PNRE: Proactive Learning for Neural Relation Extraction with Multiple Annotators

Rui Qiu[1,2], Wen Ji[1(✉)], Yundan Liang[3], Haini Qu[4], and Jingce Xu[5]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{qiurui,jiwen}@ict.ac.cn
[2] University of Chinese Academy of Science, Beijing, China
[3] State Grid Corporation of China, Beijing, China
yundan-liang@sgcc.com.cn
[4] State Grid Shanghai Electric Power Company, Shanghai, China
quhn@sh.sgcc.com.cn
[5] State Grid Energy Research Institute, Beijing, China
xujingce@sgeri.sgcc.com.cn

Abstract. Relation extraction is one of the essential tasks of information extraction, and it is also a fundamental part of knowledge graph construction. Many works have been proposed for supervised relation extraction, which commonly requires a massive amount of human-annotated data with both time and cost. To reduce the labeling time and cost, active learning has been proposed with the assumption of a single perfect annotator that always furnishes the correct label. However, more generally, the annotator will provide incorrect labels according to their labeling capabilities, and different labeling capabilities correspond to distinct costs. To unleash the power of annotators with diverse expertise level and unlabeled data for better model performance with the lowest cost, we develop PNRE, a novel proactive learning based framework for neural relation extraction that actively select the most suitable sample-annotator pairs to construct high-quality relation extraction corpus. Specifically, PNRE utilizes (1) Expert Performance Estimation module to precompute each annotator's performance considering class prediction probability; (2) Sample Selection module to select the most informative and representative sample based on a hybrid query strategy; (3) Sample Allocation module to allocate appropriate sample to each annotator under the condition of annotation utility maximization. The framework is evaluated on three corpora and is shown to achieve promising results with a significant reduction in labeling costs.

Keywords: Proactive Learning · Relation Extraction · Cost-Effectiveness

1 Introduction

Relation Extraction (RE) is a fundamental yet challenging subtask of Information Extraction (IE), which involves extracting structured information, that can be interpreted easily by a machine or a program, from plain unstructured text [3]. It plays an important role

K. Subramanian et al. (Eds.): CTMCD 2022, ACSR 99, pp. 352–364, 2023.
https://doi.org/10.2991/978-94-6463-046-6_42

in many natural language processing (NLP) applications like knowledge graph completion, question answering (QA) and search engine [16] Given a corpus, RE is to recognize the semantic relationships between all entity mention pairs. However, many state-of-art research on relation extraction focuses on improving performance on benchmark corpus, which are high-quality annotated by human. As is known to all, manually annotating a corpus for RE task is both time-consuming and costly, especially domain-relevant corpus. To address this issue, approaches such as distant supervision and active learning have been proposed.

Distant supervision is a method for automatically constructing datasets for relation extraction tasks. However, the generated datasets have wrong labels and long-tail problems due to the strong assumptions. The assumptions proposed in [8] results in that although a pair of entities appear in the same sentence, the corresponding relationship does not appear in the knowledge graph. If wrong instances have a large proportion, the model is more likely to fit noisy data. In addition, distant supervision mainly uses the knowledge graph of the general domain, which leads to the fact that the number of samples of general relations is much larger than that of non-general relations, resulting in extremely unbalanced training samples. Therefore, manual annotation by experts is critical, especially for domain-specific long-tail relationships.

Active learning, a semi-supervised machine learning algorithm, aims to achieve better accuracy with fewer labeled data [1]. At each iterative annotation, active learning select the most informative and representative samples to annotators, which can produce a high-quality annotated corpus in less time and at lower cost than traditional labeling methods [7]. Several studies have shown that active learning can select the most beneficial instances to be labeled for further improving the model performance in a variety applications, including information extraction, network/graph analysis [9], etc. Nevertheless, active learning relies on two strong assumptions, resulting in real-world applications: (1) it assumes the existence of a single perfect annotator, however more generally the annotator from multiple sources may have different reliability varies by experience level; (2) it assumes that the labeling of different samples has a uniform cost, ignoring the difficulty of the samples and the distinction in annotation ability.

To relax the above-mentioned, proactive learning has been proposed to jointly select the optimal annotator and instance by casting the problem as a utility optimization problem subject to a budget constraint [14]. Proactive learning assumes that (1) not all annotators are perfect, there is at least one "perfect" annotator and one "fallible" annotator; (2) the higher the reliability of the annotator, the higher the annotation cost. Same as the annotation process of the traditional active learning, at each annotation iteration, proactive learning query label from annotators for the selected unlabeled dataset and append new labeled to the labeled dataset. However, the difference with active learning is that, proactive learning select the optimal sample-annotator pair to reduce annotation cost.

To unleash the power of annotators with diverse expertise level and unlabeled data for better model performance with the lowest cost, we develop PNRE, a novel proactive learning based framework for neural relation extraction that actively select the most suitable sample-annotator pairs to construct high-quality relation extraction corpus. To the best of our knowledge, Our work is the first study to apply proactive learning to neural relation extraction considering multiple noisy annotators.

For the purpose of evaluation, we train BERT-based relation extraction model [22] on training data with different ratios to simulate different performance annotators. To verify the effectiveness of the method, we conduct experiments on three common corpora, two general domain, and one in AI domain. The results demonstrate that using proactive learning can improve the quality of data annotation while reducing the cost of annotation, thereby improving model performance.

To summarize, we make the following main contributions:

- We propose PNRE framework on relation extraction task to construct high quality labeled corpus with the lowest cost considering multiple noisy annotators.
- We present a hybrid query strategy to select a batch of samples with more valuable information and use the optimization objective based on the maximization of annotation utility to allocate the selected samples to the right annotators.
- By conducting extensive experiments on three corpus, it is proved that our method is applicable to multiple fields and has certain practical value.

The rest of this paper is organized as follows. In Sect. 2, we review the related work on neural relation extraction and proactive learning. In Sect. 3, the propose method is introduced. Section 4 presents the experiments, followed by the conclusion and future work in Sect. 5.

2 Related Work

Active Learning for Relation Extraction: Active learning approaches proposed in recent years are mainly query-based methods. Active learning aims to spend less annotation costs while maintaining an acceptable quality of annotated data or improve model performance, that is, selecting the instance with the most informative and representative by designing query strategies (sampling rules) [10], including (1) selecting samples with the most uncertainty [4 5, 20]. (2) selecting an optimal subset based on diversity [12], and their combinations [17]. Zhang and Huang [6] present an unified active learning framework for biomedical relation extraction, addressing some practical issues during active learning process. Mallart and Nouy [2] proposed a lightweight active-learning based relation extraction pipeline for newspaper scenario, where dedicated to local information, various relation and highly specific type. As a generalization of active learning, proactive learning also applied in several NLP task, such as NER (name entity recognition) [7].

Cost-Effectiveness Active Learning: Cost-effectiveness active learning is a type of active learning method that considers the annotation cost, e.g., budget, time or effort required to complete the annotation process [19]. Since proactive learning also models the reliability or expertise of each annotator in addition to the annotation cost, it can be considered as another case of cost-effectiveness active learning [21]. There are a few studies trying to select annotators with matching expertise on the specific instance to be labeled [11, 13]. A common shortcoming of these algorithms is that they do not consider the difference on the costs of multiple annotators, and thus may get an accurate yet expensive solution. Proactive learning considers a simple case where two annotators exist: one perfect which always returns the ground truth, and one fallible which make mistakes with a probability. Although they assume different costs for the two annotators, the oversimple setting limits its application [14]. Huang and Chen [19] expands two experts to multiple experts, under the same utility function. Chakraborty [18] pose the optimal sample and annotator selection as a constrained optimization problem and derive a linear programming relaxation to select a batch of (sample-annotator) pairs. The datasets evaluated by the above algorithms are all one sample corresponding to one label instance, however, for relation extraction tasks, a sentence may contain multiple relation instances.

3 Methodology

In this section, we present a proactive learning framework for relation extraction when multiple noisy annotators are present. Before introducing the framework, we need to formulate two problems, one is relation extraction, and the other is proactive learning. Next, we describe the methods corresponding to the three modules–Expert Performance Estimation, Sample Selection, Sample Allocation in detail.

3.1 Problem Formulation

Let $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ represent a sentence where n is the length of sentence and s_i represent the i-th token. Let $\mathcal{E} = \{e_1, e_2, ..., e_m\}$ be the set of entity mentions with entity types in the sentence, and $\mathcal{R} = \{r_{11}, r_{12}, ..., r_{ij}\}$ be the relation set between entity e_i and e_j. Given an input sentence $\mathcal{S}$ and the entities set contained in the sentence, the RE model is to predict a relation type y_r for each entity pair (e_i, e_j) or no relation between them. Therefore, it can be regarded as a classification problem. In this paper, we apply the RE model proposed in [22].

Problem Definition 1: (Relation Extraction) Given a sentence $\mathcal{S} \in \mathcal{U}$, the relation extraction model problem is to find the relation set $\widehat{\mathcal{R}}$ from $\mathcal{S}$.

Let $\mathcal{L} = \{(x, y)\}_L$ with n_l examples be the corpus of labeled sentences, where y may contains multiple relation labels, and $U = \{(x, y)\}_U$ with n_u unlabeled instances, typically $n_l \ll n_u$. There is a set of candidate annotators $A = \{a_1, a_2, \ldots, a_m\}$ with all m annotators offering different annotation capabilities, corresponding to different annotation costs.

Problem Definition 2: (Proactive Learning for Relation Extraction) Given an unlabeled corpus $\mathcal{U}$, it iteratively selects a sample-annotator pair for labeling in order to maximize the performance of relation extraction until the budget is reached or the performance requirements of the model are met.

3.2 Expert Performance Estimation

Inspired by the equation proposed in [15], different experts have certain biases for different class of annotation capabilities. Therefore, we can estimated the annotation ability of each expert by two probabilities: the class probability, $p(label|k, c)$ and the sentence probability $p(ans|x, k)$.

3.3 Sample Selection

The difference from the previous related work is that each sample corresponds to only one label instance, such as UCI repository. For example, each instance in *spambase* labeled spam(1) or not(0), while a sentence may contain multiple relations in relation extraction task. Therefore, it is necessary to propose a better query strategy to select more informative samples.

We need a metric to quantify the utility score to determine the most informative set of samples to query. Our study incorporates informativeness and redundancy criteria to calculate utility scores. The sample selection module in which these two conditions drive samples ensures that the selected samples are individually informative with minimal redundancy (duplication) between them.

3.4 Sample Allocation

After calculating each expert annotation ability and the ranking score of the unlabeled sample, we can assign the samples to appropriate experts, getting the optimal sample-annotator pairs. The pseudo-code of the proposed algorithm, termed PNRE is outlined in Algorithm 1.

At each iteration of PNRE, the algorithm selects the most useful sample-annotator pair (x^*, k^*), and queries the label of x^* from k^*. Then x^* is removed from the unlabeled set U, and is appended into L with its queried label y^* from annotator k^*. After that, the RE model is retrained on the new labeled dataset L. Finally, we evaluate the retrained RE model on the test set. This process is repeated until it meets a given condition, e.g., the given cost budget is reached or the expected model performance is reached.

Algorithm 1 The PNRE algorithm

Input: Labeled set L , Unlabeled set UL , test set T , budget B , annotator k and label cost C_k , allocate unlabeled set UL_k to annotator k , current label cost C

Output: Total labeled set

First, the annotation ability of each annotator is evaluated on theseed labeled set

while $C < B$ **do**

Train the RE model on *L*, and evaluate the model on the test set

Sort unlabeled samples

Select Top-N samples

for all sentence x **do**

Initialize utility score u_x and candidate annotator *k*

for all annotator *k* **do**

Calculate annotation utility

if $u_x < U(x,k)$ **then**

$k' = k$; $u_x = U(x,k)$

end if

end for

Allocate sample x to UL_k

end for

for all annotator *k* **do**

$L_k \leftarrow$ expert k label all sample in UL_k

$L = L \cup L_k$

$UL = UL - UL_k$

$C = C + C_k * | L_k |$

end for

end while

4 Experiments

4.1 Dataset

We evaluate our method on three popular relation extraction datasets: Conll04, ACE05, and SciERC. The Conll04 and ACE05 are collected from general domains, such as newswire and online forums, e.g. Work for, OrgBased in, Part-Whole, PER-SOC. The SciERC dataset is collected from 500 AI paper abstracts and defines scientific terms and relations, especially for scientific knowledge graph construction.

For ACE05 we only considered the top-level classes, ignoring that top-level relation classes. All three datasets are divided into the train, test, and validation datasets. In the experimental set, we divided the train set into two parts, 1% as the initial seed labeled data and the rest as the data to be labeled. Table 1 shows statistic information of the three corpora. The first column represents the dataset name, the second column represents the relation types contained in the dataset, the third column represents the number of sentences for each relation in the initial seed labeled data, and the fourth column represents the number of sentences for each relation in the unlabeled data. Sentences that do not contain relations are filtered. We save the optimal model on the validation set to evaluate the experimental performance of the current iteration.

4.2 Expert Simulation

We simulated annotators with different annotation capabilities by using the RE model proposed in [8] on different size of train set. The RE model processes each pair of spans independently and inserts typed markers at the input layer to emphasize the subject and object and their types. The final representation of span-pair is the concatenation of the output representations of span start typed markers by pre-trained encoder, and feed into a feed forward network to predict the probability distribution of relation type.

In our settings, we trained the RE model on 100%, 80%, 60%, and 40% training set to represent experts with diminishing annotation ability and evaluate the performance on the test set. The results of F1 score for each relation type on different percentage of train set show on Tables 2, 3 and 4.

We use the macro F1 score evaluate on test set to represent the overall annotation ability for each expert. As illustrated in Table 5, the more training data, the higher the expert's performance. For expert cost, similar to previous research [19], each expert was assigned an integer between 1 and 4 in increasing order of macro F1 score.

4.3 Comparison with Baseline

We compare the proposed algorithm PNRE against the following baseline methods:

- **RR**: randomly select a batch of sample and query each label from one randomly selected annotators.
- **AR**: select a batch of sample using Sample Selection module and query each label from one randomly selected annotators.

Table 1. Statistic information of the three corpora.

Corpus	Relation	labeled	Unlabeled
Conll04	Work-For	5	251
	Kill	1	178
	OrgBased-In	2	269
	Live-In	2	328
	Located-in	0	247
ACE05	PHYS	9	1410
	PART-WHOLE	7	971
	PER-SOC	11	812
	GEN-AFF	3	703
	ORG-AFF	26	1944
	ART	6	632
SciERC	Used-of	12	1675
	Feature-of	2	171
	Hyponym-of	5	293
	Evaluate-for	1	311
	Part-of	5	174
	Compare	1	165
	Conjunction	5	395

Table 2. F1 score for each relation on Conll04.

Relation	100%	80%	60%	40%
Work-For	0.848	0.795	0.775	0.771
Kill	0.886	0.877	0.89	0.888
OrgBased-In	0.752	0.74	0.691	0.637
Live-In	0.775	0.75	0.755	0.687
Located-In	0.815	0.765	0.769	0.697

- **RL**: randomly select a batch of sample and query each label by the lowest cost annotators.
- **AL**: select a batch of sample using Sample Selection module and query each label from one randomly selected annotators.
- **RG**: randomly select a batch of sample and query each label using Sample Allocation module.

Table 3. F1 score for each relation on ACE05.

Relation	100%	80%	60%	40%
PHYS	0.83	0.781	0.764	0.711
PART-WHOLE	0.832	0.818	0.790	0.750
PER-SOC	0.861	0.844	0.817	0.779
GEN-AFF	0.70	0.678	0.621	0.604
ORG-AFF	0.905	0.903	0.882	0.863
ART	0.858	0.84	0.815	0.718

Table 4. F1 score for each relation on SciERC.

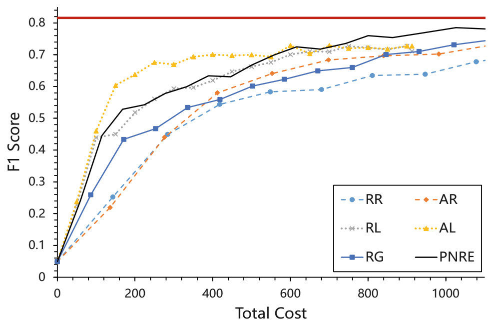
Relation	100%	80%	60%	40%
Used-for	0.777	0.776	0.747	0.720
Feature-of	0.5299	0.472	0.4314	0.326
Hyponym-of	0.8571	0.852	0.8209	0.808
Evaluate-for	0.6984	0.705	0.6809	0.604
Part-of	0.4848	0.455	0.419	0.278
Compare	0.6479	0.630	0.5758	0.64
Conjunction	0.83	0.808	0.83	0.801

Table 5. The overall label ability of each expert on the three corpora.

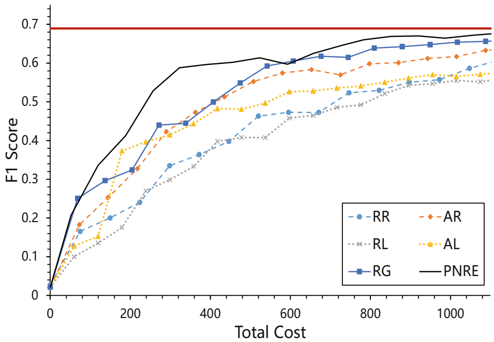
Corpus	100%	80%	60%	40%
Conll04	0.815	0.785	0.777	0.736
ACE05	0.831	0.811	0.781	0.738
SciERC	0.689	0.671	0.643	0.598

Figure 1 plots the F1 score curves with the increasing cost for all compared baselines. In each graph, the x-axis denotes the current cost of each iteration and the y-axis denotes the F1 score on the test set.

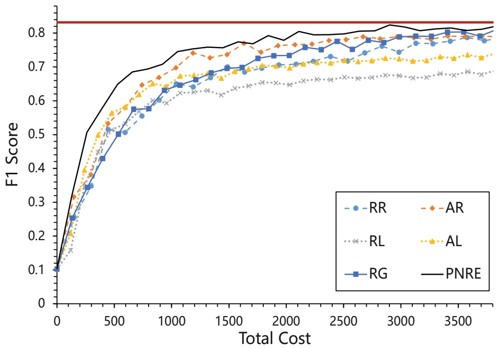
In general, Using Sample Selection module can usually converge to a higher F1 score faster. Take the corpus Conll04 and ACE05 as examples to compare the AL and RL. In the early iteration, the Sample Selection module can often select more informative samples (plotted with yellow dot line), resulting in a larger model performance improvement, compared with randomly selected samples (plotted with gray line). However, for corpus SciERC, adding the Sample Selection module does not bring much performance improvement. The main reason is that the corpus itself is complex (lower macro F1



Conll04



SciERC



ACE05

Fig. 1. The best PNRE results on the three corpora in comparison to the baselines.

score for 100% train set), and the AL and RL select the experts with the lowest cost for labeling, the labels contain much noise, and the model is challenging to learn from the noise samples effectively. The above conclusions verify the effectiveness of the Sample Selection module.

Comparing PNRE (plotted with black solid line) with other baseline methods, we can see that our proposed framework outperforms the baseline by different degrees.

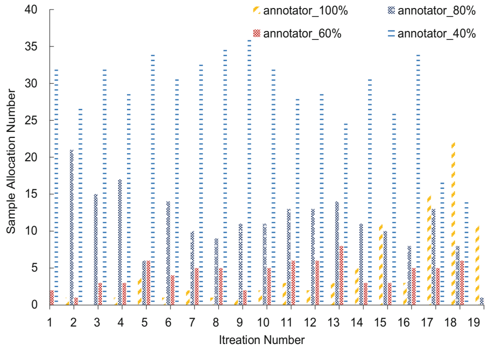


Fig. 2. Number of samples allocated to each annotator

Table 6. Percentage of saving cost compared with baseline.

Method	Corpus		
	Conll04 Threshold:0.75	SciERC Threshold:0.71	ACE05 Threshold:0.79
RR	2066 (**61.2%**)	1719 (**26.8%**)	4028 (**46.2%**)
AR	2025 (**60%**)	1377 (**8.6%**)	3410 (**36.5%**)
RL	-	1602 (**21.4%**)	4059 (**46.6%**)
AL	-	1552 (**18.9%**)	3939 (**45%**)
RG	1295 (**38.2%**)	1567 (**19.7%**)	3302 (**34.4%**)
PNRE	800	1258	2164

On corpus Conll04 and SciERC, although the AL strategy is better than PNRE in the early iteration, with the increase of labeled samples, PNRE will more reasonably assign samples to the most suitable experts for labeling. Therefore, we can see that after the annotation is completed, the final F1 score of AL and RL is lower than that of PNRE. Therefore, it can be concluded that PNRE can label more informative samples with less cost, thereby bringing labeling efficiency and improving model performance.

In order to further verify the effectiveness of PNRE, we set a F1 score threshold for each corpus, and calculate the proportion of the cost saved by PNRE when the threshold is reached for the first time compared to the baseline, and the results are shown in Table 6. Finally, we take Conll04 as an example to show the changing trend of the number of samples assigned to each expert by the PNRE strategy in the iterative process, as shown in Fig. 2. At the beginning of the iteration, due to the lack of labeled data, the RE model's performance is low. The samples filtered by the sample selection module are mostly easy samples, so most of the samples in the early stage are allocated the lowest cost annotator. As the performance of the model increases, and the difficulty of the selected samples also increases. Therefore, in the later iteration stage, experts with higher cost and better labeling ability are required to label, which also verifies the effectiveness of PNRE.

5 Conclusions

In this paper, we constitute the first attempt to apply proactive learning for neural relation extraction (PNRE) under a novel setting, where multiple noisy annotation costs. We simulate different expertise level annotators by applying BERT-based RE model, and calculate each expert performance by Expert Performance module. To save annotation costs and to ensure acceptable quality of the annotated data, we design Sample Selection module to choose the most informative and representative instance and Sample Allocation module to select appropriate sample-annotator pair under the object of annotation utility maximization. Experimental results on three corpora demonstrate that the proposed framework PNRE is able to achieve higher accuracy with lower query cost for relation extraction task.

One potential limitation of our approach is that accurately estimating an expert's performance requires a gold standard corpus, however in some domains, such corpus is more difficult to obtain. Another potential limitation is that, due to the use of pre-trained language model, each iteration will take a long time, resulting in training slowly. Therefore, we will explore how to estimate the annotator's performance with a small set of gold corpus and further verify that the PNRE framework utilizing lightweight RE models is also practical. As a further extension to our work, we will explore the deployment of our method on Crowdsourcing platforms.

Acknowledgements. This work is supported by the Key Technologies of Electric Power Data Sharing and Opening for Digital Services (SGSH0000HLJS2000541).

References

1. Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S. Yu. Active learning: A survey. In *Data Classification: Algorithms and Applications*, pages 571–606. CRC Press, 2014.
2. Cyrielle Mallart, Michel Le Nouy, Guillaume Gravier, and Pascale Se´billot. Active learning for interactive relation extraction in a french newspaper's articles. In *RANLP*, pages 886–894. INCOMA Ltd., 2021.
3. Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In *CoNLL*, pages 1–8. ACL, 2004.
4. Donggeun Yoo and In So Kweon. Learning loss for active learning. In *CVPR*, pages 93–102. Computer Vision Foundation/IEEE, 2019.
5. Haw-Shiuan Chang, Shankar Vembu, Sunil Mohan, Rheeya Uppaal, and Andrew McCallum. Using error decay prediction to overcome practical issues of deep active learning for named entity recognition. *Mach. Learn.*, 109(9–10):1749–1778, 2020.
6. Hongtao Zhang, Minlie Huang, and Xiaoyan Zhu. A unified active learning framework for biomedical relation extraction. *J. Comput. Sci. Technol.*, 27(6):1302–1313, 2012.
7. Maolin Li, Nhung T. H. Nguyen, and Sophia Ananiadou. Proactive learning for named entity recognition. In *BioNLP*, pages 117–125. Association for Computational Linguistics, 2017.
8. Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *ACL/IJCNLP*, pages 1003–1011. The Association for Computer Linguistics, 2009.

9. Mustafa Bilgic, Lilyana Mihalkova, and Lise Getoor. Active learning for networked data. In *ICML*, pages 79–86. Omnipress, 2010.
10. Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP/IJCNLP (1)*, pages 3980–3990. Association for Computational Linguistics, 2019.
11. Ofer Dekel, Claudio Gentile, and Karthik Sridharan. Selective sampling and active learning from single and multiple teachers. *J. Mach. Learn. Res.*, 13:2655–2697, 2012.
12. Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR (Poster)*. OpenReview.net, 2018.
13. Panagiotis G. Ipeirotis, Foster J. Provost, Victor S. Sheng, and Jing Wang. Repeated labeling using multiple noisy labelers. *Data Min. Knowl. Discov.*, 28(2):402–441, 2014.
14. Pinar Donmez and Jaime G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *CIKM*, pages 619– 628. ACM, 2008.
15. Seungwhan Moon and Jaime G. Carbonell. Proactive learning with multiple class-sensitive labelers. In *DSAA*, pages 32–38. IEEE, 2014.
16. Shantanu Kumar. A survey of deep learning methods for relation extraction. CoRR, abs/1705.03645, 2017.
17. Shayok Chakraborty, Vineeth Nallure Balasubramanian, Qian Sun, Sethuraman Panchanathan, and Jieping Ye. Active batch selection via convex relaxations with guaranteed solution bounds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(10):1945–1958, 2015.
18. Shayok Chakraborty. Asking the right questions to the right users: Active learning with imperfect oracles. In *AAAI*, pages 3365–3372. AAAI Press, 2020.
19. Sheng-Jun Huang, Jia-Lve Chen, Xin Mu, and Zhi-Hua Zhou. Cost-effective active learning from diverse labelers. In *IJCAI*, pages 1879–1885. ijcai.org, 2017.
20. William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. The power of ensembles for active learning in image classification. In *CVPR*, pages 9368–9377. Computer Vision Foundation/IEEE Computer Society, 2018.
21. Yan Zhuang, Guoliang Li, Wanguo Xue, and Fu Zhu. An active learning based hybrid neural network for joint information extraction. In *WISE (2)*, volume 12343 of *Lecture Notes in Computer Science*, pages 84–100. Springer, 2020.
22. Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In *NAACL-HLT*, pages 50–61. Association for Computational Linguistics, 2021.