# Corpus-Based Relation Extraction by Identifying and Refining Relation Patterns

Sizhe Zhou[1], Suyu Ge[2], Jiaming Shen[3], and Jiawei Han[2](✉)

[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, 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 Fig. 1, is crucial to various downstream tasks

S. Zhou and S. Ge—Equal contribution.

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14172, pp. 20–38, 2023.
https://doi.org/10.1007/978-3-031-43421-1_2

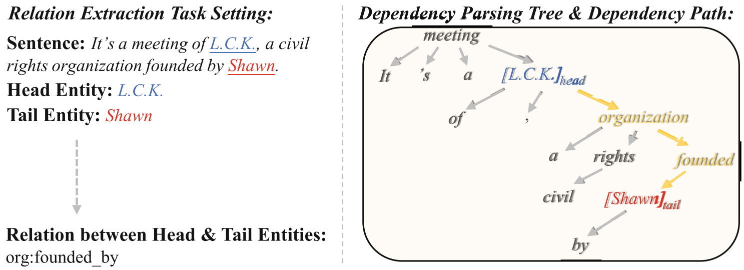
Relation Extraction Task Setting:
Sentence: It's a meeting of L.C.K., a civil rights organization founded by Shawn.
Head Entity: L.C.K.
Tail Entity: Shawn
Relation between Head & Tail Entities:
org:founded_by
Dependency Parsing Tree & Dependency Path:
meeting
It
's
a
[L.C.K.]head
of
,
organization
a
rights
founded
civil
[Shawn]tail
by

**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. (Color figure online)

including hypernymy detection [33], knowledge base construction [25], and question answering [34,38,40]. 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 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 [24,26,47]. 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 [14,27].

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 [14,20,24,28]. 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 [5,26,47]. Typically, distributional approaches utilize the alignment-based weak supervision or scarce human annotations at the initial stage to train neural encoder models [33]. However, such dependence introduces the severe problem of initial noise propagation [44,45]. 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 Fig. 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 Fig. 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 [7,12,35,42]. 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 $\langle org., company - founder\_team - talks, person\rangle$ 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.

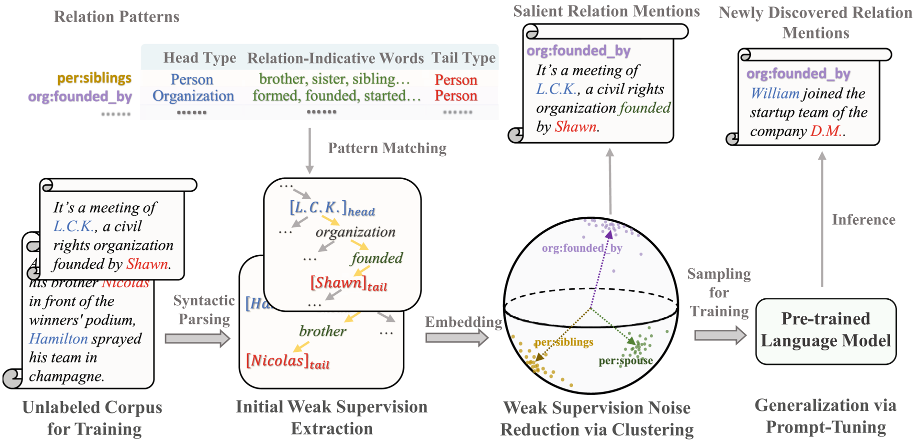
Relation Patterns
Head Type
Relation-Indicative Words
Tail Type
per:siblings
Person
brother, sister, sibling…
Person
org:founded_by
Organization
formed, founded, started…
Person
Pattern Matching
Salient Relation Mentions
org:founded_by
It's a meeting of L.C.K., a civil rights organization founded by Shawn.
Newly Discovered Relation Mentions
org:founded_by
William joined the startup team of the company D.M..
It's a meeting of L.C.K., a civil rights organization founded by Shawn.
his brother Nicolas in front of the winners' podium, Hamilton sprayed his team in champagne.
Syntactic Parsing
[L.C.K.]head
organization
founded
[Shawn]tail
[Ha
brother
[Nicolas]tail
Embedding
org:founded_by
per:siblings
per:spouse
Sampling for Training
Pre-trained Language Model
Inference
Unlabeled Corpus for Training
Initial Weak Supervision Extraction
Weak Supervision Noise Reduction via Clustering
Generalization via Prompt-Tuning

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

## 2 Problem Formulation

Let corpus $\mathcal{S} := \{S_1, \ldots, S_N\}$ be a set of sentences with each sentence $S_i$ consists of a word sequence $[w_{i,1}, \ldots, w_{i,n_i}]$. 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 Fig. 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 (Sect. 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 (Sect. 3.3), and (3) *generalization via prompt-tuning* which leverages salient samples from the clustering space to recall implicit and varied relation expressions (Sect. 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 Fig. 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 [7,12,35,42] which makes it perfect to isolate noise from irrelevant contexts. As shown in Fig. 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 Fig. 1, a representation of relation triple would be $\langle org., organization\text{-}founded, person\rangle$ associated with relation `org:founded_by`.

**Embedding for Representation of Relation Triple.** Suppose relation triple representation extraction gives $M$ representations of relation triples $\{K_1, \ldots, K_M\}$. 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 $\left\langle \vec{h}_{h_i}, \vec{h}_{r_i}, \vec{h}_{t_i} \right\rangle$ 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 embeddings[1] 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 `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 $h_{r_i}^{cont}$, we feed sentence $S_{i_K}$ to pre-trained language model and retrieve the corresponding encoded vecors 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: $\vec{h}_{r_i} = [\vec{h}_{r_i}^{mask}; \vec{h}_{r_i}^{cont}] \in \mathbf{H_r}$ 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 $\left\langle \vec{h}_{h_i}, \vec{h}_{r_i}, \vec{h}_{t_i} \right\rangle \in \mathbf{H_h} \times \mathbf{H_r} \times \mathbf{H_t}$ 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 Fig. 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 typing[2] on each

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

[2] For convenience, we use the Stanford CoreNLP toolkit [19].

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, 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 [21,23], 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 $\mathbf{Z} \subset \mathbb{S}^{d-1}$[3] 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 $p(z|\mu, \kappa) = n_d(\kappa) \cdot \exp(\kappa \cdot \cos(z, \mu))$. Here $n_d(\kappa)$ is the normalization constant defined as

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

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 $\left\langle \vec{h}_{h_i}, \vec{h}_{r_i}, \vec{h}_{t_i} \right\rangle$ as follows: (1) A

[3] $S^{d-1} := \{z \in \mathbb{R}^d | \|z\| = 1\}$. We assume that $d \ll \min(\dim(\mathbf{H_h}), \dim(\mathbf{H_r}), \dim(\mathbf{H_t}))$.

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 $\mu_c$ and concentration parameter $\kappa$: $z_i \sim \text{vMF}_d(\mu_c, \kappa)$, (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 $\vec{h}_{t_i}$: $\vec{h}_{h_i} = g_h(z_i), \vec{h}_{r_i} = g_r(z_i), \vec{h}_{t_i} = g_t(z_i)$.

To enhance joint optimization, we follow the autoencoder structure [15] to jointly optimize the decoding mappings $g_h : \mathbf{Z} \to \mathbf{H_h}$, $g_r : \mathbf{Z} \to \mathbf{H_r}$, $g_t : \mathbf{Z} \to \mathbf{H_t}$ and an encoding mapping $f : \mathbf{H_h} \times \mathbf{H_r} \times \mathbf{H_t} \to \mathbf{Z}$.

**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.

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

$$p(\mu_c|z_i) = \frac{p(z_i|\mu_c)p(\mu_c)}{\sum_{c'=1}^{C} p(z_i|\mu_{c'})p(\mu_{c'})} = \frac{\exp(\kappa \cdot z_i^T \cdot \mu_c)}{\sum_{c'=1}^{C} \exp(\kappa \cdot z_i^T \cdot \mu_{c'})} \tag{2}$$

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

$$q(\mu_c|z_i) = \frac{p(\mu_c|z_i)^2/s_c}{\sum_{c'=1}^{C} p(\mu_{c'}|z_i)^2/s_{c'}} \tag{3}$$

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 [22,41].

The corresponding clustering-promoting objective is defined as

$$\mathcal{O}_{clus} = \sum_{j=1}^{K} \sum_{c'=1}^{C} q(\mu_{c'}|z_j) \cdot \log p(\mu_{c'}|z_j) \tag{4}$$

and the reconstruction objective is defined as

$$\mathcal{O}_{recon} = \sum_{j=1}^{K} \sum_{l \in \{h,r,t\}} \cos(\vec{h}_{l_j}, g_l(f(\vec{h}_{h_j}, \vec{h}_{r_j}, \vec{h}_{t_j}))) \tag{5}$$

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 models[4]. 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 [13] 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".

[4] For this work, we use RoBERTa_Large [48] as the backbone model and maintain the consistency between baselines in experiments.

# 4 Experiments

In the following[5] we first show the effectiveness of RClus on three relation extraction datasets (Sect. 4.1) and the data sampling for prompt-tuning (Sect. 4.2). Then, we illustrate the clustering results utilizing t-SNE [18] (Sect. 4.3) and study the importance of each component of RClus with an ablation study (Sect. 4.4).

## 4.1 Relation Extraction

*Datasets:* We carry out the experiments on three relation extraction datasets: (1) TACRED [46] which is the most widely used large scale dataset for sentence-level relation extraction. There are 41 common relations and 1 `NA`[6] label for negative samples which are defined to have relations beyond labeled relations.. (2) TACREV [3] which corrects labels of part of dev and test set of TACRED. (3) ReTACRED [36] which refactors the TACRED and 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.

*Baselines:* We compare RClus with: (1) Exact Matching: prediction is given by pre-defined relation patterns. (2) COSINE [44]: weakly-supervised

**Table 1.** $F_1$ 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 *.

| Model | TACRED | | | | TACREV | | | | ReTACRED | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| K | 4 | 8 | 16 | Mean | 4 | 8 | 16 | Mean | 4 | 8 | 16 | Mean |
| *w/ weak supervision* | | | | | | | | | | | | |
| Exact Matching* | – | – | – | 48.87 | – | – | – | 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 | 71.03 | 63.38 |
| 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 |
| *w/ ground truth supervision* | | | | | | | | | | | | |
| 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 |

[5] The code for this work is available at https://github.com/KevinSRR/RClus.

[6] `no_relation` for TACREV and ReTACRED.

model that utilizes contrastive self-training to extend labeled dataset and denoise. (3) Fine-Tuning: a RoBERTa_Large [48] backbone plus a classification head whose input is the sequence classification embedding concatenated with the averaged embeddings of head and tail entities. (4) GDPNet [43]: it constructs a multi-view graph on top of BERT. (5) PTR [13]: RClus's backbone prompt-tuning model except for some modifications. For training with weak supervision, we assume the negative samples in the train set are known but only $2 \times K \times$ Number of Positive Labels of negative samples can be used.

As RClus requires applying pre-defined patterns on positive examples, there will be examples that match with zero or multiple patterns. RClus will ignore such examples while RClus Noisy will respectively assign negative label and the first matched relation, 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 whose details are in Sect. 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 provide 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 relations with samples fewer than K using weak supervisions until reaching K while RClus Balanced will cut down exceeded samples at the same time to keep 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 calculate this metric over positive samples only. We set the training epochs as 20 and the evaluation frequency on dev set as once every 2 epochs. Best checkpoint on dev set is chosen for evaluation.

**Table 2.** Statistics of datasets

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

**Table 3.** Ablation study of RClus

| 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 |

*Experiment Setups:* Baseline implementation details and the pre-defined patterns of RClus are uploaded with source codes[7]. Before applying patterns, NER

[7] https://github.com/KevinSRR/RClus.

will be leveraged for typing head and tail entity mentions. Sentences with unrecognized mentions will not be considered for training and will be seen as negative sample if in test set.

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 ReLU activation [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 $\kappa$ of each vMf distribution is set as 10, the $\lambda$ 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, and the interval for updating the target assignment distribution $q(\mu_c|z_i)$ is set as 100. The remaining experiment setups are similar to [31].

For prompt-tuning with sub-prompts, we used the verbalizer and the label word set from [13] 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 hyperparameter search space for sampling interval $I$ is $\{2, 3\}$ and the $M_{negative}$ is $\{10000, 20000, 30000\}$. A found good combination is $I$ as 3 and $M_{negative}$ as 30000.

*Main Analysis:* The results are shown in Table 1. Generally, compared with weakly-supervised baselines and supervised baselines with ground truth, our model achieves better performances under low-resource scenarios. The advantage is more significant when compared to weakly-supervised baselines, demonstrating the overall effectiveness.

Compared with PTR which is the backbone of our prompt-tuning method, RClus, with weak supervision and clustering, has improved PTR's performance by a large margin. Additionally, 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 data scarcity, RClus's advantage over baselines with ground truth supervision is most significant when ground truth samples are scarce, as the pre-defined patterns and the pattern generalizability of RClus will reach a limit while baselines with ground truth supervision can access more patterns from more samples.

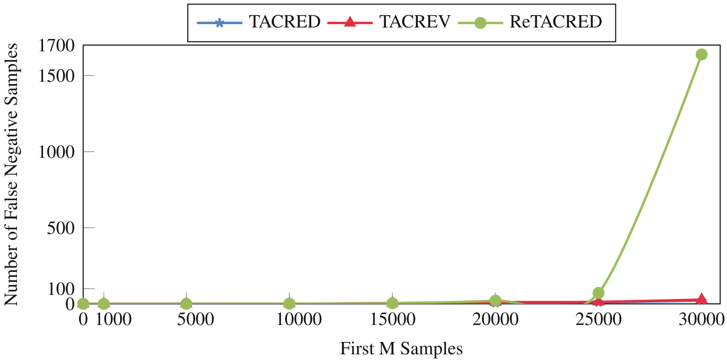

**Fig. 3.** Number of false negative samples among first M samples with smallest max-assignment-probability (consider both negative examples and weakly labeled positive examples).

### 4.2 Positive and Negative Samples

To obtain quality samples for prompt-tuning, RClus adopts sampling with intervals. Based on the cluster assignment probability by Eq. 2, sampling with interval $I$ means for each cluster, starting from highest assignment probability in descending order, taking one sample among every $I$ candidate samples. The purpose is to avoid repetitions of similar samples as there are numerous similar or reused samples from the datasets.

As relation triple representation extraction and clustering is targeted at positive samples that fall into the range of defined relations, for model training, RClus also needs 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 by their maximal assignment probability among all clusters given by Eq. 2 in ascending order. Then negative relation triples are sampled uniformly from the first $M_{negative}$ sorted relation triple representations.

This method follows the intuition that clusters trained using positive samples well represent the salient features of positive relations. Therefore, negative samples will be projected as outliers. 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

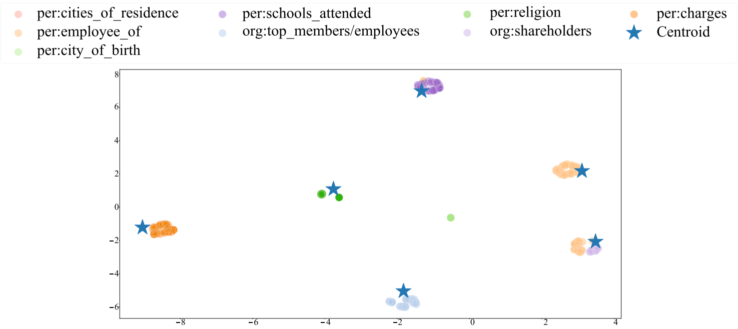

**Fig. 4.** Visualization of the clusters with t-SNE. For clarity, we sample 6 cluster centroids for TACREV dataset and visualize them along with first 40 data points closest to each centroid.

Clustering result visualized with t-SNE [18] is shown in Fig. 4. We can see that clusters generally have well separated boundaries, which means that the clusters well capture the salient features of different relation patterns. 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 Fig. 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 long 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 the seed as 212 if needed. *Weak* refers to weak supervision, *Prompt* refers to prompt-tuning, and *Cluster* refers to latent space 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 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 boosting recall as it helps 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:** Previous methods design probabilistic models to aggregate multiple weak supervisions to achieve de-noising yet ignoring the contexts for further improvement [2,27,37]. 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 [29,39]. A recent study proposes a contrastive self-training based de-noising method but cannot bypass potential issues of noise propagations from initial weak supervision [44]. Different from them, RClus adopts unsupervised relation triple representation clustering which captures salient semantic features of relation expressions to filter noises from weak supervision.

**Prompt-Tuning for Low-Resource Settings:** Enhanced by the birth of GPT-3 [6], prompt-tuning has introduced various studies [11,16,17,30,32]. KnowPrompt tries to include prior knowledge on labels into prompts for relation extraction. They focused on extending prompt models' generalizability without making use of weak supervision to boost the performance. In contrast, 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 relation extraction framework. 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 weakly-supervised baselines 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.

## 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 (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 (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 (2015). https://doi.org/10.18653/v1/D15-1056, https://aclanthology.org/D15-1056
6. Brown, T., et al.: 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/1457c0d6bfcb4967418bfb8ac142f64a-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. In: Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics, pp. 172–180 (2008)
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. Ser. 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 (2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
11. Ding, N., et al.: Prompt-learning for fine-grained entity typing. ArXiv abs/2108.10604 (2021)

12. Küffner, K., Zimmer, R., Fundel, R.: Relex - relation extraction using dependency parse trees. Bioinformatics **23**(3), 365–71 (2007)
13. 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/10.1016/j.aiopen.2022.11.003, https://www.sciencedirect.com/science/article/pii/S2666651022000183
14. 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 (2018). https://doi.org/10.18653/v1/P18-1175, https://aclanthology.org/P18-1175
15. 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, pp. 3–10. NIPS'93, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
16. 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)
17. 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
18. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(86), 2579–2605 (2008). http://jmlr.org/papers/v9/vandermaaten08a.html
19. 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 (2014). https://doi.org/10.3115/v1/P14-5010, https://aclanthology.org/P14-5010
20. 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 (2012). https://aclanthology.org/D12-1048
21. Meng, Y., et al.: Spherical text embedding. In: Advances in Neural Information Processing Systems (2019)
22. 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, pp. 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
23. 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, pp. 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

24. 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 (2012). https://aclanthology.org/D12-1104
25. Nayak, T., Majumder, N., Goyal, P., Poria, S.: Deep neural approaches to relation triplets extraction: a comprehensive survey. Cognitive Comput. **13**, 1215–1232 (2021)
26. 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, pp. 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
27. Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S., Ré, C.: Snorkel: rapid training data creation with weak supervision. Proc. VLDB Endowment **11**(3), 269–282 (2017). https://doi.org/10.14778/3157794.3157797
28. 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, pp. 3574–3582. NIPS'16, Curran Associates Inc., Red Hook, NY, USA (2016)
29. Ren, W., Li, Y., Su, H., Kartchner, D., Mitchell, C., Zhang, C.: Denoising multisource weak supervision for neural text classification. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 3739–3754. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.334, https://aclanthology.org/2020.findings-emnlp.334
30. 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) (2020). https://doi.org/10.18653/v1/2020.coling-main.488, https://aclanthology.org/2020.coling-main.488
31. 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 (2021). https://doi.org/10.18653/v1/2021.emnlp-main.441, https://aclanthology.org/2021.emnlp-main.441
32. 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)
33. 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 (2016). https://doi.org/10.18653/v1/P16-1226, https://aclanthology.org/P16-1226
34. Simmons, R.F.: Answering English questions by computer: a survey. Commun. ACM **8**(1), 53–70 (1965). https://doi.org/10.1145/363707.363732
35. Socher, R., Karpathy, A., Le, Q.V., Manning, C.D., Ng, A.Y.: Grounded compositional semantics for finding and describing images with sentences. Trans. Assoc. Comput. Linguist. **2**, 207–218 (2014). https://doi.org/10.1162/tacl_a_00177, https://aclanthology.org/Q14-1017

36. Stoica, G., Platanios, E.A., P'oczos, B.: Re-TACRED: addressing shortcomings of the TACRED dataset. In: AAAI Conference on Artificial Intelligence (2021)
37. Varma, P., Ré, C.: Snuba: Automating weak supervision to label training data. Proc. VLDB Endow. **12**(3), 223–236 (2018). https://doi.org/10.14778/3291264.3291268
38. Wang, C., Kalyanpur, A., Fan, J., Boguraev, B.K., Gondek, D.C.: Relation extraction and scoring in deepqa. IBM J. Res. Dev. **56**(3.4), 9:1–9:12 (2012). https://doi.org/10.1147/JRD.2012.2187239
39. 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 (2019). https://doi.org/10.18653/v1/D19-1655, https://aclanthology.org/D19-1655
40. 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)
41. 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, pp. 478–487. ICML'16, JMLR.org (2016)
42. 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 (2015). https://doi.org/10.18653/v1/D15-1206, https://aclanthology.org/D15-1206
43. 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)
44. 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)
45. Zhang, J., et al.: 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
46. 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 (2017). https://doi.org/10.18653/v1/D17-1004, https://aclanthology.org/D17-1004
47. Zhou, W., et al.: NERO: a neural rule grounding framework for label-efficient relation extraction. In: Proceedings of The Web Conference 2020, pp. 2166–2176. WWW '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3366423.3380282
48. 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 (2021). https://aclanthology.org/2021.ccl-1.108