Information Processing and Management 61 (2024) 103816

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

KGRED: Knowledge-graph-based rule discovery for weakly supervised data labeling

Wenjun Hou [a], Liang Hong [a,*], Ziyi Zhu [b]

[a] School of Information Management, Wuhan University, Wuhan 430072, China
[b] School of Computer, Wuhan University, Wuhan 430072, China

ARTICLE INFO

Keywords:
Data labeling
Weakly supervised learning
Rule discovery
Rule knowledge graph

ABSTRACT

In weakly supervised learning, labeling rules can automatically label data to train models. However, due to insufficient prior knowledge, rule discovery often suffers from semantic drift. Since misclassified rules are generated from wrongly matched sentences, the sentences matched by rules shift from the target labels to other labels. It is worth noting that rules do not exist in isolation. The multi-dimensional semantic associations among rules can impose semantic constraints for rule generation, as well as enrich the semantic information of rules for rule matching. Therefore, we propose a Knowledge-Graph-based RulE Discovery method (KGRED), which can leverage the multi-dimensional semantic associations among rules to alleviate semantic drift in rule discovery. Specifically, to decrease misclassified rules, we design a label-aware rule generation approach to attentively propagate prior knowledge from seed rules to candidate rules based on rule KG. To reduce wrongly-matched sentences, we present a cross-attention-based semantic matching mechanism to refine the semantic information of sentences while enriching that of rules. Moreover, we propose an inconsistency-directed active learning strategy to verify rules that perform inconsistently in rule generation and matching. Experiments on two public datasets prove that KGRED can achieve at least 5.1 % gain in F1 score compared to state-of-the-art methods.

1. Introduction

Deep learning models usually require sufficient high-quality labeled data to perform well (Sambasivan et al., 2021). However, due to the high cost of human annotation, collecting labeled data to train deep learning models is challenging for real-world applications, especially in expertise domains (Zhou et al., 2020; Zhang et al., 2021). While large language models have demonstrated significant performance in general natural language processing, they are not immune to "hallucination" issues, arising from limited domain-specific knowledge (Kojima et al., 2022). Labeling rules (rule for short) can extract typical patterns from domain corpus and generate labels automatically from unlabeled data, which have been widely used in weakly supervised learning (Li et al., 2021). For example, in Fig. 1, the rule r_1 is composed of a rule body "PER, PER, stranger" and a rule label "stranger". "PER" is the abbreviation of the entity category "Person". However, rule discovery is not an easy task, since it often suffers from semantic drift (Liang et al., 2021). As misclassified rules and wrongly matched sentences are introduced into the iterations of rule discovery, the sentences matched by rules shift from the target labels to other labels.

* Corresponding author.
E-mail address: hong@whu.edu.cn (L. Hong).

https://doi.org/10.1016/j.ipm.2024.103816
Received 3 December 2023; Received in revised form 27 February 2024; Accepted 15 June 2024
Available online 19 June 2024

Since sentences usually contain many semantically unrelated words with regard to rules, they may be assigned wrong labels. For example, in Fig. 1(a), s_1 represents the "mother" relation. However, it is wrongly matched by r_1 as the "stranger" relation. Meanwhile, r_3 represents the "mother" relation. However, s_1 contains unrelated words, such as "stranger", the semantic similarity between r_3 and s_1 is low. Therefore, r_3 cannot match s_1. Moreover, these wrongly matched sentences will generate misclassified rules. For example, based on the wrong label of s_1, r_2 may be misclassified as the "stranger" relation. As the rule discovery proceeds, wrongly matched sentences and misclassified rules will gradually dominate the iterations, which hurt the quality of rule labeling.

It is worth noting that rules do not exist in isolation in rule discovery, they have inherent semantic associations, which reveal the agreement and disagreement between the semantics of rules (Zhang & de Marneffe, 2021). On the one hand, by leveraging these associations, we can employ seed rules as semantic anchors (Xia et al., 2019) to impose semantic constraints on the labels of candidate rules. In Fig. 1(b), the label of r_1 is "stranger". Since the semantic information expressed by the body of r_1 contradicts that of r_2, the label of r_2 is less related to "stranger". In contrast, since the body of r_2 entails that of r_3, the label of r_2 may be the same as r_3. On the other hand, through the associations among rules, we can enrich the semantic information of rules to improve the coverage of rule matching. For instance, r_2 can help r_3 to match more related semantic information of s_1.

Previous studies have attempted to alleviate semantic drift in rule discovery. To decrease misclassified rules, Snorkel requires experts to write and improve rules based on labeling results (Ratner et al., 2017). Darwin generates a rule hierarchy to model the subsequence relations among rules and proposes traversal strategies to select candidate rules for human verification (Galhotra et al., 2021). However, these methods rely on experts to generate a final rule set, which requires a large amount of human effort. To decrease wrongly matched sentences, NERO adopts a soft matching mechanism to calculate the semantic similarity of sentences and rules (Zhou et al., 2020). However, it represents sentences into fixed vectors regardless of rules, which may introduce wrongly matched sentences.

Few existing works can proactively mitigate misclassified rules and wrongly matched sentences in rule discovery simultaneously. Therefore, we construct a rule Knowledge Graph (KG) based on multi-dimensional semantic associations among rules to alleviate semantic drift in rule discovery. However, it is not an easy task due to following challenges:

(1) In rule generation, multi-dimensional semantic associations among rules may impose conflicting semantic constraints on rule labels. (2) In rule matching, sentences may contain semantic information unrelated to rules, resulting in wrongly matched sentences. (3) It is difficult to verify rules based on limited prior knowledge, which in turn impacts the quality of newly generated rules.

In response to these challenges, we propose a Knowledge-Graph-based RulE Discovery method (KGRED) to alleviate semantic drift in rule discovery. Specifically, we construct a rule KG based on seed rules and candidate rules from the corpus. Then, we realize attentive information propagation to predict the labels of candidate rules through the label-aware rule generation approach. Next, to decrease the wrongly matched sentences, we present a cross-attention-based semantic matching mechanism to adaptively refine the semantic information of sentences while enriching that of rules. After rule generation and matching, we propose an inconsistency-directed active learning strategy to verify rules that perform inconsistently in rule discovery. Finally, the feedback of annotators is utilized to update the rule KG.

The main contributions of this study are as follows:

(1) To reduce misclassified rules, we design a label-aware rule generation approach to attentively propagate prior knowledge from seed rules to candidate rules based on rule KG.
(2) To reduce wrongly matched sentences, we present a cross-attention-based semantic matching mechanism to refine the semantic information of sentences while enriching that of rules based on rule KG.
(3) To improve the overall quality of rule discovery, we propose an inconsistency-directed active learning strategy to verify inconsistent rules in rule generation and matching.

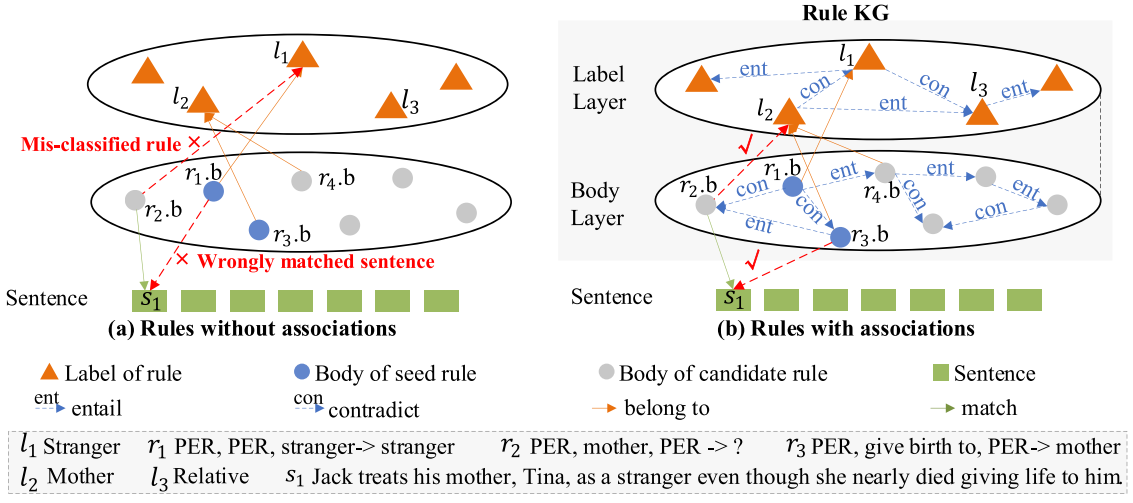

Fig. 1. Examples of rule generation and rule matching.

The rest of this paper is organized as follows. Section 2 provides a brief review of previous studies. Section 3 introduces the basic concepts and framework. Section 4 describes the proposed method. Section 5 presents experimental results. Section 6 presents the implications and conclusion.

2. Related work

2.1. Weakly supervised learning

Obtaining sufficient high-quality labeled data is critical for training successful deep learning models and it is often a bottleneck in response to changing real-world applications (Sambasivan et al., 2021; Liang et al., 2022). Since human annotation is time-consuming and labor-intensive, weakly supervised learning methods can be applied to (semi-)automatically generate labels based on limited prior knowledge (Whang et al., 2023). Weakly supervised sources have been adopted as labeling functions to provide labels for unlabeled corpus, e.g., labeling rules (Zhou et al., 2020), knowledge bases (Feng et al., 2017), and pre-trained language models (Zhang et al., 2022).

One of the main challenges in weakly supervised learning is wrong labels produced by labeling functions (Zhou, 2018). Some studies proposed distant-supervised methods to create training data (Zhang et al., 2024). Deng et al. (2021) designed a loss function that can minimize the negative impacts of the noisy label and class imbalance problems in distant supervision. Ye and Luo (2020) proposed a general ranking-based multi-label learning framework combined with convolutional neural networks to relieve the class imbalance problem in distant supervision. However, external knowledge is limited and cannot satisfy the requirements of varying domains. Zhao et al. (2023) employed a pre-trained language model as knowledge source to derive pseudo-labels for unlabeled policy texts. Although the pre-trained language model can achieve comparable performance, it needs to be continuously trained by policy texts, which is difficult to extend to other domains. Fries et al. (2021) extracted domain knowledge by manually constructing ontology and performed heuristic annotation of clinical data based on the ontology. However, the construction of domain ontology still requires additional supervision of experts.

2.2. Labeling rule discovery

Rule-based data labeling methods mine frequent patterns as rules to generate labels automatically (Zhou, 2018). However, rule discovery is often a challenging task. Data programming paradigm relies on domain experts manually developing rules to label data (Ratner et al., 2016; Kartchner et al., 2020). Snorkel invites users to interactively write labeling rules according to the feedback of labeling results and uses generative models to resolve the conflicts between multiple rules (Ratner et al., 2017). Safranchik et al. (2020) introduced linked hidden Markov models to obtain labels from noisy rules. Since manually designing rules can be time-consuming, some works generated rules automatically from matched sentences. However, the semantic drift problem may introduce misclassified rules and wrongly matched sentences into the iterations of rule discovery, which will affect the quality of rule labeling.

For rule generation, Li et al. (2018) utilized positive and negative rules to match positive and negative sentences respectively. Varma and Ré (2018) selected high-quality labeling rules through iterations of labeling, evaluating, and feedback. Liang et al. (2021) used conceptual taxonomy to filter misclassified rules, which can reduce the negative effect caused by semantic drift. However, they still relied on prior knowledge, which cannot be scalable to other domains. Yang et al. (2018) designed a game-based crowdsourcing mechanism to generate rules. Darwin selects potential rules through a rule hierarchy tree based on human annotation (Galhotra et al., 2021).

For rule representation, knowledge graphs are able to model multiple relationships among rules (Wang et al., 2020; Rossi et al., 2021). Wang et al. (2019) proposed an attention-based information propagation mechanism to represent the embeddings of nodes in KG. However, they tended to obtain semantic information of neighbor nodes and could not aggregate the common information of the same type of nodes. Zhong et al. (2023) generated a hierarchical structure for a graph and developed three propagation manners to realize hierarchical common information propagation between nodes. However, this work cannot handle multi-dimensional relationships in knowledge graphs. In KGRED, we design a label-aware rule generation approach to realize neighboring and common information propagation among rule bodies and labels in rule KG to decrease misclassified rules.

For rule matching, NERO (Zhou et al., 2020) adopts a self-attention-based soft matching mechanism to capture similar semantic sentences. However, it only considers the local semantic information of sentences, which still cannot adaptively solve the unbalanced semantic information between rules and sentences. Instead, in KGRED, we consider the mutual influence between semantic information of rules and sentences to refine the semantic information of sentences while enriching that of rules based on rule KG.

For rule verification, active learning strategies aim to select the most informative instances from unlabeled data, which can maximize the model's performance while minimizing the annotated cost (Ren et al., 2021). Typical active learning approaches can be grouped into two categories: uncertainty-based and diversity-based active learning (Buchert et al., 2022). Holub et al. (2008) used information entropy to assess uncertainty in unlabeled samples. Liu et al. (2021) selected the unlabeled samples that can provide the most positive influence on model performance. Du et al. (2023) proposed a contrastive active learning method to select diversity samples based on the semantics and distinctiveness of the instances. Different from existing works, we detect and verify inconsistent rules based on the results of rule generation and matching.

3. Preliminaries

3.1. Problem formulation

Definition 1 (Labeling rule). A labeling rule $r \in R = \{r_1, r_2, \ldots, r_i\}$ is composed of a rule body and a rule label as: $r.b \rightarrow l$. Rule body $r.b \in R.b = \{r_1.b, r_2.b, \ldots, r_i.b\}$ is a textual pattern $T = [w, SUB, w, OBJ, w]$. w denotes the context word sequence of subject *SUB* and object *OBJ*. Rule label $l \in L = \{l_1, l_2, \ldots, l_i\}$ indicates the labeling function of a rule.

Definition 2 (Rule KG). Rule KG is defined as $G = \{(n_i, p, n_j) \mid n_i, n_j \in V, p \in P\}$. Each triplet describes the semantic association p from subject n_i to object n_j. $V = V_L \cup V_{R.b}$ denotes a node set, where V_L and $V_{R.b}$ represent the nodes of rule bodies and labels. Meanwhile, $P = P_{B-L} \cup P_{B-B} \cup P_{L-L}$ denotes an edge set. $P_{B-L} = \{(n_i, p, n_j) | n_i \in V_{R.b}, n_j \in V_L\}$ denotes the semantic associations among rule bodies and labels. $P_{B-B} = \{(n_i, p, n_j) | n_i, n_j \in V_{R.b}\}$ denotes the semantic associations among rule bodies. $P_{L-L} = \{(n_i, p, n_j) | n_i, n_j \in V_L\}$ denotes the semantic associations among rule labels.

Specifically, in P_{B-L}, rule bodies and labels are connected through *belong to* associations. This type of association is generated based on the structure of rules. Meanwhile, to model the semantic agreement and disagreement among rule bodies and among rule labels, we classify P_{B-B} and P_{L-L} into two categories: *entail* and *contradict*. Specifically, *entail* associations indicate a subject and an object have the same semantic information, and the object can be inferred from the subject, i.e., $\forall v_i \in V, \exists v_j \in V,$ if v_i *entails* v_j, then v_i is True $\Rightarrow$ v_j is True. In contrast, *contradict* associations indicate a subject and an object have opposite semantic information, i.e., $\forall v_i \in V, \exists v_j \in V,$ if v_i *contradicts* v_j, then v_i is True $\Rightarrow$ v_j is False. It is worth noting that other semantic associations can also be applied to the construction of rule KG according to requirements.

Example 1. In Fig. 1, given $<r_1.b$, *belong to*, $l_1>$, since $r_1.b$ *entails* $r_4.b$, then $<r_1.b$, *belong to*, $l_1>$ can increase the likelihood of $<r_4.b$, *belong to*, $l_1>$. Meanwhile, since $r_1.b$ *contradicts* $r_2.b$, $\langle r_1.b$, *belong to*, $l_1\rangle$ will decrease the likelihood of $<r_2.b$, *belong to*, $l_1>$. Moreover, if $r_1.b$ is *neutral* with $r_5.b$, $\langle r_1.b$, *belong to*, $l_1\rangle$ cannot influence the likelihood of $<r_5.b$, *belong to*, $l_1>$.

Definition 3 (Rule matching). Given a rule r, a sentence s, if $s.SUB = r.SUB$, $s.OBJ = r.OBJ$ and $r.b$ is a subsequence of s, denoted as $r.b \sqsubseteq s$, then s can be labeled by r through pattern matching. Meanwhile, if $s.SUB = r.SUB$, $s.OBJ = r.OBJ$, and matching score $Score(r, s) \geq \delta$, then s can be semantically matched by r. When a rule r matches a sentence s, s can be labeled as l.

Example 2. In Fig. 1, since $r_1.b \sqsubseteq s_1$, r_1 can match s_1. Meanwhile, if the matching score $Score(r_3, s_1) \geq \delta$, r_3 can semantically match s_1.

Research objective: Given a corpus $C = \{s_1, s_2, \ldots, s_n\}$, seed rules R^*, pre-defined labels $L = \{l_1, l_2, \ldots, l_m\}$, candidate rules are mined from corpus C. Then, the rule KG is constructed to model the multi-dimensional semantic associations among rules. Next, KGRED decreases misclassified rule and wrongly matched sentences based on rule KG to discover a final rule set R.

3.2. Overview of KGRED

KGRED adopts rule KG to discover rules through the iterations of core modules: rule generation, rule matching, and active learning (see Fig. 2).

As shown in Algorithm 1, we first mine rules and construct rule KG based on the semantic associations between rule bodies and rule labels. Entities in sentences are replaced by entity types to ensure the scalability of rules. Then, we mine the longest common subsequences (LCS) from sentences as rule bodies (Lines 2–9). Candidate rules whose labels are not determined are mined from unlabeled corpus. Consequently, rule KG will be constructed based on seed rules, candidate rules, and pre-defined rule labels (Line 11). In rule KG, the initial *belong to* associations are constructed between the bodies and labels of seed rules. The *entail* and *contradict* associations are constructed by existing natural language inference methods (Gardner et al., 2018), which predicts the associations between texts through a decomposable attention model.

Then, a label-aware rule generation approach is designed to determine the labels of candidate rules (Line 12). The prior knowledge can be propagated from seed rules to candidate rules through neighboring and common information propagation. Then, pattern

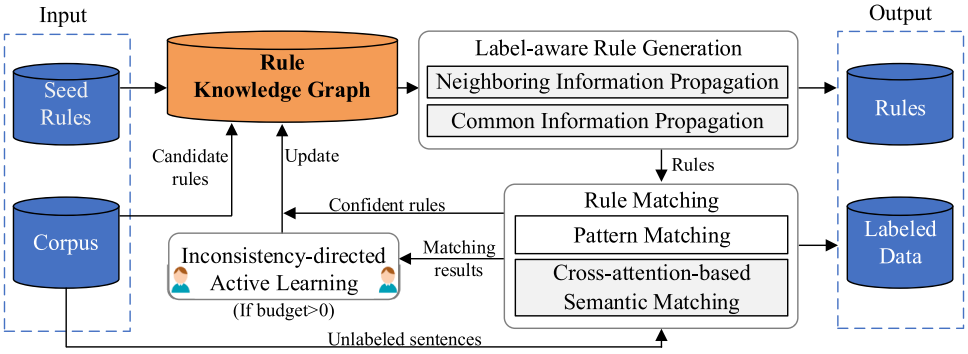

Fig. 2. Overview of the KGRED framework.

Algorithm 1
KGRED(R, R^*, L, C, b, t^*).

Require: Rule set R; Seed rule set R^*, Label set L; Corpus C; Budget b; Maximum number of iterations t^*
Ensure: The final rule set R, and labeled sentences S_t
1: $t = 0$;
2: Function Mine(C): // Mine rules.
3: $R \leftarrow LCS(C)$; // Mine the longest common sequences.
4: $R' \leftarrow R$;
5: While $R' \neq \emptyset$:
6: $R' \leftarrow LCS(R')$;
7: $R \leftarrow R \cup R'$;
8: $R_c \leftarrow Mine(C)$; //Mine candidate rules.
9: While $t \leq t^*$:
10: $G_t \leftarrow ConstructKG(R^*, R_c, L)$; // Construct rule KG.
11: $R_t \leftarrow GenerateRule(G_t)$; // Rule generation.
12: $S_t \leftarrow MatchSentence(R_t, G_t)$; //Rule matching.
13: $R'_t \leftarrow Vote(S_t)$; //Verify rules based on matching results.
14: For r_i in R_t:
15: If the label of r_i is consistent in R_t and R'_t:
16: $R \leftarrow R \cup r_i$; //Update rule KG based on confident rules.
17: If $b > 0$:
18: $R^*_t \leftarrow ActiveLearning(R_t \backslash R)$; //Verify inconsistent rules.
19: $R \leftarrow R \cup R^*_t$;
20: $b = b - |R^*_t|$;
21: $t = t + 1$;
22: return R and S_t

matching and cross-attention-based semantic matching mechanisms are adopted to match sentences. In the semantic matching mechanism, the weights of words in sentences are adaptively adjusted according to rules to weaken the influence of semantically unrelated words. Meanwhile, the semantic information of rules is enriched based on rule KG (Line 13). Consequently, the labels of rules can be also predicted by matched sentences (Line 14). Then, we update rule KG using confident rules whose labels are consistent in rule generation and matching modules (Lines 15–17). It is worth noting that human participation is optional. If the budget $b > 0$, we conduct the inconsistency-directed active learning strategy to select potentially misclassified rules for human annotation. Finally, the annotation results will be fed back to the rule KG (Lines 18–21). The updated KG will be used for the next iteration of rule discovery.

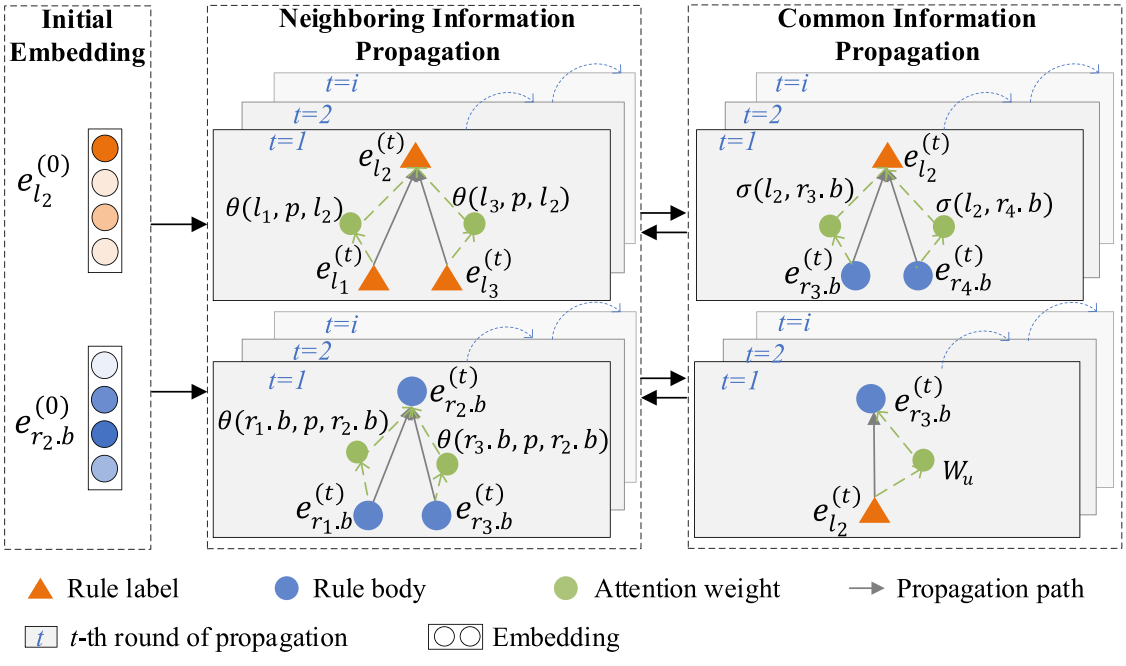

Fig. 3. Neighboring and common information propagation in rule KG.

4. KGRED method

4.1. Label-aware rule generation

Since multi-dimensional semantic associations among rules may impose conflicting semantic constraints on rule labels, we determine the weights of information propagation between nodes based on attention mechanism. However, using the associated neighbor rule bodies can only achieve neighboring information propagation. Rule labels can aggregate the common semantic information of rules that belong to the same label. Therefore, we propose a label-aware rule generation approach to realize neighboring information propagation and common information propagation among rule bodies and labels (Fig. 3).

Specifically, we first obtain the embedding of the rule label as e^l_m. Then, we compute the embedding of a rule body $e_{r_i.b}$ as the average of the embeddings of sentences from which this rule body is mined. Next, neighboring information propagation and common information propagation is conducted based on rule KG.

4.1.1. Attentive information propagation

According to Definition 2, rule KG is composed of two kinds of nodes: rule bodies and rule labels. Neighboring information propagation focuses on propagating neighboring semantic information through the associations among rule bodies, as well as among rule labels, i.e., P_{B-B} and P_{L-L}. Meanwhile, common information propagation focuses on propagating common semantic information of the same types of rules through the associations among rule bodies and labels, i.e., P_{B-L}.

Neighboring information propagation. To distinguish the weight of each semantic association, we aggregate the semantic information of neighbor rules by calculating relational attention weights. In rule body layer, as for the target rule body $r_i.b$, $N_i = \left\{ \left(r_i.b, p_j, r_n.b \right) \middle| \left(r_i.b, p_j, r_n.b \right) \in G \right\}$ is the set of neighbor rule bodies of $r_i.b$. We compute the linear combination of $r_i.b$'s neighbor rule bodies:

$$e_{N_i} = \sum_{\left(r_i.b, p_j, r_n.b \right) \in N_i} \theta\left(r_i.b, p_j, r_n.b \right) e_{r_n.b} \tag{1}$$

$$\theta'\left(r_i.b, p_j, r_n.b \right) = \left(W_p e_{r_n.b} \right)^T \tanh\left(W_p e_{r_i.b} + e_{p_j} \right) \tag{2}$$

$$\theta\left(r_i.b, p_j, r_n.b \right) = \frac{\exp\left(\theta'\left(r_i.b, p_j, r_n.b \right) \right)}{\sum_{\left(r_i.b, p'_j, r_n.b' \right) \in N_i} \exp\left(\theta'\left(r_i.b, p'_j, r_n.b' \right) \right)} \tag{3}$$

where $\theta\left(r_i.b, p_j, r_n.b \right)$ is the relational weight to indicate how much neighboring semantic information has been propagated from neighbor rule bodies to target rule body through associations p_j. $W_p \epsilon \mathbb{R}^{d' \times d}$ is a trainable weight matrix. Through Eq. (2), we assign high attention scores to neighbor rule bodies that are close to the target rule body. Through normalization processing (Eq. (3)), the association weights are obtained. Then, we aggregate the neighbor rule bodies' representations e_{N_i} to the target rule body $r_i.b$:

$$e_{r_i.b} = f\left(W_r \left(e_{r_i.b} + e_{N_i} \right) \right) \tag{4}$$

where we set the activation function $f()$ as LeakyReLU (Dubey & Jain, 2019). W_r is a trainable weight matrix. The neighboring information propagation is also conducted in the rule label layer to generate the target label's representation e^l_m. For example, in Fig. 3, we adopt the semantic associations to propagate semantic information from $r_1.b$ and $r_3.b$ to $r_2.b$ in the rule body layer. Since $r_2.b$ *entails* $r_3.b$, and $r_2.b$ *contradicts* $r_1.b$, the weight of $r_3.b$ is higher than that of $r_1.b$ in representing the semantic information of $r_2.b$.

Common information propagation. To combine semantic information of the rule bodies and labels, we use a self-attention mechanism to aggregate common information contained in rule bodies that belong to the same labels:

$$e_{l_m} = f\left(W_l \left(e_{l_m} + \sum_{r_i.b \in R_{l_m}} \sigma(l_m, r_i.b) e_{r_i.b} \right) \right) \tag{5}$$

$$\sigma(l_m, r_i.b) = \frac{\exp\left(W_c^T \tanh\left(B e_{r_i.b} \right) \right)}{\sum_{r_i.b' \in R_{l_m}} \exp\left(W_c^T \tanh\left(B e_{r_i.b'} \right) \right)} \tag{6}$$

where $\sigma(l_m, r_i.b)$ is a weight function to indicate how much common semantic information has been propagated from rule bodies to rule labels. B, W_l and W_c are learnable model parameters for integrating semantic information of rule bodies to rule labels. Then, to enrich the representation of rule bodies, we inject the semantic constraints between rule labels into the representation of rule bodies.

$$e_{r_i.b} = f\left(W_u \left(e_{r_i.b} + e_{l_m} \right) \right) \tag{7}$$

where W_u is the trainable weight matrix for the information integration of rule labels and rule bodies. For example, in Fig. 3, through the common information propagation, we can aggregate the semantic information of r_3 b and r_4 b to l_2. Then, the semantic information of l_2 can be integrated into r_3 b. The common information propagation can enhance semantic similarity among rules of the same type and boost differentiation between rules of different types.

Moreover, we obtain the multiple-hop semantic information from rule KG by stacking multiple rounds of propagation. Formally, in the t-th round of propagation, the representation of rule bodies can be defined as:

$$e_{r_i.b}^{(t)} = f\left(e_{r_i.b}^{(t-1)} + e_{N_i}^{(t-1)}\right) \tag{8}$$

Finally, we choose the label with the highest similarity to the target rule body as the rule's true label:

$$l_{mod}(r_i.b) = argmax_{(l_m \in L)} \left(e_{r_i.b}\right)^T e_{l_m} \tag{9}$$

4.1.2. Optimization

To optimize the results of rule generation, the objective functions are defined as follows:

$$loss_{total} = loss_{nei} + loss_{com} \tag{10}$$

$$loss_{nei} = \sum_{(i,j) \in P_e} dis(e_i,\ e_j) - \sum_{(i,j) \in P_c} dis(e_i,\ e_n) \tag{11}$$

$$loss_{com} = Max\left(0, \left[-dis\left(e_{r_i.b}, e_{l_m}\right) + dis\left(e_{r_i.b}, e_{l'_m}\right) + \gamma\right]\right) \tag{12}$$

where $dis()$ is a function to compute the semantic distance between two vectors, such as Euclidean distance. $loss_{nei}$ aims to minimize the distance between nodes with *entail* associations (P_e) and maximize the distance between nodes with *contradict* associations (P_c). l'_m denotes the true label of r_i. l_m denotes other labels except for l'_m. $loss_{com}$ aims to minimize the distance between the rule bodies and their true labels.

4.2. Cross-attention-based semantic matching

Since sentences contain much unrelated semantic information with regard to rules, resulting the low semantic similarity between rules and sentences. Therefore, we design a cross-attention mechanism to adaptively weaken the influence of semantically unrelated words in sentences (see Fig. 4). Specifically, we obtain rule embeddings from the rule generation approach. Then, we propose sentence-to-rule (S-R) attention to adaptively refine the semantic information of sentences. Meanwhile, based on rule-to-rule (R-R) attention, we utilize the neighbor rules to enrich the semantic information of rules.

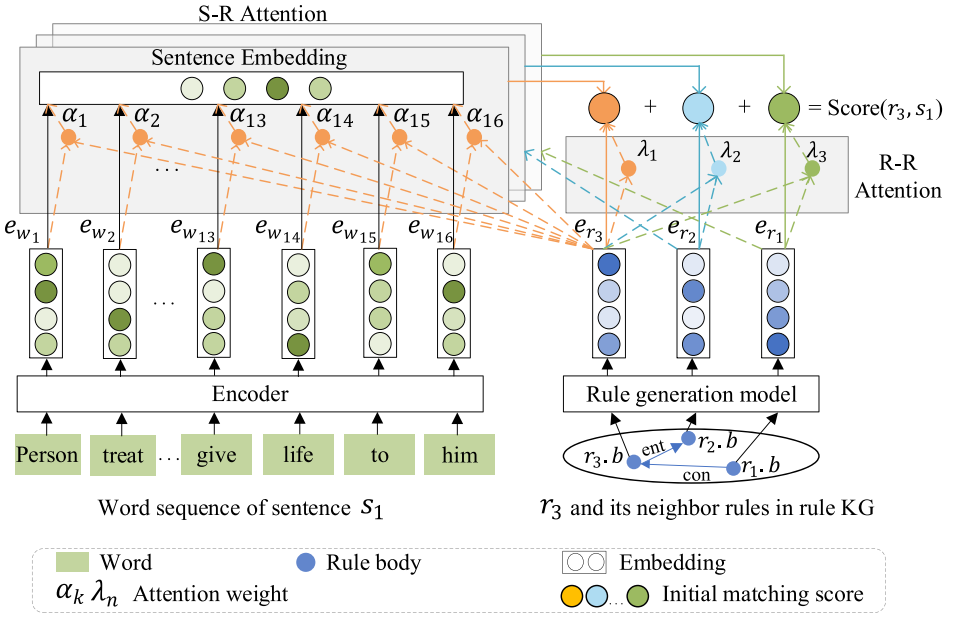

Fig. 4. Architecture of cross-attention-based semantic matching.

4.2.1. S-R attention

To adaptively refine the semantic information of sentences, the weight of each word in sentences can be measured by the semantic similarity between the word and the target rule:

$$\alpha_k = \frac{\exp\left(\tanh\left(W_b^T dis\left(e_{r_i},\ e_{w_k}\right) + b\right)\right)}{\sum_{w_k \in s_j} \exp\left(\tanh\left(W_b^T dis\left(e_{r_i},\ e_{w_k}\right) + b\right)\right)} \tag{13}$$

where α_k denotes the attention weight of the word w_k in s_j. W_b is an intermediate matrix and b is an offset.

Subsequently, the attention weights of words are employed to calculate the semantic vector of sentence s_j:

$$e_{s_j} = \sum_{w_k \in s_j} \alpha_k e_{w_k} \tag{14}$$

The initial matching score of r_i and s_j can be defined as the semantic similarity between them:

$$Score'\left(r_i,\ s_j\right) = Sim\left(e_{r_i},\ e_{s_j}\right) \tag{15}$$

where we adopt the cosine similarity as $Sim()$. In Fig. 4, if we calculate the initial matching score of r_3 and s_1, the weight of each word of s_1 depends on the semantic similarity between this word and r_3.

4.2.2. R-R attention

In R-R attention, we adopt neighbor rules to help match additional semantic information from the sentence. Specifically, we first calculate the initial matching scores of neighbor rules and the sentence. Then, we aggregate the matching score of the target rules and neighbor rules based on the semantic associations of rule KG:

$$Score\left(r_i,\ s_j\right) = hScore'\left(r_i,\ s_j\right) + (1-h)Score'\left(N_i,\ s_j\right) \tag{16}$$

$$Score'\left(N_i,\ s_j\right) = \sum_{(i,p,n) \in N_i} \lambda(r_i, p, r_n) Score'\left(r_n,\ s_j\right) \tag{17}$$

$$\lambda(r_i, p, r_n) = \frac{\exp(W_s e_{r_n})^T \tanh\left(W_s e_{r_i} + e_p\right)}{\sum_{\left(r_i, p', r'_n\right) \in N_i} \exp\left(W_s e_{r'_n}\right)^T \tanh\left(W_s e_{r_i} + e_{p'}\right)} \tag{18}$$

where h denotes the contribution of the initial matching score of the neighbor rule to the final matching score. $\lambda(r_i, p, r_n)$ is a relational weight to indicate the similarity of neighbor rules with r_i. For example, in Fig. 4, by aggregating the initial matching scores of $r_1,\ r_2$ and r_3, the final matching scores $Score(r_3,\ s_1)$ can be calculated.

4.2.3. Optimization

We utilize seed rules and their matched sentences for the optimization of rule matching. If the labels of rules are the same as the matched sentences, we take these rules and sentences as correct matching pairs $\left(r_i,\ s_j\right)$. Then, we use *contradicted* rules to generate wrong matching pairs $\left(r_f,\ s_j\right)$. To guarantee the scores of correct matching pairs $\left(r_i,\ s_j\right)$ are higher than the wrong matching pairs $\left(r_f,\ s_j\right)$, the training loss is given as follows:

$$loss_{matching} = Max\left(0, \left[-Score\left(r_i,\ s_j\right) + Score\left(r_f,\ s_j\right)\right] + \pi\right) \tag{19}$$

where π is the range between correct and wrong matching pairs. We adopt stochastic gradient descent (Bottou, 2012) to minimize the learning process.

Through pattern matching and semantic matching mechanisms, the matching results can be obtained. We assign the label with the most matching rules to s_j. Moreover, we choose the most frequent label in the matched sentences for r_i, i.e., $l_{sen}(r_i)$. When the labels of rules are consistent in rule generation and matching modules, i.e., $l_{mod}(r_i) = l_{sen}(r_i)$, we take these rules as confident rules. Then, these confident rules are utilized to update rule KG.

4.3. Inconsistency-directed active learning

Due to insufficient prior knowledge, we utilize active learning to detect misclassified rules for human verification. These misclassified rules often have inconsistent performance in rule discovery: (1) In rule generation, the labels of rules are not consistent with rule bodies. Specifically, rules that belong to *contradicted* labels share similar semantic information in rule bodies. Meanwhile, rules that belong to the same or *entailed* labels have low semantic similarity in rule bodies. (2) In rule matching, the labels of matched sentences are not consistent with that of rules. To detect these inconsistent rules, we employ unsupervised contrastive learning techniques to calculate the semantic distance between rules, as well as between rules and sentences. Then, we design active learning functions to verify inconsistent rules in rule generation and matching.

Specifically, we construct an input text set based on the bodies of rules and matched sentences. Then, we adopt data augmentation

strategies (Yan et al., 2021) to generate different embeddings of the same input texts as e_i and e_j. Then, we adopt the normalized temperature-scaled cross-entropy loss (Chen et al., 2020) as the contrastive learning objective to make embeddings of the same texts closer and embeddings of different texts further apart:

$$loss_{con(i,j)} = -log\frac{\exp(dis(e_i, e_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k\neq 1]}\exp(dis(e_i, e_j)/\tau)} \tag{20}$$

where 1 is an indicator function and τ represents a temperature parameter. Then, based on the representations of rules and sentences, we can model the inconsistent rules in rule generation and rule matching modules.

Inconsistency in rule generation. By comparing rules with their neighbor rules in rule KG, we divide neighbor rules into two categories: Neighbor rules with *contradicted* labels are *contradicted* rules. Neighbor rules with *entailed* or the same labels are *entailed* rules. Then, we evaluate the inconsistency score of rules in rule generation as follows:

$$RC(r_i) = \frac{Min_{r_j\in N_{ent}(r_i)} dis(r_i,\ r_j)}{Max_{r_x\in N_{con}(r_i)} dis(r_i,\ r_x)} \tag{21}$$

where $N_{con}(r_i)$ and $N_{ent}(r_i)$ represent *contradicted* and *entailed* rules. The $Min()$ function obtains the minimum distance between the target rule and *entailed* rules. The $Max()$ function obtains the maximum distance between the target rule and *contradicted* rules. For example, in Fig. 1, r_2 and r_3 are *contradicted* rules for r_1, and r_1 is further away from r_3 than r_2. Meanwhile, r_4 is *entailed* rule for r_3. Therefore, $RC(r_3) = dis(r_3,\ r_4)/dis(r_3,\ r_1)$.

Inconsistency in rule matching. By comparing the labels of rules and matched sentences, we can obtain the inconsistency score of rules in rule matching as follows:

$$RM(r_i) = \frac{Min_{s_n\in S_c(r_i)} dis(r_i,\ s_n)}{Max_{s_m\in S_w(r_i)} dis(r_i,\ s_m)} \tag{22}$$

where $S_c(r_i)$ and $S_w(r_i)$ represent correctly and wrongly matched sentences of r_i respectively. The $Min()$ function denotes the minimum distance between rules and correctly matched sentences. The $Max()$ function obtains the maximum distance between rules and wrongly matched sentences. For example, if r_3 wrongly matched s_1 and s_2 and $dis(r_3,\ s_1) \geq dis(r_3,\ s_2)$. Meanwhile, r_3 correctly matched s_3 and $s4$ and $dis(r_3,\ s_3) \geq dis(r_3,\ s_4)$. Then, $RM(r_3) = dis(r_3,\ s_2)/dis(r_3,\ s_3)$.

Finally, the inconsistency score of r_i can be calculated as:

$$r_i = argmax_{r_i\in R} RC(r_i)RM(r_i) \tag{23}$$

where $argmax()$ aims to select rules with the highest inconsistency score for human verification. The feedback is utilized to update rule KG for later rule discovery.

5. Experiments

In this section, we introduce the datasets and compared baselines. Then, we present the detailed experimental results and case study with analysis.

5.1. Settings

Datasets. We adopt widely-used sentence-level relation extraction datasets in our experiments: (1) SemEval 2010 Task 8 contains about 10,000 sentences with 19 relation types (Hendrickx et al., 2010). (2) Wiki80 contains about 56,000 sentences with 80 relation types (Hendrickx et al., 2010).

Weakly supervised learning baselines. To prove the effectiveness of KGRED, we compare following methods:

(1) **Rule labeling**. In this method, rules match sentences to mine new rules. This method serves as a lower bound for rule-based labeling methods.
(2) **Snorkel**. In Snorkel, rules are designed by experts through a data programming paradigm. Then the labeled data can be obtained by probabilistic generative models (Ratner et al., 2017).
(3) **NERO**. It requires human annotation to generate a rule set and then implements soft matching by calculating the semantic similarity between rules and sentences. Finally, joint learning of rules and sentences is used to train the relation extraction model (Zhou et al., 2020).
(4) **Darwin**. It constructs a rule hierarchy tree by analyzing subsequence relations among rules. Then, it proposes three traversal strategies to select potential rules for human verification. Through the feedback of experts, the rule hierarchy tree can be updated for later traversal (Galhotra et al., 2021).

Variants of KGRED. To evaluate the key modules of KGRED, we design following ablation experiments: (1) **w/o rule generation** (w/o RC). This method removes the label-aware rule generation approach. Then, the labels of rules are determined by matched

sentences. (2) **w/o semantic matching** (w/o SM). This method removes the cross-attention-based semantic matching module and only performs pattern matching to match sentences. (3) **w/o active learning** (w/o AL). It removes the inconsistency-directed active learning module.

Rule generation baselines. We evaluate the label-aware rule generation approach of KGRED with following methods: (1) **KGAT**. This method realizes attention-based information propagation in KG. It calculates the weights of neighbor nodes based on the relations of target node and neighbor nodes (Wang et al., 2019). In rule generation, we run the KGAT method on the rule KG and exploit the semantic associations among rule bodies for information propagation. (2) **Hierarchical passing**. This method proposes a hierarchical information propagation method to make the node representation learning process aware of long-range interactive information (Zhong et al., 2023). In rule generation, we conduct this method based on the semantic associations among rule bodies and labels.

Rule matching baselines. We compare the cross-attention-based semantic matching mechanism of KGRED with the following rule matching methods: (1) **Pattern matching**. Pattern matching is to determine whether the patterns of rules and sentences are consistent. In this experiment, we follow the common practice of converting rules to regular expressions to match sentences. (2) **Soft matching**. Soft rule matcher of NERO obtains sentence and rule embeddings based on self-attention mechanism (Zhou et al., 2020).

Active learning baselines. We evaluate inconsistency-directed active learning strategy (CO_AL) with other methods: (1) **Uncertainty-based active learning** (UC_AL). It utilizes entropy to measure the uncertainty of unlabeled data (Holub et al., 2008). (2) **Contrastive active learning** (CL_AL). It uses a contrastive learning method to calculate the distinctive and similar scores of unlabeled data (Du et al., 2023).

Implementation. We implement baselines from scratch using Tensorflow 1.12.0 except for those that have released their codes. Pre-trained BERT model is utilized to initialize word embeddings. Specifically, we use 5 % labeled data of datasets to generate seed rules. The length of rule body is set to be longer than 2. Then, we conduct 8 iterations of rule discovery for each dataset. In each iteration, the top 5 % of inconsistent rules are verified until human resource budget is used up. In human verification, each selected rule is verified by three annotators. For a fair comparison, we conduct baselines based on the same human resource and labeled sentences as KGRED. Following the common practice, precision, recall, and F1 score (Zhou et al., 2020; Galhotra et al., 2021) are adopted to evaluate KGRED and compared baselines. Five training and testing runs are conducted based on different random seed rules. Then, the mean and standard deviation of the evaluation metrics are presented.

5.2. Experimental results

Performance of baselines. Table 1 shows the performance of KGRED and the baselines on the SemEval and Wiki80 datasets. Compared to other methods, rule labeling method achieves lower F1 score. Since rule labeling method lacks a verification strategy for generated rules and labeled sentences during iterations, it is difficult to prevent semantic drift. Instead, KGRED outperforms Snorkel by 6.7 % F1 score on the SemEval dataset and 10.3 % F1 score on the Wiki80 dataset. This is because Snorkel relies on experts to write rules. Due to the limited domain knowledge of experts, some important rules remain to be discovered. Meanwhile, KGRED outperforms NERO by 5.5 % F1 score on the SemEval dataset and 5.1 % F1 score on the Wiki80 dataset. NERO utilizes a self-attention mechanism to calculate the semantic similarity of sentences and rules, which can improve the coverage of rules. However, it ignores the mutual influence between semantic information of rules and sentences, which may generate errors in rule matching. Moreover, Darwin has a higher recall score than KGRED, but a lower precision score on the SemEval dataset. Since Darwin uses subsequence relations among rules to generate rules, it tends to choose rules with a high degree of generalization, resulting in a higher recall score for data annotation. However, Darwin does not consider the semantic information of rules and sentences in rule matching, so it cannot avoid incorrectly matched sentences. It is worth noting that KGRED can achieve a better balance between precision and recall scores than other methods across datasets. This is because KGRED generates rules from the entire corpus to ensure the coverage of the rule set. Moreover, the multi-dimensional semantic associations among rules are used to reduce errors in rule labeling.

Furthermore, KGRED outperforms its variants. Specifically, w/o RC uses sentences matched by rules to determine rule labels. Therefore, it cannot reduce misclassified rules that are generated from wrongly matched sentences in rule discovery. Meanwhile, w/o SM ignores the semantic similarity between rules and sentences, which may introduce wrongly matched sentences in rule discovery. In w/o AL, misclassified rules cannot be verified by human verification, resulting in lower F1 score than KGRED.

Furthermore, we analyze the intermediate results of rule discovery. In Fig. 5, the F1 scores of rule generation and rule matching fluctuate in some rounds. Due to the limited number of seed rules, rule generation is prone to errors. These errors may affect rule

Table 1
Results of compared baselines.

Method	SemEval			Wiki80		
	Precision	Recall	F1	Precision	Recall	F1
Rule labeling	22.7 ± 5.1	15.6 ± 4.0	17.4 ± 1.1	28.6 ± 6.7	38.0 ± 6.6	32.5 ± 6.7
Snorkel	58.6 ± 3.0	57.5 ± 2.7	58.0 ± 0.1	56.6 ± 3.2	57.1 ± 5.6	56.8 ± 4.4
NERO	61.0 ± 5.1	58.0 ± 2.2	59.2 ± 1.3	61.1 ± 4.0	63.1 ± 6.8	62.0 ± 5.3
Darwin	28.4 ± 0.2	**74.0 ± 13.1**	41.0 ± 0.8	41.6 ± 2.1	55.8 ± 4.8	44.1 ± 5.1
w/o RC	30.5 ± 5.8	34.1 ± 2.0	31.7 ± 2.4	35.5 ± 0.5	40.4 ± 2.1	37.5 ± 1.9
w/o SM	49.8 ± 5.8	51.7 ± 6.3	50.7 ± 6.1	35.3 ± 1.9	41.9 ± 4.1	38.3 ± 1.1
w/o AL	38.0 ± 13.3	39.2 ± 14.6	37.8 ± 13.0	33.0 ± 0.2	42.8 ± 0.4	37.2 ± 1.7
KGRED	**63.0 ± 2.4**	**66.7 ± 5.7**	**64.7 ± 3.9**	**64.3 ± 7.8**	**70.5 ± 1.8**	**67.1 ± 5.1**

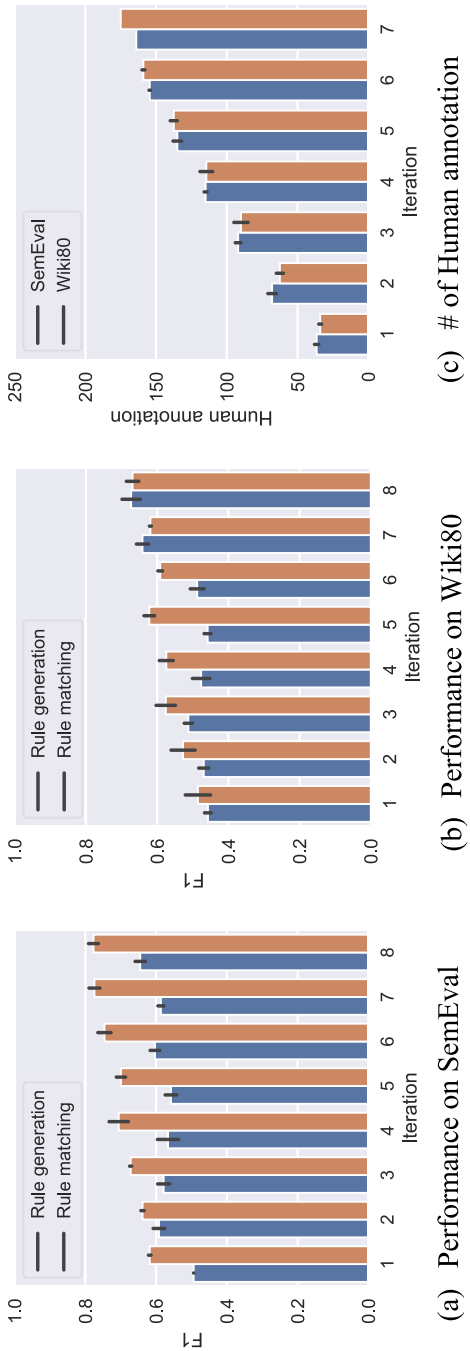

Fig. 5. Iteration experiments of KGRED.

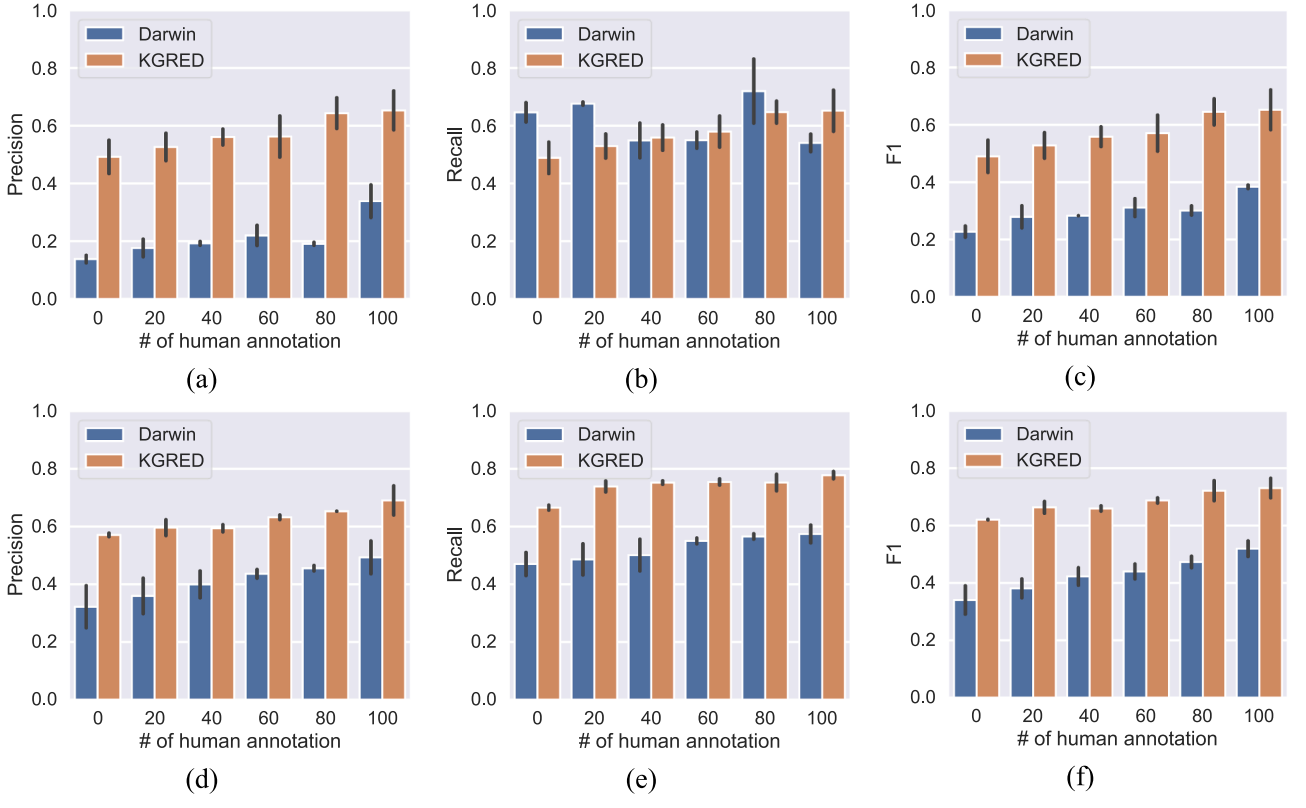

Fig. 6. Comparison of different amounts of human annotation. (a-c) for SemEval. (d-e) for Wiki80.

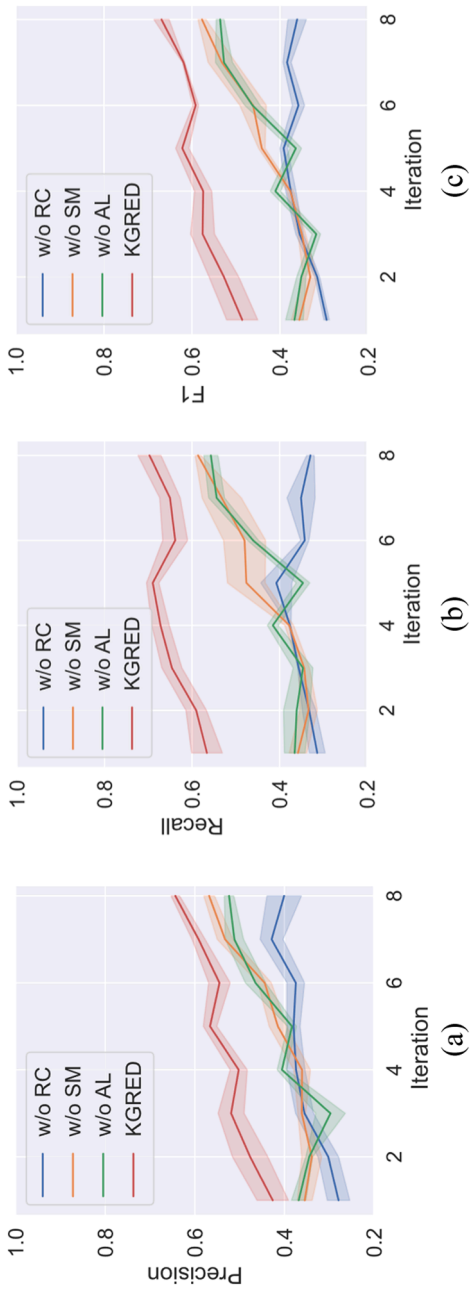

Fig. 7. Comparison of variants of KGRED.

matching in the early stages of rule discovery. Through the inconsistency-directed active learning method, the previously misclassified rules can be verified to improve the quality of rule discovery. Moreover, Fig. 5(c) shows the cumulative amount of human annotation in iterations. In the 4–7th iterations, the amount of human annotation decreases gradually. As the quality of rule generation and rule matching improves in rule discovery, the inconsistent rules decrease.

Effect of human annotation on rule discovery. Fig. 6 shows the changes in Darwin and KGRED when varying the amount of human annotation. When the amount of human annotation is zero, KGRED achieves 26 % higher F1 score on the SemEval dataset and 28 % higher F1 score on the Wiki80 dataset compared to Darwin. It proves the effectiveness of KGRED in rule discovery without human annotation. As the amount of human annotation increases, although Darwin has achieved a recall score comparable to or even better than KGRED, its precision score is lower than KGRED. The reason is that Darwin tends to choose rules that can match more positive samples while ignoring the semantic information of the rules. Rules may contain some common words, resulting in a high recall score but not relevant to the target task. Compared with zero human annotation, when the amount of human annotation reaches 100, the F1 score of KGRED has been improved by at least 21 %. KGRED adopts semantic associations among rules to more accurately represent the semantics of rules, helping to achieve higher quality rules with limited labor costs.

Performance of variants of KGRED. Fig. 7 demonstrates the performance of KGRED's variants on the SemEval dataset in iterations. Variants of KGRED are affected by semantic drift and show fluctuations. Specifically, the F1 score of w/o RC method shows a downward trend. It reflects the effectiveness of the rule generation model in predicting correct labels for generated rules. For the w/o AL method, due to the lack of an active learning strategy, the rule iteration process is easily affected by noise, resulting in large fluctuations in F1 score. For the w/o SM method, in the fifth to sixth iterations, the recall score of the w/o SM has a higher error than other methods. Since rule matching of the w/o SM has limited coverage and is not learnable, some wrongly matched sentences are introduced.

Effect of different rule generation approaches. Table 2 summarizes the performances of different rule generation approaches. Specifically, KGRED outperforms KGAT by 27 % F1 score. KGAT utilizes multiple semantic associations in the rule KG to realize the propagation of semantic information between neighbor nodes. However, it is difficult to aggregate semantic association information, which is shared in the same type of rules. Meanwhile, KGRED outperforms the hierarchical passing method by 11.4 % F1 score. The hierarchical passing method can utilize the *belong to* associations among rule bodies and labels to realize the hierarchical semantic information propagation. However, it is unable to adopt the *entail* and *contradict* associations to realize relation-weighted information propagation among rule bodies and labels. Compared to other approaches, KGRED propagates neighboring semantic information and common semantic information of rules simultaneously.

Effect of different rule matching mechanisms. Table 3 summarizes the experimental results of different rule matching mechanisms. Pattern matching mechanism examines the pattern of sentences to determine sentences' labels. However, it ignores the semantic information of the sentence, which achieves lower precision score than KGRED. Compared to pattern matching, the soft matching mechanism of NERO can achieve higher recall score. However, wrongly matched sentences will be introduced because NERO focuses on the local semantic information of sentences and ignores the mutual influence between rules and sentences. In contrast, cross-attention-based semantic matching mechanism can adaptively calculate the weights of words in sentences according to rules while enriching the semantic information of rules based on rule KG. Through this mechanism, the unbalanced semantic information between sentences and rules can be mitigated, thereby improving the F1 score of rule matching.

Effect of different active learning strategies. Fig. 8 summarizes the performances of different active learning strategies. UC_AL performs weaker than other methods. Since UC_AL utilizes the confidence of the rule generation model to select samples, it is easy to be influenced by model's overconfident predictions. Meanwhile, there are large fluctuations in the F1 score of CL_AL. This method uses a contrastive learning method to select related and informative samples to increase labeled samples' diversity. However, it cannot evaluate the possibility that a rule label is wrong in rule discovery. Instead, inconsistency-directed active learning method comprehensively considers the entire process of rule discovery to find misclassified rules in rule generation and rule matching for human verification, which can effectively prevent noise propagation in rule discovery.

Effect of different loss functions in rule generation. As shown in Table 4, all two loss functions contribute to the final performance of rule generation, while $loss_{nei}$ helps the most. When removing $loss_{nei}$, the F1 score drops 15 %. It proves the effectiveness of propagating neighboring semantic information based on multi-dimensional semantic associations among rule bodies, as well as among rule labels. When removing $loss_{con}$, the F1 score drops 5.3 %, which shows the effectiveness of propagating common semantic information among rule bodies and labels.

5.3. Case study

By comparing the discovered rules and matched sentences (Table 5), we can obtain an intuitive understanding of KGRED. The rules discovered by Snorkel and NERO tend to have fixed patterns. Both Snorkel and NERO rely on experts to discover rules. Since experts

Table 2
Comparison of different rule generation approaches.

Methods	Precision	Recall	F1
KGAT	40.8 ± 5.6	40.9 ± 2.3	38.2 ± 0.7
Hierarchical passing	51.2 ± 2.1	57.3 ± 5.9	53.8 ± 1.4
Label-aware rule generation (KGRED)	65.8 ± 0.4	63.1 ± 0.7	65.2 ± 0.5

Table 3
Comparison of different rule matching mechanisms.

Methods	Precision	Recall	F1
Pattern matching	57.4 ± 1.7	69.5 ± 4.5	62.8 ± 0.9
Soft matching	42.8 ± 8.8	74.1 ± 5.7	53.2 ± 5.5
Cross-attention-based rule matching (KGRED)	**63.0 ± 2.4**	**66.7 ± 5.7**	**64.7 ± 3.9**

are difficult to cover the entire corpus, some rules remain to be discovered. Meanwhile, Darwin discovers many redundant rules, since it utilizes subsequences of labeled sentences as rules. Compared to the baselines, the rules discovered by KGRED have greater diversity in pattern and semantics. KGRED considers the entire corpus comprehensively to discover rules. Rule labels are effectively predicted based on the multi-dimensional semantic associations among rules. Therefore, KGRED can discover rules that cannot be discovered by other methods. As for True Positive (TP) sentences, NERO matches more TP sentences than Snorkel and Darwin. This is because NERO adopts soft matching mechanism to extend the coverage of rules. However, NERO matches fewer TP sentences than KGRED. The diverse rules discovered by KGRED can match related sentences comprehensively. Moreover, KGRED leverages cross-attention-aware semantic matching mechanism to reduce wrongly matched sentences.

Rule generation based on rule KG. Fig. 9 shows the processing of determining the labels of rules based on rule KG on the SemEval dataset. In Fig. 9, since $r_5.b$ *entails* $r_1.b$, r_1 is related to the "Product_producer" label. Since $r_1.b$ *contradicts* $r_3.b$, $r_4.b$ and $r_6.b$, the label of r_1 is less related with "Entity_origin" or "Entity_destination". Finally, according to the label-aware rule generation approach, the label of r_1 is determined as "Product_producer". Through label-aware rule generation approach, the multi-dimensional semantic associations among rules help to predict the labels of rules while providing interpretability for rule generation.

Rule matching based on rule KG. Fig. 10 shows the processing of semantic matching based on rule KG on the SemEval dataset. In Fig. 10, s_8 contains other semantically unrelated words according to r_3 and r_7. If calculating the semantic similarity between r_3 and s_8, or between r_7 and s_8 in isolation, s_8 is difficult to be semantically matched. According to the *entail* association between the bodies of r_3 and r_7 in the rule KG, the semantic information of r_7 can be aggregated into r_3, and the semantic matching of r_3 and s_8 can be realized. It demonstrates that cross-attention-based semantic matching mechanism can adaptively adjust the weights of the words in the sentence according to rules while enriching the semantic information of rules based on rule KG.

6. Implications and conclusion

6.1. Implications

This study has the following theoretical implications. First, we model the multi-dimensional semantic associations among rules to reveal the semantic constraints among rules. Second, different from previous works that considered rules in isolation in rule discovery, we propose a new way to discover rules based on rule KG.

In terms of practical implications, this study investigates the possibility of adopting rule KG to address semantic drift in rule discovery. Previous rule discovery methods cannot decrease misclassified rules and wrongly matched sentences simultaneously in rule discovery. Instead, our proposed method utilizes the multi-dimensional semantic associations among rules to realize the joint optimization of rule generation and rule matching. Specifically, the representation of rules in rule generation modules can be utilized in semantic matching. Meanwhile, inconsistent rules are selected and verified by evaluating the performance of rules in rule generation and matching. Moreover, as KGRED requires limited prior knowledge, it can be extended to areas of expertise.

6.2. Conclusion

In conclusion, we construct rule KG to model the multi-dimensional semantic associations among rules and utilize this rule KG to generate high-quality rules to alleviate semantic drift in rule discovery. The experimental results on two public datasets show that KGRED can achieve at least 5.1 % gain in F1 score over the baselines. Specifically, to decrease misclassified rules, we design a label-aware rule generation approach to convert the semantic constraints among rules into attentive information propagation. To reduce the wrongly matched sentences, we present a cross-attention-based semantic matching mechanism to adaptively calculate the semantic similarity of rules and sentences based on the mutual influence between them. Moreover, we use an inconsistency-directed active learning strategy to verify rules in rule discovery. Moreover, the case study proves that KGRED can discover a comprehensive rule set and provide interpretability for rule discovery.

Regarding shortcomings of the proposed method, since the rule KG is updated in each iteration, rule generation and matching models need to be re-trained, which may increase training costs. In the future, we will explore how to incrementally train and optimize these models. Moreover, we will explore the potential of combining rule discovery and large language models in data labeling. By injecting high-quality domain knowledge automatically discovered by KGRED into large language models, the scalability of data labeling can be improved.

CRediT authorship contribution statement

Wenjun Hou: Software, Methodology, Conceptualization. **Liang Hong:** Writing – review & editing, Methodology. **Ziyi Zhu:**

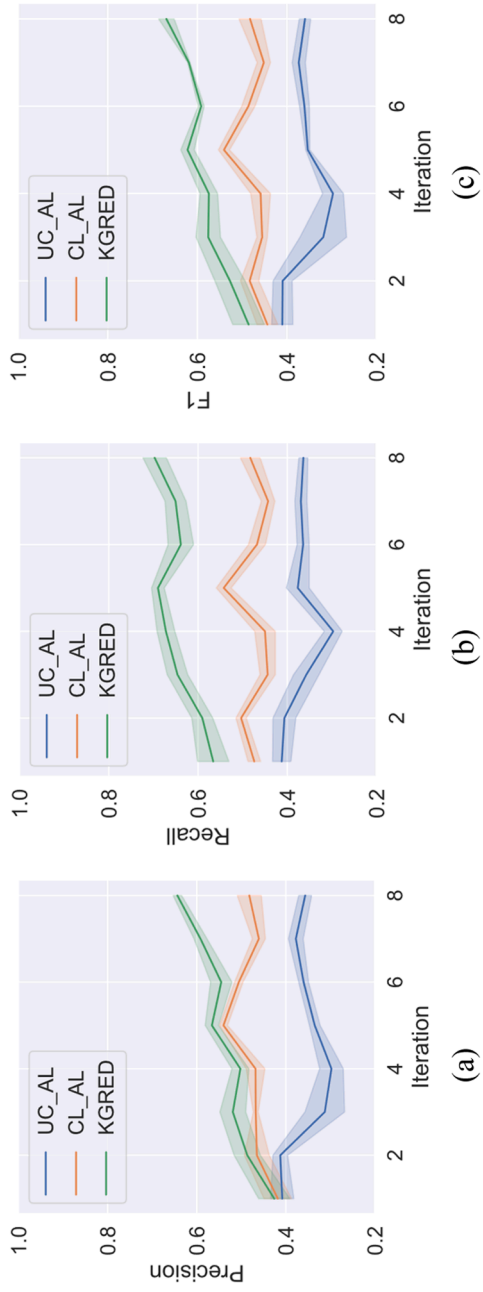

Fig. 8. Comparison of different active learning strategies.

Table 4
Results of optimization objectives in rule generation.

Objective	Precision	Recall	F1
w/o $loss_{nei}$	54.8 ± 1.6	53.7 ± 2.1	54.2 ± 1.9
w/o $loss_{con}$	64.0 ± 1.9	63.7 ± 2.2	63.9 ± 2.0
$loss_{total}$	**65.8** ± **0.4**	**73.1** ± **0.7**	**62.9** ± **0.5**

Table 5
Comparison of discovered rules and matched sentences.

Method	Discovered rules	# of TP sentences
Snorkel	def P_P(x): return Product_producer if "made by" in x else Abstain def P_P(x): return Product_producer if "founded by" in x else Abstain def E_O(x): return Entity_origin if "made from" in x else Abstain def E_O(x): return Entity_origin if "release from" in x else Abstain def E_D(x): return Entity_destination if "into my" in x else Abstain def E_D(x): return Entity_destination if "into the" in x else Abstain	4089
NERO	"SUBJ-O, produced by the, OBJ-O"->Product_producer "SUBJ-O, produces, OBJ-O"->Product_producer "SUBJ-O, from past, OBJ-O"->Entity_origin "SUBJ-O, from outer, OBJ-O"->Entity_origin "SUBJ-O, into my, OBJ-O"->Entity_destination "SUBJ-O, into their, OBJ-O"->Entity_destination	4089
Darwin	"SUBJ-O, produced, OBJ-O" (Product_producer) "SUBJ-O, produced by, OBJ-O" (Product_producer) "SUBJ-O, went away from, OBJ-O" (Entity_origin) "SUBJ-O, went away from the, OBJ-O" (Entity_origin) "SUBJ-O, into, OBJ-O" (Entity_destination) "SUBJ-O, into my, OBJ-O" (Entity_destination)	2395
KGRED	{e1, produce, e2}->Product_producer {e1, e2, construct}->Product_producer {e1, run away from, e2}->Entity_origin {release, e1, e2}->Entity_origin {e1, inside, e2}->Entity_destination {e1, move into, e2}->Entity_destination	4322

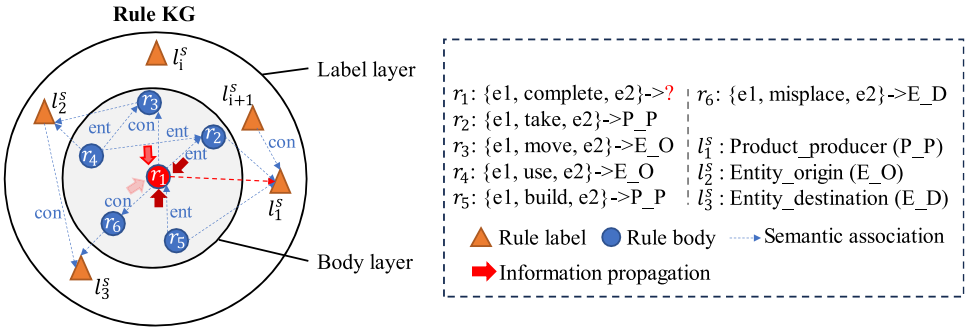

Fig. 9. Cases of rule generation.

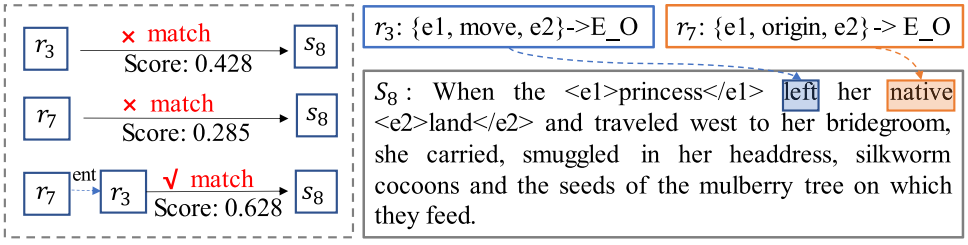

Fig. 10. Cases of semantic matching.

Validation, Software.

Data availability

Data will be made available on request.

Acknowledgements

This work is funded by National Natural Science Foundation of China General Program (Grant No. 72074172).

References

Bottou, L. (2012). Stochastic gradient descent tricks. Neural Networks: Tricks of the Trade: Second Edition, 7700, 421–436. 10.1007/978-3-642-35289-825.

Buchert, F., Navab, N., & Kim, S.T. (2022). Exploiting diversity of unlabeled data for label-efficient semi-supervised active learning. Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), 2063–2069. 10.1109/ICPR56361.2022.9956305.

JulyH. D. III Chen, T., Kornblith, S., Norouzi, M., & Hinton, G (2020). A simple framework for contrastive learning of visual representations. Eds.. In A. Singh (Ed.), *119. Proceedings of the 37th international conference on machine learning* (pp. 1597–1607). PMLR https://proceedings.mlr.press/v119/chen20j.html.

Deng, L., Yang, B., Kang, Z., Yang, S., & Wu, S. (2021). A noisy label and negative sample robust loss function for DNN-based distant supervised relation extraction. *Neural Networks, 139*, 358–370. https://doi.org/10.1016/j.neunet.2021.03.030

Du, P., Chen, H., Zhao, S., Chai, S., Chen, H., & Li, C. (2023). Contrastive active learning under class distribution mismatch. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(4), 4260–4273. https://doi.org/10.1109/TPAMI.2022.3188807

Dubey, A.K., & Jain, V. (2019). Comparative study of convolution neural network's RELU and leaky-RELU activation functions. Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018, 553, 873–880. https://doi.org/10.1007/978-981-13-6772-476.

Feng, X., Guo, J., Qin, B., Liu, T., & Liu, Y. (2017). Effective deep memory networks for distant supervised relation extraction. *IJCAI, 17*, 1–7.

Fries, J. A., Steinberg, E., Khattar, S., Fleming, S. L., Posada, J., Callahan, A., & Shah, N. H. (2021). Ontology-driven weak supervision for clinical entity classification in electronic health records. *Nature Communications, 12*(1), 2017.

Galhotra, S., Golshan, B., & Tan, W.-C. (2021). Adaptive rule discovery for labeling text data. Proceedings of the 2021 International Conference on Management of Data, 2217–2225. 10.1145/3448016.3457334.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N.F., & Zettlemoyer, L. (2018, July). AllenNLP: A Deep Semantic Natural Language Processing Platform. In Proceedings of the Workshop for NLP Open Source Software (NLP-OSS) (pp. 1–6).

Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Séaghdha, D.O., Padó, S., Pennacchiotti, M., Romano, L., & Szpakowicz, S. (2010). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. ACL 2010, 33.

Holub, A., Perona, P., & Burl, M.C. (2008). Entropy-Based active learning for object recognition. Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 1–8. https://doi.org/10.1109/CVPRW.2008.4563068.

Kartchner, D., Ren, W., Nakajima An, D., Zhang, C., & Mitchell, C.S. (2020). Regal: Rule-Generative active learning for model-in-the-loop weak supervision. Advances in neural information processing systems.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems, 35*, 22199–22213.

Li, Q., Jiang, M., Zhang, X., Qu, M., Hanratty, T.P., Gao, J., & Han, J. (2018). TruePIE: Discovering reliable patterns in pattern-based information extraction. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 1675–1684. 10.1145/3219819.3220017.

Li, J., Ding, H., Shang, J., McAuley, J., & Feng, Z. (2021). Weakly supervised named entity tagging with learnable logical rules. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 4568–4581. 10.18653/v1/2021.acl-long.352.

Liang, J., Feng, S., Xie, C., Xiao, Y., Chen, J., & Hwang, S.-W. (2021). Bootstrapping information extraction via conceptualization. Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), 49–60. 10.1109/ICDE51399.2021.00012.

Liang, W., Tadesse, G. A., Ho, D., Fei-Fei, L., Zaharia, M., Zhang, C., & Zou, J. (2022). Advances, challenges and opportunities in creating data for trustworthy AI. *Nature Machine Intelligence, 4*(8), 669–677.

Liu, Z., Ding, H., Zhong, H., Li, W., Dai, J., & He, C. (2021). Influence selection for active learning. Proceedings of the IEEE/CVF International Conference on Computer Vision, 9274–9283. Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., & Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases, 11(3), 269-282. 10.14778/3157794.3157797.

Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., & Ré, C. (2016). Data programming: Creating large training sets, quickly. *Advances in neural information processing systems, 29*, 3567–3575.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., Chen, X., & Wang, X. (2021). A survey of deep active learning. *ACM Computing Surveys, 54*(9). https://doi.org/10.1145/3472291

Rossi, A., Barbosa, D., Firmani, D., Matinata, A., & Merialdo, P. (2021). Knowledge graph embedding for link prediction: A comparative analysis. *ACM Trans. Knowl. Discov. Data*, (2), 15. https://doi.org/10.1145/3424672

Safranchik, E., Luo, S., & Bach, S. (2020). Weakly supervised sequence tagging from noisy rules. Proceedings of the AAAI Conference on Artificial Intelligence, 34 (04), 5570–5578. 10.1609/aaai.v34i04.6009.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L.M. (2021). "Everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 1–15. https://doi.org/10.1145/3411764.3445518.

Varma, P., & Ré, C. (2018). Snuba: Automating weak supervision to label training data. Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases, 12 (3), 223–236. https://doi.org/10.14778/3291264.3291268.

Wang, X., He, X., Cao, Y., Liu, M., & Chua, T.-S. (2019). KGAT: Knowledge graph attention network for recommendation. Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 950–958. 10.1145/3292500.3330989.

Wang, M., Wang, H., Qi, G., & Zheng, Q. (2020). Richpedia: A large-scale, comprehensive multi-modal knowledge graph. *Big Data Research, 22*, Article 100159.

Whang, S. E., Roh, Y., Song, H., & Lee, J. G. (2023). Data collection and quality challenges in deep learning: A data-centric AI perspective. *The VLDB Journal, 32*(4), 791–813.

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., & Sugiyama, M. (2019). Are anchor points really indispensable in label-noise learning?. *Advances in neural information processing systems* (p. 32). Proceedings of the First 12 Conferences.

Yan, Y., Li, R., Wang, S., Zhang, F., Wu, W., & Xu, W. (2021). ConSERT: A contrastive framework for self-supervised sentence representation transfer. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 5065–5075. 10.18653/v1/2021.acl-long.393.

Yang, J., Fan, J., Wei, Z., Li, G., Liu, T., & Du, X. (2018). Cost-effective data annotation using game-based crowdsourcing. *Proceedings of the VLDB Endowment, 12*(1), 57–70. https://doi.org/10.14778/3275536.3275541

Ye, H., & Luo, Z. (2020). Deep-ranking-based cost-sensitive multi-label learning for distant supervision relation extraction. *Information Processing & Management, 57* (6), Article 102096. https://doi.org/10.1016/j.ipm.2019.102096

Zhang, X.F., & de Marneffe, M.C. (2021). Identifying inherent disagreement in natural language inference. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 4908–4915). https://doi.org/10.18653/v1/2021.naacl-main.390.

Zhang, J., Yu, Y., Li, Y., Wang, Y., Yang, Y., Yang, M., & Ratner, A. (2021). WRENCH: A comprehensive benchmark for weak supervision. Thirty- fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Zhang, R., Yu, Y., Shetty, P., Song, L., & Zhang, C. (2022). Prompt-based rule discovery and boosting for interactive weakly-supervised learning. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 745–758. 10.18653/v1/2022.acl-long.55.

Zhang, H., Zhang, C., & Wang, Y. (2024). Revealing the technology development of natural language processing: A Scientific entity-centric perspective. *Information Processing & Management, 61*(1), Article 103574.

Zhao, J., Song, R., Yue, C., Wang, Z., & Xu, H. (2023). Weak-PMLC: A large-scale framework for multi-label policy classification based on extremely weak supervision. *Information Processing & Management, 60*(5), Article 103442. https://doi.org/10.1016/j.ipm.2023.103442

Zhong, Z., Li, C.-T., & Pang, J. (2023). Hierarchical message-passing graph neural networks. *Data Mining and Knowledge Discovery, 37*(1), 381–408.

Zhou, W., Lin, H., Lin, B.Y., Wang, Z., Du, J., Neves, L., & Ren, X. (2020). Nero: A neural rule grounding framework for label-efficient relation extraction. Proceedings of the Web Conference 2020, 2166–2176. 10.1145/3366423.3380282.

Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National Science Review, 5*(1), 44–53. https://doi.org/10.1093/nsr/nwx106