embedding $\vec{h}_{r_i}^{mask}$ with BERT [10] to obtain the other component of the dependency path embedding $\vec{h}_{r_i}$.

Assume the dependency path r_i is composed of m_{r_i} words $\{w_{i_K,1}, \dots, w_{i_K,m_{r_i}}\}$ from original sentence $S_{i_K} \in \mathcal{S}$ which are not necessarily consecutive in S_{i_K} but are necessarily consecutive in the dependency parse tree by definition. To obtain $\vec{h}_{r_i}^{cont}$, we feed sentence S_{i_K} to pre-trained language model and retrieve the corresponding encoded vectors of r_i as $\{\vec{h}_{w_{i_K,1}}, \dots, \vec{h}_{w_{i_K,m_{r_i}}}\}$. $\vec{h}_{r_i}^{cont}$ can be calculated with average pooling.

To obtain $\vec{h}_{r_i}^{mask}$, we replace each word in the dependency path with a mask token [MASK], feed the masked sentence to the pre-trained language model and retrieve the corresponding encoded vectors of r_i as $\{\vec{h}_{mask_{i_K,1}}, \dots, \vec{h}_{mask_{i_K,m_{r_i}}}\}$. $\vec{h}_{r_i}^{mask}$ can be similarly calculated with average pooling.

Finally, the dependency path embedding is constructed by the concatenation of two components:

$$\vec{h}_{r_i} = [\vec{h}_{r_i}^{mask}, \vec{h}_{r_i}^{cont}] \in \mathbf{H}_r \quad (1)$$

where $\mathbf{H}_r$ denotes the semantic space for dependency path.

After the above feature acquisition process, for each extracted representation of the relation triple K_i , we have obtained the relation triple embedding $\langle \vec{h}_{h_i}, \vec{h}_{r_i}, \vec{h}_{t_i} \rangle \in \mathbf{H}_h \times \mathbf{H}_r \times \mathbf{H}_t$ for relation triple representation clustering.

3.2 Initial Weak Supervision Extraction

Based on weakly-supervised setting and the formulation of relation triple representation, we maintain corresponding entity types and a limited set of relation-indicative words for each relation to construct the pre-defined relation patterns (see the table in Figure 2 as an example). In contrast to previous weakly-supervised approaches that applies pattern matching in a precision-first manner, we first adopts the philosophy of recall-first and later improve the precision for weak supervision. Given our pursuit of high recall, we assign weak labels once the entity types are matched and the relation-indicative words are captured in the shortest dependency path.

Utilizing the pre-defined relation patterns for constructing initial weak supervision, we first conduct dependency parsing and named entity typing⁵ on each sentence $S_i \in \mathcal{S}$. Based on the parsing results, we find the shortest dependency path between each pair of head entity $W_{h,i}$ and tail entity $W_{t,i}$ so that each sentence S_i will correspond to one candidate representation of relation triple. Second, we align the pre-defined relation patterns and the relation triple representation candidates so that relation triple representations which have the matched entity types and the indicative words will be assigned with a weak label.

3.3 Weak Supervision Noise Reduction via Clustering

As the matching-based extraction of the initial weak supervision only focuses on in-sentence indicative words which leads to noisy weak labels and hence low precision,

⁵ For convenience, we use the Stanford CoreNLP toolkit [20].

RCLUS introduces latent space clustering which highlights salient relation-related contexts across the corpus for noise filtering. Given the semantic spaces of the head entity, the tail entity and the relation-related context, RCLUS fuses the three semantic spaces onto a joint latent spherical space of much lower dimensionality for clustering. The rationale for such fusing method for clustering are two folds: (1) Angular similarity in spherical space is more effective than Euclidean metrics to capture word semantics [24,22], and (2) clustering while optimizing the projection onto a joint lower dimensional space can force the RCLUS to model the interactions between the head entity, the tail entity and the relation related contexts, discarding irrelevant information in the relation-related contexts. In contrast, a naïve clustering method on the dimension reduced or simply concatenated semantic spaces of the relation triple representations without integrating any clustering promoting objective is weak to guarantee the above suitability.

Clustering Model We use the clustering model to regularize the interactions between the head entity and the tail entity and discard noise in relation-related contexts. We assume that there exists a latent space $\mathbf{Z} \subset \mathbb{S}^{d-1}$ ⁶ with C clusters. Each cluster corresponds to one relation and is represented by a von Mises-Fisher (vMF) distribution [4].

The vMF distribution is controlled by a mean vector $\mu \in \mathbf{Z}$ and a concentration parameter $\kappa \in \mathbb{R}^+ \cup \{0\}$. The vMF probability density function for a unit vector z is given by

$$p(z|\mu, \kappa) = n_d(\kappa) \cdot \exp(\kappa \cdot \cos(z, \mu)). \quad (2)$$

Here $n_d(\kappa)$ is the normalization constant defined as

$$n_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}, \quad (3)$$

where $I_{d/2-1}(\cdot)$ represents the modified Bessel function of the first kind at order $d/2 - 1$.

With the assumption on the relation clusters, we further make assumptions on the generation of relation triple embeddings $\langle \vec{h}_{h_i}, \vec{h}_{r_i}, \vec{h}_{t_i} \rangle$ as follows: (1) A relation type c is uniformly sampled over C relations: $c \sim \text{Uniform}(C)$, (2) a latent embedding z_i is generated from the vMF distribution with mean vector μ_c and concentration parameter κ : $z_i \sim \text{vMF}_d(\mu_c, \kappa)$, (3) three functions $g_h(\cdot)$, $g_r(\cdot)$, $g_t(\cdot)$ respectively map the latent embedding z_i to the original relation triple embeddings $\vec{h}_{h_i}$, $\vec{h}_{r_i}$ and $\vec{h}_{t_i}$:

$$\vec{h}_{h_i} = g_h(z_i), \vec{h}_{r_i} = g_r(z_i), \vec{h}_{t_i} = g_t(z_i) \quad (4)$$

To enhance joint optimization, we follow the autoencoder structure [16] to jointly optimize the decoding mappings $g_h : \mathbf{Z} \rightarrow \mathbf{H}_h$, $g_r : \mathbf{Z} \rightarrow \mathbf{H}_r$, $g_t : \mathbf{Z} \rightarrow \mathbf{H}_t$ and an encoding mapping $f : \mathbf{H}_h \times \mathbf{H}_r \times \mathbf{H}_t \rightarrow \mathbf{Z}$.

Model Training To optimize the salient context mining without supervision, we adopt a pre-training and EM optimization process [9] with the reconstruction objective and the clustering-promoting objective.

⁶ $\mathbb{S}^{d-1} := \{z \in \mathbb{R}^d \mid \|z\| = 1\}$. We assume that $d \ll \min(\dim(\mathbf{H}_h), \dim(\mathbf{H}_r), \dim(\mathbf{H}_t))$.

In the E-step, we update the clustering assignment estimation $q(\mu_c|z_i)$ by computing the posterior distribution as

$$p(\mu_c|z_i) = \frac{p(z_i|\mu_c)p(\mu_c)}{\sum_{c'=1}^C p(z_i|\mu_{c'})p(\mu_{c'})} = \frac{\exp(\kappa \cdot z_i^T \cdot \mu_c)}{\sum_{c'=1}^C \exp(\kappa \cdot z_i^T \cdot \mu_{c'})} \quad (5)$$

. The target distribution is derived as $q(\mu_c|z_i)$:

$$q(\mu_c|z_i) = \frac{p(\mu_c|z_i)^2/s_c}{\sum_{c'=1}^C p(\mu_{c'}|z_i)^2/s_{c'}} \quad (6)$$

with $s_c := \sum_{j=1}^K p(\mu_c|z_j)$. The squaring-then-normalizing formulation is shown to introduce a sharpening effect which shifts the estimation towards the most confident area so that different clusters will have more distinct separation [23,43].

The corresponding clustering-promoting objective is defined as

$$\mathcal{O}_{clus} = \sum_{j=1}^K \sum_{c'=1}^C q(\mu_{c'}|z_j) \cdot \log p(\mu_{c'}|z_j) \quad (7)$$

and the reconstruction objective is defined as

$$\mathcal{O}_{recon} = \sum_{j=1}^K \sum_{l \in \{h,r,t\}} \cos(\vec{h}_{l_j}, g_l(f(\vec{h}_{h_j}, \vec{h}_{r_j}, \vec{h}_{t_j}))) \quad (8)$$

The reconstruction objective leads the model to preserve the input space semantics while conducting mappings.

In the M-step, the mapping functions $g_h(\cdot)$, $g_r(\cdot)$, $g_t(\cdot)$, $f(\cdot)$ and cluster distribution parameters are updated by maximizing $\mathcal{O}_{recon} + \lambda \mathcal{O}_{clus}$.

After convergence, there are C well-separated clusters $\{\mu_{c'}\}_{c'=1}^C$. Each cluster centroid $\mu_{c'}$ is associated with a cluster of relation triples $\{K_j^{(c')}\}_{j=1}^{M_{c'}}$ where $M_{c'}$ denotes the number of relation triples affiliated with cluster centroid $\mu_{c'}$.

3.4 Generalization via Prompt-Tuning

Even with high recall and the improved precision, the weak supervision still suffer from following deficiencies. First of all, the weak supervision extraction by hard matching the pre-defined patterns with the extracted relation triple representations is deficient to handle implicitly expressed relations that need to be inferred from the whole sentence context beyond dependency path. One example for the relation `per:grandparent` is *Alice, the wife of Mike, gave birth to Beck three months before Mike's father, John, visited her*. This example sentence indicates John is the grandparent of Beck by incorporating the `per:mother`, `per:spouse` and `per:father` relations and it is hard to cover such complicated and implicit patterns for applying weak supervision. Second, the pre-defined relation patterns suffer from limited coverage due to the hard matching nature of the weak supervision construction. For example, the set of relation-indicative words for `org:founded_by` is far from completeness.

To tackle the first deficiency, we select samples with salient relation-related contexts from the clustering space for tuning pre-trained language models⁷. These high-quality samples well represent the pre-defined patterns whose noise from initial weak supervision construction is largely reduced after clustering. By tuning the pre-trained language models leveraging these samples, RCLUS is capable to learn the essence of the pre-defined patterns and to generalize to other implicit relation patterns that need to be reasoned from the context.

To tackle the second deficiency, instead of fine-tuning, we tune the language models with prompts. As prompt-tuning has a much closer objective to the pre-training objective, RCLUS is hence much more efficient in distilling the knowledge acquired from pre-training for generalizing the high-quality patterns under low resource setting. Sticking to our philosophy of designing the pre-defined relation patterns, we follow [14] to aggregate three sub-prompts to jointly contribute to the inference of prompt-tuning. As each target relation is generally equivalent to the combination of the head entity type, the tail entity type and the semantic relationship between the head and the tail. The three sub-prompts are hence designed corresponding: (1) the sub-prompt for inferring the head entity type and it consists of a mask and the head entity mention, (2) the sub-prompt for inferring the semantic relationship independent to entity types (e.g., “gave birth to”) and it consists of three masks, and (3) the sub-prompt for inferring the tail entity type same as (1). The original sentence and the three sub-prompts will be concatenated in order to tune the pre-trained language model. RCLUS integrates the inference of the three sub-prompts to give the extracted relation. As an example, to give the relation `org:founded_by`, the three sub-prompts will need to predict the head entity as an organization, the tail entity as a person and the semantic relationship between entities as “was founded by”.

Model K	TACRED				TACREV				ReTACRED			
	4	8	16	Mean	4	8	16	Mean	4	8	16	Mean
<i>w/ weak supervision</i>												
EXACT MATCHING*	-	-	-	47.98	-	-	-	53.67	-	-	-	54.86
COSINE	23.28	26.60	37.16	29.01	21.43	30.85	41.21	31.16	28.12	35.00	44.54	35.89
COSINE*	-	-	-	58.88	-	-	-	60.80	-	-	-	68.59
RCLUS NOISY	45.35	50.94	55.73	50.67	50.41	61.67	66.85	59.64	56.89	65.81	71.09	64.60
RCLUS BALANCED	45.19	55.71	59.33	53.41	55.36	58.74	64.56	59.55	53.84	65.27	69.60	62.90
RCLUS	49.89	56.65	60.26	55.60	56.94	63.75	66.50	62.40	61.03	68.78	72.23	67.35
<i>w/ ground truth supervision</i>												
FINE-TUNING	13.62	26.09	32.07	23.93	18.75	25.21	35.12	26.36	17.36	31.77	42.63	30.59
GDPNET	13.79	28.42	43.11	28.44	15.61	24.59	42.12	27.44	19.20	35.79	52.84	35.94
PTR	39.16	49.46	54.67	47.76	47.18	51.58	59.17	52.64	51.27	62.60	71.11	61.66

Table 1. F₁ scores (%) on full test set with different sizes ($K = 4, 8, 16$) for each relation label. 3 seeds (212, 32, 96) are used for uniformly random sampling and the median value is taken as the final result for robustness against extreme values. Note that - means for that setting, no size limitation on labeled samples for training is assumed and the evaluation results will be indicated under *Mean* column. Models under such setting is also indicated with *.

⁷ For this work, we use RoBERTa_Large [50] as the backbone model and maintain the consistency between baselines in experiments.

4 Experiments

In the following experiments⁸, we first show the effectiveness of RCLUS on three classical relation extraction datasets (Section 4.1) and the details of prompt tuning data sampling (Section 4.2). Then, we illustrate the clustering results with the help of t-SNE [19] (Section 4.3) and study the importance of each part of RCLUS with an ablation study (Section 4.4). Additionally, we qualitatively show some sources of improvement brought by RCLUS with a case study (see Appendix C).

4.1 Relation Extraction

Datasets: We carry out the experiments on three relation extraction datasets: (1) TACRED [48] which is the most widely used large scale dataset for sentence-level relation extraction. There are 41 common relations and 1 NA⁹ label for negative samples. (2) TACREV [3] which corrects wrong labels for part of dev and test data of TACRED. (3) ReTACRED [37] which refactors the TACRED dataset and also modifies some relations for suitability.

Without loss of generality, we sampled 30 relations from original relations of each dataset for the convenience of designing weak supervisions. 27 relations are shared across 3 datasets. For the left 3 relations, the `org:country_of_headquarters`, the `org:stateorprovince_of_headquarters` and the `org:city_of_headquarters`, ReTACRED modifies the *headquarters* to *branch*. The statistics are shown in Table 2.

Dataset	#train	#dev	#test	#relations
TACRED	65,044	21,226	14,637	30
TACREV	65,044	21,238	14,681	30
ReTACRED	49,419	15,780	10,375	30

Table 2. Statistics about datasets experimented

Model	TACREV		
	Precision	Recall	F ₁
RCLUS			
w/ Weak	59.30	49.02	53.67
w/ Prompt	48.25	75.73	58.95
w/ Weak + Prompt	58.80	72.07	64.76
w/ Weak + Cluster	63.62	40.61	49.57
w/ Weak + Cluster + Prompt	60.76	74.29	66.85
w/ Weak + Cluster + Prompt*	57.85	78.47	66.61

Table 3. Ablation study of RCLUS

Baselines: We compare RCLUS with : (1) EXACT MATCHING: prediction is given based on our pre-defined relation patterns. (2) COSINE [46]: weakly-supervised model that utilizes contrastive self-training to extend labeled dataset and de-noise. (3) FINE-TUNING: a RoBERTa_Large [50] backbone plus a classification head whose input is the sequence classification embedding concatenated with the averaged embeddings of head and tail entities. (4) GDPNET [45]: it constructs a multi-view graph on top of BERT. (5) PTR [14]: RCLUS’s backbone prompt-tuning model except for some modifications.

⁸ The code for this work is available at <https://github.com/KevinSRR/RClus>

⁹ no_relation for TACREV and ReTACRED.

For training with weak supervision, we assume the negative samples in the train set are known but only $2 \times K \times \text{Number of Positive Labels}$ of negative samples can be used.

As RCLUS requires applying predefined patterns to positive examples, there will be examples that do not match any pattern or match with multiple patterns. RCLUS will ignore such examples while RCLUS NOISY will assign negative label and assign the first matched relation respectively, only not using them for clustering. This is the only difference between RCLUS and RCLUS NOISY. For RCLUS, as prompt-tuning requires data sampled from clusters, it involves sampling of both positive and negative samples. The sampling details are in Section 4.2.

For fair comparisons, we study the low-resource setting performance. For weakly-supervised baselines without being denoted with *, we provide small training sets as weakly-labeled data while leaving the remaining data as unlabeled data. For fine-tuning based or prompt-tuning baselines, we also provide them same sizes of training sets but with ground truth labels. The difference between RCLUS and RCLUS BALANCED is that, after positive data sampling, RCLUS compensate the relation classes with samples fewer than K using weak supervisions to reach K while RCLUS BALANCED will cut down exceeded samples after the same compensation to keep the sample size of each positive relations as K .

Evaluation Metrics: We follows the micro F_1 metric adopted by most of the works that experiment on three datasets. Note that mainstream approaches only calculate this metric over positive samples while neglecting negative samples.

Experiment Setups: For detailed experimental setups, please refer to Appendix B.

Main Analysis: The results of RCLUS and other baselines are shown in Table 1. In general, compared with weakly-supervised baselines and supervised baselines with ground truth, RCLUS achieves better performances under low-resource scenarios. The advantage is more significant when compared to strong weakly-supervised baselines, demonstrating the overall effectiveness of RCLUS.

Compared with PTR which is the backbone model of our prompt-tuning method, it can be seen that with latent space clustering, RCLUS, with weak supervision, has improved PTR’s performance by a large margin. Additionally, it can be seen that RCLUS can be easily adapted for integrating other more powerful prompt-tuning backbones for better performance. This shows the effectiveness of the whole pipeline design as well as the further potential of RCLUS.

Considering different levels of labeled data scarcity, RCLUS’s advantage over baselines with ground truth supervision is most significant when the ground truth samples are scarce, as the pre-defined patterns and the pattern generalizability of RCLUS will reach a limit as baselines with ground truth supervision get to reach more and more patterns directly from more provided samples.

In summary, considering low-resource settings where ground truth annotations are scarce, RCLUS is effective to boost the performance and it maintains the advantage across different levels of data scarcity compared with current strong baselines.

4.2 Positive and Negative Samples

As the positive training samples for RCLUS’s final step of prompt-tuning needs to be sampled from clusters, we design RCLUS’s sampling method to sampling with intervals.

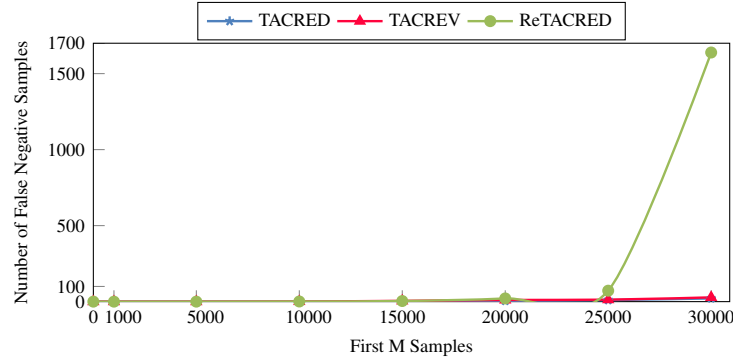


Fig. 3. Number of false negative samples among first M samples (consider both negative examples and weakly labeled positive examples) with smallest max assignment probability.

Based on the cluster assignment probability (see Eq. 5), sampling with interval I means, for each centroid starting from highest assignment probability in descending order, one sample will be taken among every I candidate samples. The purpose is to avoid repetitions of similar samples as there are many similar or reused samples from three datasets.

Opposite to positive samples, for the three datasets, negative samples are defined to have relations that are beyond labeled relations. The relation triple representation extraction and clustering is targeted at positive samples that fall into the range of defined relations. Therefore, for model training, RCLUS needs to obtain negative samples.

To obtain quality negative samples, RCLUS follows a min max approach. After the latent space clustering on extracted relation triple representations with positive relations, we apply the trained mapping function $f : \mathbf{H}_h \times \mathbf{H}_r \times \mathbf{H}_t \rightarrow \mathbf{Z}$ to project the unextracted relation triple representations (or negative samples) and sort them by their maximal centroid assignment probability to each centroids given by Eq. 5 in ascending order. Then the negative relation triples are sampled uniformly from the first $M_{negative}$ sorted relation triple representations.

This method follows the intuition that trained clusters using positive relations well represent the salient features of positive relations. Therefore, negative samples will be projected as outliers in the same clustering space. To further enhance the prompt tuning effectiveness, the range of first $M_{negative}$ samples guarantees minimal distinction between the sampled negative and positive samples. While uniform sampling introduces different levels of difficulty for prompt tuning to distinguish the sampled positive and negative samples. Figure 3 verifies our intuition as the ratios of false negative samples are 0.070%, 0.093%, and 5.46%, against the overall negative sample ratios as 9.92%, 9.54%, and 15.43% for TACRED, TACREV and ReTACRED respectively.

4.3 Cluster Visualization

Clustering result visualized with t-SNE [19] is shown in Figure 4. We can see that clusters generally have well separated boundaries, which means that the clusters well

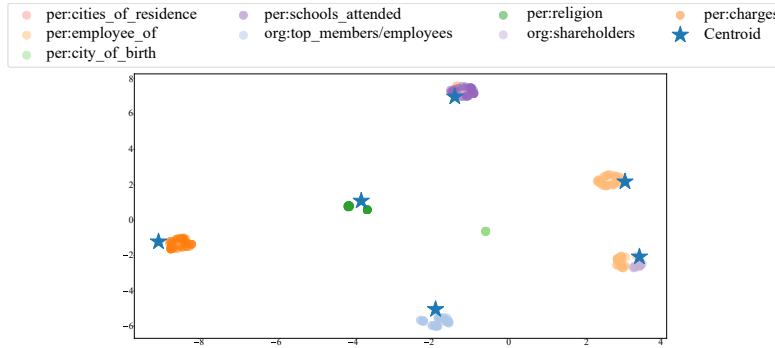


Fig. 4. Visualization of the clusters with t-SNE. For clarity, we sample 6 cluster centroids for TACREV dataset and visualize them along with first 40 data points closest to each centroid.

capture the salient features of relation triple patterns of different relations. In rare cases, relations that have semantically close patterns might have close latent representations. For example, the patterns of `org:employee_of` and `org:shareholders` share the same head and tail entity types and the two relations can generally have similar contexts. So they are in the same cluster as shown by Figure 4. As our clustering is unsupervised, some clusters may represent more than one relation. However, instead of being leveraged to assign labels, our clustering is only used to filter noisy relation triple representations which are expected to be outliers after clustering. Hence such problem will not influence any relation extraction results. For example, even the above discussed cluster contains samples of `org:employee_of` and `org:shareholders`, we will only sample them with their labels assigned by pattern matching as they are not outliers.

4.4 Ablation Study

In order to show the importance of each component of RCLUS, an ablation study is conducted with results shown in Table 3. Note that we take 16 for few-shot settings and seed as 212 if needed. *Weak* refers to weak supervision, *Prompt* refers to prompt-tuning, and *Cluster* refers to relation triple clustering. And * denotes RCLUS BALANCED. Generally, each component is indispensable to the whole framework based on the evaluation performance. Specifically, it can be seen that the weak supervision and the clustering are both important for the precision metric as they are important for the model to capture certain patterns and reduce initial weak supervision noise. The weak supervision provides relatively high recall while the clustering provides high precision. Additionally, the prompt-tuning is important for the boost of recall as it helps the model to comprehend the whole context, generalize the patterns and infer the implicit relations. This is in accordance with design expectations.

5 Related Work

De-noising for Weakly-Supervised Learning: Our work is related to de-noising models for weakly-supervised learning. Previous methods design probabilistic models to aggregate multiple weak supervisions to achieve de-noising yet ignoring the contexts for further improvement [28,38,2]. Other studies either focus on noise transitions without dealing with instance-level de-noising or require too much supervision to be suitable for weakly-supervised relation extraction [40,30]. A recent study proposes a contrastive self-training based de-noising method but cannot bypass potential issues of noise propagations from initial weak supervision [46]. Different from them, RCLUS adopts unsupervised relation triple clustering which captures salient semantic features of relation expressions to filter noises from weak supervision.

Prompt-Tuning: Another related line of research is prompt-tuning for low-resource settings. Enhanced by the birth of GPT-3 [6], prompt-tuning has introduced various studies [17,11,31,33,18]. For prompt-tuning with relation extraction, PTR incorporates logic rules with sub-prompts to construct prompts [14]. KNOWPROMPT tries to include prior knowledge on labels into prompts for relation extraction. In contrast, their studies are based on extending prompt models’ generalizability without making use of weak supervision to boost the performance. Based on prompt-tuning, RCLUS adopts prompt-tuning to achieve generalizability with a strong base on learnt noise-reduced patterns.

6 Conclusions

In this work, we propose a novel weakly-supervised framework that extracts relations from unlabeled data. Specifically, our framework: (1) has designed new relation patterns for a novel *recall-first* philosophy for weak supervision construction, (2) designed a novel representation of relation triple for initial weak supervision construction for high recall and then utilized clustering to mine salient contexts to improve precision, (3) leveraged the samples from clusters for prompt-tuning to enhance generalization and context understanding. Experiments show RCLUS largely outperforms the strongest weakly-supervised baseline and achieves better performance than fully-supervised methods under low-resource settings. We also show the importance of each component of RCLUS and verify design expectations quantitatively and qualitatively.

Acknowledgements

Research was supported in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and INCAS Program No. HR001121C0165, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

Ethical Statement

To the best of our knowledge, there is no specific ethical concern for the methodology of RCLUS. However, since RCLUS is dependent on external entity typing tools, pre-trained language models and also the given corpus, potential errors or bias should be given appropriate awareness and be taken good care of.

A Rules for Weak Supervision

The pre-defined relation triple patterns adopted by RCLUS are shown in Table 4 for reference. Before applying the predefined patterns, named entity recognition will be applied for each sentence and the recognition results will be aligned with the given head and tail entity mentions for entity typing. During this step, sentences which are not recognized will not be considered for training and will be considered as negative sample if it comes from the test set.

B Experiment Setups

We conduct all the experiments on 2 NVIDIA RTX A6000 GPU’s with PyTorch 1.12.1 with Huggingface Library [42]. For all experiments, we set the training epochs as 20 and the evaluation frequency on dev set is set to be once every 2 epochs. Best performing checkpoint on dev set is chosen for evaluation on the test set.

B.1 RCLUS Implementation Details

For pre-defined relation triple representation patterns, please refer to Appendix A.

For the clustering model’s decoding function $g_l(\cdot)$ with $l \in \{h, r, t\}$, we implement them as feed-forward neural networks with each layer followed by the ReLU activation function [1]. We adopt 100, 1000, 2000, 1000, dim_l for the hidden states dimensions of each layer. The dim_l is 1024 for head/tail entity embeddings and 2048 for dependency path embeddings. For encoding mapping f , we basically reverse the layout of the three decoding functions and concatenate them to form the latent space vector $z \in \mathbb{R}^{300}$. For the clustering, the concentration parameter κ of each vMf distribution is set as 10, the λ is chosen as 5. During training, the batchsize is 256 while the learning rate is $5e - 4$. Additionally, we set the tolerance threshold for optimization convergence as 0.001 which means when the ratio of the examples with changed cluster assignment is fewer than this threshold, the training will stop. The pre-training epochs with only objective $\mathcal{O}_{recon}$ is set as 100, the interval for updating the target assignment distribution $q(\mu_c|z_i)$ is set as 100 and the max training epochs is set as 500 which are never reached. The remaining experiment setups are the same as [32].

For prompt-tuning with sub-prompts, we used the verbalizer and the label word set from [14] except that we have modified some prompt templates and search the learning rate from $\{3e - 5, 4e - 5\}$ and we search the max input sequence length from $\{256, 512\}$. The computation time on 2 GPU’s are generally within one and half an hour across all settings.

The hyperparameter search space for sampling interval I is $\{1, 2, 3\}$ and the $M_{negative}$ is searched among $\{10000, 20000, 30000\}$. The currently found best combinations are $I = 3$ and $M_{negative} = 30000$.

B.2 BASELINE Implementation Details

For baselines that have been originally implemented on TACRED, we adopt the same experimental setups with the original settings on TACRED.

Type	Example Relation	Ours	PTR Baseline
Succinctness (avoids redundant information from original corpus)	per:stateorprovince_of_birth	(i) <i>Jane Matilda Bolin</i> was born on April 11, 1908, in Poughkeepsie, <i>NY</i> .	(i) <i>Jane Matilda Bolin</i> was born on April 11, 1908, in Poughkeepsie, <i>NY</i> . (ii) <i>Jane Matilda Bolin</i> was born on April 11, 1908, in Poughkeepsie, <i>N.Y.</i> (iii) "Born, raised and educated in our home state of <i>Ohio</i> , Paul never lost sight of the reason <i>he</i> came to Congress -- to serve this great institution and his constituents with dedication and distinction," House Republican Leader John Boehner, also of <i>Ohio</i> , said in a statement. (iv) "Born, raised and educated in our home state of <i>Ohio</i> , Paul never lost sight of the reason he came to Congress -- to serve this great institution and <i>his</i> constituents with dedication and distinction," House Republican Leader John Boehner, also of <i>Ohio</i> , said in a statement.
Human Annotation Noise Filtration (avoids noise from human annotations by applying explicit patterns)	per:charges	(i) In 1968, <i>Flowers</i> was accused with two others on federal charges of <i>extorting payments</i> from life insurance companies in return for being allowed to do business in the state when <i>Flowers</i> was attorney general. (ii) A judge said Wednesday she was denying a prosecutor's request to increase bail following <i>Remy Ma</i> 's new indictment on <i>witness tampering</i> and gang assault charges.	(i) <i>Flowers</i> always contended politics was behind the <i>extortion</i> investigation, but appeals courts ruled against <i>him</i> . (ii) Three blocks away, officers discovered a luxury sport utility vehicle owned by <i>Remy Ma</i> , 26, that had been involved in a single - <i>car crash</i> and abandoned.

Fig. 5. Case study for showing the effectiveness of RCLUS’s weakly-supervised relation extraction. The aspects of effectiveness is shown by the *Type* column. *Ours* stands for the relation instances sampled from the clustering of RCLUS. *PTR Baseline* stands for the relation instances obtained by random sampling. To aid the quantitative comparison between RCLUS and PTR, the dataset is selected as ReTACRED and the training size is selected as 16 per positive relation. Note that head entities are indicated in blue while tail entities are indicated in red.

One exception is COSINE and we search self_training_eps from {0.7, 0.8}, self_training_update_period from {100}, self_training_max_step from {2000, 3000}, and training epochs from {3, 20}. Other experimental setups are the same as the original implementation.

C Case Study

In order to qualitatively study the improvement brought by RCLUS, we show the result of the conducted case study here. The examples are randomly collected on ReTACRED task for comparison between RCLUS and PTR in terms of their input relation instances. Note that RCLUS and shares the same prompt-tuning backbone model while RCLUS achieves a large margin under low-resource settings.

In Figure 5, the observed qualitative improvements brought by our weak supervisions and latent space clustering are summarized into succinctness and human annotation noise filtration. In terms of succinctness, we can see that our clustering method gets rid of extremely similar training samples from the original corpus. This might be due to the fact that our clustering is capable to capture the semantic similarities of relation patterns. In terms of human annotation noise filtering, we can see RCLUS filters relation instances that originally contain human annotation noise thanks to our weak supervision rules and relation triple representation’s condensed formulation of relation expressions.

Relation	Head Type	Relation-Indicative Word	Tail Type
per:charges	Person	face; sentence; sentenced; sentence; sentenced; accused; charge; charged; charges; convicted; conviction; indictment; sued; sues; trial;	Criminal_charge
per:date_of_death	Person	deaths; dies; deaths; dies; dead; death; die; died; killed;	Date
per:country_of_death	Person	death; died; killed;	Country
per:cause_of_death	Person	succumbed; death; deaths; die; died; dies; killed;	Cause_of_death
org:founded_by	Organization	set; set; create; co-founder; cofounder; established; form; formed; found; founded; founder; founders; founding; founds; started;	Person
org:founded	Organization	creation; establish; form; establish; co-founder; cofounder; form; founder; co-founded; created; established; formed; founded; founding; founds; set; started;	Date
per:city_of_death	Person	murdered; murdered; death; died; dies; killed;	City
per:stateorprovince_of_death	Person	death; died; killed;	State_or_province
per:date_of_birth	Person	birth; birthday; born;	Date
per:stateorprovince_of_birth	Person	native; birth; birthday; born;	State_or_province
per:country_of_birth	Person	native; birth; birthday; birthplace; born;	Country
per:city_of_birth	Person	native; birth; birthday; birthplace; born;	City
org:shareholders	Person	acquired; bought; holding; share; holding; share; owned; stock; invest; invested; investment; investor; investors; invests; owner; owns; shareholder; shareholders; shares; stake; stakes;	Organization
per:other_family	Person	relative; sister-in-law; brother-in-law; father-in-law; mother-in-law; nephew; nephews; niece; grandfather; grandmother; grandson; granddaughter; cousin; aunts; uncle;	Person
per:title	Person	Length(Shortest Dependency Path) ≤ 2	Title
org:dissolved	Organization	bankruptcy; shuttering; bankruptcy; shuttering; merger; close; closed; disband; disbanded; dissolved; merged; shut; shutdown; split; takeover;	Date
per:countries_of_residence	Person	ambassador; congressman; congresswoman; lives; sen.; senator; senators; undersecretary; ambassador; ambassadors; grew; leave; return; returned; returns; apartment; citizen; emigrated; home; hometown; house; inhabitant; live; lived; living; moved; resided; resident; resides; sen.; senator; senators;	Country
per:stateorprovinces_of_residence	Person	congressman; congresswoman; grew; native; raised; represent; representative; represented; representing; sen.; senator; senators; apartment; emigrated; home; hometown; house; live; lived; lives; living; moved; resided; resident; resides; native; settled; settled; grew; move; moving; returned;	State_or_province
per:cities_of_residence	Person	apartment; emigrated; home; hometown; house; live; lived; lives; living; moved; resided; resident; resides;	City
per:religion	Person	Length(Shortest Dependency Path) ≤ 2	Religion
org:top_members/employees	Organization	cfo; heads; leaders; leads; vice-chairman; cfo; governor; heads; leaders; leadership; vice; vice-chairman; vice-governor; ceo; chairman; chairwoman; chief; commander; coo; director; head; lead; leader; led; president; vice-president;	Person
org:number_of_employees/members	Organization	employs; employs; workforce; employees; members; membership; people; workers;	Number
per:schools_attended	Person	attending; education; studied; studies; education; educated; graduation; students; study; studying; attended; attends; degree; degrees; doctorate; graduate; graduated; graduates; graduating; schooled; schools; student	Organization
per:employee_of	Person	boss; chief; directors; driver; executive; heads; leader; leaders; manager; officer; officials; professor; undersecretary; worker; works; chief; attorney; co-chairman; correspondent; driver; elected; executive; fellow; minister; officer; officials; professor; serve; speaker; undersecretary; appointed; chairman; chairwoman; director; employed; employee; employees; head; hired; member; members; official; president; secretary; served; spokesman; spokeswoman; worked; workers;	Organization
per:siblings	Person	sibling; siblings; brother; brothers; sister; sisters;	Person
per:spouse	Person	marrying; divorce; marrying; wedding; ex-wife; husband; marriage; married; spouse; wife;	Person
org:stateorprovince_of_headquarters	Organization	base; based; located; founded; firm; firms; office; headquarters; headquartered; belongs;	State_or_province
org:country_of_headquarters	Organization	agency; headquarters; company; companies; based; root; firm; founded; located;	Country
org:city_of_headquarters	Organization	base; located; based; headquarters; headquartered; company; firm; office; founded;	City
org:stateorprovince_of_branch	Organization	branches; run; runs; base; based; in; operate; operated; operates; office; site; headquarters; headquartered; plant; office; company; firm; firms; branch; operations; operators; operator;	State_or_province
org:country_of_branch	Organization	branch; branches; in; office; offices; base; based; headquarters; headquartered; run; runs; operations; operate; operated; operates; operators; operator; agency; company; head; firm; subsidiary; companies; plants; site; sites;	Country
org:city_of_branch	Organization	branch; branches; site; run; runs; in; operations; operates; operate; operated; base; based; headquarters; headquartered; plant; office; offices; company; firm; firms; subsidiary;	City

Table 4. Pre-defined relation patterns of RCLUS. The patterns are designed based on example sentences and the relation definitions from *TAC KBP 2014 Slot Descriptions* [12]. Note that the joint relation-indicative words are in black. Additional relation-indicative words for TACRED are in red. Additional relation-indicative words for TACREV are in blue. Additional relation-indicative words for ReTACRED are in cyan.

References

1. Agarap, A.F.: Deep learning using rectified linear units (relu). CoRR **abs/1803.08375** (2018), <http://arxiv.org/abs/1803.08375>
2. Aina, L., Gulordava, K., Boleda, G.: Putting words in context: LSTM language models and lexical ambiguity. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 3342–3348. Association for Computational Linguistics, Florence, Italy (Jul 2019). <https://doi.org/10.18653/v1/P19-1324>, <https://aclanthology.org/P19-1324>
3. Alt, C., Gabryszak, A., Hennig, L.: TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1558–1569. Association for Computational Linguistics, Online (Jul 2020). <https://doi.org/10.18653/v1/2020.acl-main.142>, <https://aclanthology.org/2020.acl-main.142>
4. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res.* **6**, 1345–1382 (2005)
5. Batista, D.S., Martins, B., Silva, M.J.: Semi-supervised bootstrapping of relationship extractors with distributional semantics. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 499–504. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). <https://doi.org/10.18653/v1/D15-1056>, <https://aclanthology.org/D15-1056>
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) *Advances in Neural Information Processing Systems*. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf>
7. Chen, Y.N., Hakkani-Tür, D., Tur, G.: Deriving local relational surface forms from dependency-based entity embeddings for unsupervised spoken language understanding. In: 2014 IEEE Spoken Language Technology Workshop (SLT). pp. 242–247 (2014). <https://doi.org/10.1109/SLT.2014.7078581>
8. Curran, J., Murphy, T., Scholz, B.: Minimising semantic drift with mutual exclusion bootstrapping. *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics* pp. 172–180 (12 2008)
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1), 1–38 (1977), <http://www.jstor.org/stable/2984875>
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). <https://doi.org/10.18653/v1/N19-1423>, <https://aclanthology.org/N19-1423>
11. Ding, N., Chen, Y., Han, X., Xu, G., Xie, P., Zheng, H., Liu, Z., Li, J.Z., Kim, H.G.: Prompt-learning for fine-grained entity typing. *ArXiv abs/2108.10604* (2021)
12. Ellis, J., Getman, J., Strassel, S.M.: Overview of linguistic resources for the tac kbp 2014 evaluations: Planning, execution, and results. In: Proceedings of TAC KBP 2014 Workshop, National Institute of Standards and Technology. pp. 17–18 (2014)

13. Fundel, K., Küffner, R., Zimmer, R.: Relex - relation extraction using dependency parse trees. *Bioinformatics* **23** 3, 365–71 (2007)
14. Han, X., Zhao, W., Ding, N., Liu, Z., Sun, M.: Ptr: Prompt tuning with rules for text classification. *AI Open* **3**, 182–192 (2022). <https://doi.org/https://doi.org/10.1016/j.aiopen.2022.11.003>, <https://www.sciencedirect.com/science/article/pii/S2666651022000183>
15. Hancock, B., Varma, P., Wang, S., Bringmann, M., Liang, P., Ré, C.: Training classifiers with natural language explanations. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pp. 1884–1895. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). <https://doi.org/10.18653/v1/P18-1175>, <https://aclanthology.org/P18-1175>
16. Hinton, G.E., Zemel, R.S.: Autoencoders, minimum description length and helmholtz free energy. In: *Proceedings of the 6th International Conference on Neural Information Processing Systems*. p. 3–10. NIPS’93, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
17. Hu, S., Ding, N., Wang, H., Liu, Z., Li, J.Z., Sun, M.: Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In: *Annual Meeting of the Association for Computational Linguistics* (2021)
18. Lu, Y., Bartolo, M., Moore, A., Riedel, S., Stenetorp, P.: Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *CoRR* **abs/2104.08786** (2021), <https://arxiv.org/abs/2104.08786>
19. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *Journal of Machine Learning Research* **9**(86), 2579–2605 (2008), <http://jmlr.org/papers/v9/vandermaaten08a.html>
20. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. pp. 55–60. Association for Computational Linguistics, Baltimore, Maryland (Jun 2014). <https://doi.org/10.3115/v1/P14-5010>, <https://aclanthology.org/P14-5010>
21. Mausam, Schmitz, M., Soderland, S., Bart, R., Etzioni, O.: Open language learning for information extraction. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pp. 523–534. Association for Computational Linguistics, Jeju Island, Korea (Jul 2012), <https://aclanthology.org/D12-1048>
22. Meng, Y., Huang, J., Wang, G., Zhang, C., Zhuang, H., Kaplan, L., Han, J.: Spherical text embedding. In: *Advances in Neural Information Processing Systems* (2019)
23. Meng, Y., Shen, J., Zhang, C., Han, J.: Weakly-supervised neural text classification. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. p. 983–992. CIKM ’18, Association for Computing Machinery, New York, NY, USA (2018). <https://doi.org/10.1145/3269206.3271737>, <https://doi.org/10.1145/3269206.3271737>
24. Meng, Y., Zhang, Y., Huang, J., Zhang, Y., Zhang, C., Han, J.: Hierarchical topic mining via joint spherical tree and text embedding. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. p. 1908–1917. KDD ’20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3394486.3403242>, <https://doi.org/10.1145/3394486.3403242>
25. Nakashole, N., Weikum, G., Suchanek, F.: PATTY: A taxonomy of relational patterns with semantic types. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. pp. 1135–1145. Association for Computational Linguistics, Jeju Island, Korea (Jul 2012), <https://aclanthology.org/D12-1104>

26. Nayak, T., Majumder, N., Goyal, P., Poria, S.: Deep neural approaches to relation triplets extraction: a comprehensive survey. *Cognitive Computation* **13**, 1215 – 1232 (2021)
27. Qu, M., Ren, X., Zhang, Y., Han, J.: Weakly-supervised relation extraction by pattern-enhanced embedding learning. In: *Proceedings of the 2018 World Wide Web Conference*. p. 1257–1266. WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018). <https://doi.org/10.1145/3178876.3186024>, <https://doi.org/10.1145/3178876.3186024>
28. Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S., Ré, C.: Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment* **11**(3), 269–282 (nov 2017). <https://doi.org/10.14778/3157794.3157797>, <https://doi.org/10.14778/3157794.3157797>
29. Ratner, A., Sa, C.D., Wu, S., Selsam, D., Ré, C.: Data programming: Creating large training sets, quickly. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. p. 3574–3582. NIPS'16, Curran Associates Inc., Red Hook, NY, USA (2016)
30. Ren, W., Li, Y., Su, H., Kartchner, D., Mitchell, C., Zhang, C.: Denoising multi-source weak supervision for neural text classification. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. pp. 3739–3754. Association for Computational Linguistics, Online (Nov 2020). <https://doi.org/10.18653/v1/2020.findings-emnlp.334>, <https://aclanthology.org/2020.findings-emnlp.334>
31. Schick, T., Schmid, H., Schütze, H.: Automatically identifying words that can serve as labels for few-shot text classification. In: *Proceedings of the 28th International Conference on Computational Linguistics*. pp. 5569–5578. International Committee on Computational Linguistics, Barcelona, Spain (Online) (Dec 2020). <https://doi.org/10.18653/v1/2020.coling-main.488>, <https://aclanthology.org/2020.coling-main.488>
32. Shen, J., Zhang, Y., Ji, H., Han, J.: Corpus-based open-domain event type induction. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. pp. 5427–5440. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). <https://doi.org/10.18653/v1/2021.emnlp-main.441>, <https://aclanthology.org/2021.emnlp-main.441>
33. Shin, T., Razeghi, Y., IV, R.L.L., Wallace, E., Singh, S.: Eliciting knowledge from language models using automatically generated prompts. *ArXiv abs/2010.15980* (2020)
34. Shwartz, V., Goldberg, Y., Dagan, I.: Improving hypernymy detection with an integrated path-based and distributional method. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pp. 2389–2398. Association for Computational Linguistics, Berlin, Germany (Aug 2016). <https://doi.org/10.18653/v1/P16-1226>, <https://aclanthology.org/P16-1226>
35. Simmons, R.F.: Answering english questions by computer: A survey. *Commun. ACM* **8**(1), 53–70 (jan 1965). <https://doi.org/10.1145/363707.363732>, <https://doi.org/10.1145/363707.363732>
36. Socher, R., Karpathy, A., Le, Q.V., Manning, C.D., Ng, A.Y.: Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics* **2**, 207–218 (2014). https://doi.org/10.1162/tacl_a_00177, <https://aclanthology.org/Q14-1017>
37. Stoica, G., Platanios, E.A., Póczos, B.: Re-tacred: Addressing shortcomings of the tacred dataset. In: *AAAI Conference on Artificial Intelligence* (2021)
38. Varma, P., Ré, C.: Snuba: Automating weak supervision to label training data. *Proc. VLDB Endow.* **12**(3), 223–236 (nov 2018). <https://doi.org/10.14778/3291264.3291268>, <https://doi.org/10.14778/3291264.3291268>

39. Wang, C., Kalyanpur, A., Fan, J., Boguraev, B.K., Gondek, D.C.: Relation extraction and scoring in deepqa. *IBM Journal of Research and Development* **56**(3.4), 9:1–9:12 (2012). <https://doi.org/10.1147/JRD.2012.2187239>
40. Wang, H., Liu, B., Li, C., Yang, Y., Li, T.: Learning with noisy labels for sentence-level sentiment classification. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. pp. 6286–6292. Association for Computational Linguistics, Hong Kong, China (Nov 2019). <https://doi.org/10.18653/v1/D19-1655>, <https://aclanthology.org/D19-1655>
41. Wang, H., Tian, F., Gao, B., Zhu, C., Bian, J., Liu, T.Y.: Solving verbal questions in iq test by knowledge-powered word embedding. In: *Conference on Empirical Methods in Natural Language Processing* (2015)
42. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.: Transformers: State-of-the-art natural language processing. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. pp. 38–45. Association for Computational Linguistics, Online (Oct 2020). <https://doi.org/10.18653/v1/2020.emnlp-demos.6>, <https://aclanthology.org/2020.emnlp-demos.6>
43. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. p. 478–487. ICML’16, JMLR.org (2016)
44. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pp. 1785–1794. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). <https://doi.org/10.18653/v1/D15-1206>, <https://aclanthology.org/D15-1206>
45. Xue, F., Sun, A., Zhang, H., Chng, E.S.: Gdpnet: Refining latent multi-view graph for relation extraction. In: *AAAI Conference on Artificial Intelligence* (2020)
46. Yu, Y., Zuo, S., Jiang, H., Ren, W., Zhao, T., Zhang, C.: Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. *ArXiv abs/2010.07835* (2020)
47. Zhang, J., Yu, Y., Li, Y., Wang, Y., Yang, Y., Yang, M., Ratner, A.: WRENCH: A comprehensive benchmark for weak supervision. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)* (2021), <https://openreview.net/forum?id=Q9SKS5k8io>
48. Zhang, Y., Zhong, V., Chen, D., Angeli, G., Manning, C.D.: Position-aware attention and supervised data improve slot filling. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pp. 35–45. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). <https://doi.org/10.18653/v1/D17-1004>, <https://aclanthology.org/D17-1004>
49. Zhou, W., Lin, H., Lin, B.Y., Wang, Z., Du, J., Neves, L., Ren, X.: Nero: A neural rule grounding framework for label-efficient relation extraction. In: *Proceedings of The Web Conference 2020*. p. 2166–2176. WWW ’20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3366423.3380282>, <https://doi.org/10.1145/3366423.3380282>
50. Zhuang, L., Wayne, L., Ya, S., Jun, Z.: A robustly optimized BERT pre-training approach with post-training. In: *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. pp. 1218–1227. Chinese Information Processing Society of China, Huhhot, China (Aug 2021), <https://aclanthology.org/2021.ccl-1.108>