Finding Influential Instances for Distantly Supervised Relation Extraction

Anonymous EMNLP submission

Abstract

Distant supervision has been demonstrated to be highly beneficial to enhance relation extraction models, but it often suffers from high label noise. In this work, we propose a novel model-agnostic instance subsampling method for distantly supervised relation extraction, namely REIF. It encompasses two key steps: first calculating instance-level influences that measure how much each training instance contributes to the validation loss change of our model, then deriving sampling probabilities via the proposed sigmoid sampling function to perform batch-in-bag sampling. We design a fast influence subsampling scheme that reduces the computational complexity from $\mathcal{O}(mn)$ to $\mathcal{O}(1)$, and analyze its robustness when the sigmoid sampling function is employed. Empirical experiments demonstrate our method's superiority over the baselines, and its ability to support interpretable instance selection.

1 Introduction

Relation extraction (RE) is the fundamental technique to mine the relations between entity pairs from unstructured text data. A sentence "*Bill Gates, the founder of Microsoft, has indicated ...*" contains the relation *founder* between the entities *Microsoft* and *Bill Gates*. By identifying this relation, a triplet (*Microsoft, founder, Bill Gates*) can be built, and incorporated in the existing knowledge bases (KB).

One of the main challenges in RE, as similar in many other machine learning applications, is lacking labeled data. To cope with it, distant supervision (DS) was proposed by Mintz et al. (2009). In DS, a strong assumption is made that if two entities are related in existing KBs, then all sentences contain both of them are considered to express this relation. However, this heuristic labeling process inevitably suffers from wrong labels (Takamatsu et al., 2012), and undermines RE model's performance in practice. For example, the sentence "*Bill Gates redefined the software industry, ... said Rob Glaser, a former Microsoft executive*" does not mention the relation *founder* but is still treated as a positive training sample in DS.

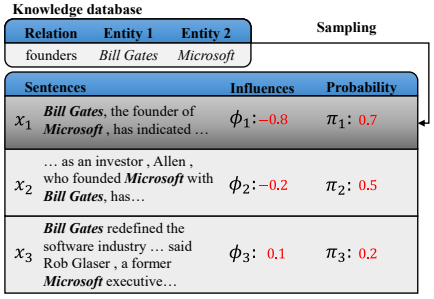


Figure 1: Finding influential instances within a bag via subsampling based on the calculated probability π. Note that here *negative* ϕ means a beneficial sample.

As a matter of fact, dealing with noisy instances in DS has been a focus in the RE research. There are three main genres in the literature: (1) incorporating an attention module in models (Lin et al., 2016) to allocate confidence level among instances in the same bag; (2) using reinforcement learning (Qin et al., 2018b) for instance selection; and (3) leveraging adversarial training (Wu et al., 2017) to enhance the RE model's robustness against noise.

Aside from the above denoising methods in DS, the influence subsampling (IFS) (Wang et al., 2020) was recently proposed for data denoising in supervised learning. Motivated by the IFS's promising performance in dropping out adverse samples for denoising, in this work, we try to fulfill the IFS's potential in distantly supervised RE, by developing the **R**elation **E**xtraction by **In**F**l**uence subsampling (REIF) framework. As shown in Fig. 1, each instance is assigned a quality measure ϕ, from which its sampling probability is obtained via the sampling function π. Accordingly, the better an instance's quality is, the more likely it being picked during training. We will explain the operational meaning of ϕ in Section 3.2.

However, several challenges are required to be resolved for using REIF in DS. First, unlike in com-

mon supervised learning, here we do not have exact labels of instances, but the labels of bags, hence the original IFS requires to be adapted for instance sampling with these highly noisy labels. Second, the IFS proposed by Wang et al. (2020) only supports specific binary classification models, e.g., logistic regression and support vector machines, thus inapplicable and not computationally efficient in RE, in which multi-class classification and deep learning models are often exploited. Last, we observe that the original IFS that samples from full dataset often causes severe class imbalance. However, in RE where there are many relations to be discriminated, an imbalanced training set is harmful for RE model's prediction capacity of those minor relations with few training samples.

Our main contributions are four-folds:

1. We develop a novel influence subsampling framework in distantly supervised relation extraction, namely REIF, for denoising RE by sampling favorable training instances.

2. To address the limitations of IFS in RE, we propose to employ a sigmoid sampling function and batch-in-bag sampling in our REIF.

3. An efficient implementation of REIF enables subsampling in $\mathcal{O}(1)$ complexity, instead of the $\mathcal{O}(mn)$ in the original IFS methods.

4. Empirical experiments show REIF's superiority over other baselines, and we identify its capability to support interpretable instance selection for RE by a case study.[1]

2 Related Work

2.1 Distant Supervision

Previous works tried to address the noisy label difficulty in DS by multi-instance learning (MIL) (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012). MIL regards the labels of training data in *bag* level instead of sentence level. That is, each bag contains at least one instance with the labeled relation while the exact label of each instance is unknown. As MIL being proved effective in relation extraction, it was firstly introduced to neural relation extraction by Zeng et al. (2015), where the piece-wise convolutional neural network (PCNN) was developed, and only one instance with the largest predicted probability was selected in each bag.

Later, attention (Lin et al., 2016; Zhou et al., 2018; Jia et al., 2019; Yuan et al., 2019), reinforcement learning (Feng et al., 2018; Yang et al., 2018; Qin et al., 2018b), and adversarial training (Wu et al., 2017; Qin et al., 2018a; Han et al., 2018) have been proposed for further improvement. However, above works usually require intense trials in fine-tuning of the hyper-parameters in practice, or are not interpretable to human-beings. In this work, we propose a model-agnostic and interpretable instance selection method via IFS, namely REIF, which is easy-to-use and effective to various deep learning models.

2.2 Influence Subsampling

Training data selection is a long lasting topic in machine learning applications. Recent works focused on how to measure data's quality quantitatively by influence function (Koh and Liang, 2017), thus conducting data selection (Wang et al., 2018; Sharchilev et al., 2018). However, the proposed two-round training suffers from prohibitive computation complexity, thus not applicable to large-scale data. Besides, Wang et al. (2020) found that a deterministic data selection scheme is not robust to distribution shift, hence they extended it to a probabilistic sampling form. Different from Wang et al. (2020), we here concentrate on data sampling under weak supervision situation, and further extend the probabilistic IFS techniques to deep learning models.

3 Methodology

In this section, we present how to incorporate influence subsampling into distantly supervised relation extraction. We first present the major steps of our REIF framework, especially our proposed *batch-in-bag* sampling strategy rather than the post-hoc sampling as previous IFS works. Then, we specify the theoretical foundation on measuring data's quality via the influence function, and how to obtain their probabilities via the *sigmoid sampling function*. At last, we provide theoretical analysis of our choice of sigmoid sampling in the framework.

3.1 Relation Extraction by Influence Subsampling (REIF)

The flowchart of our framework is shown in Fig. 2. It includes three main parts: 1) word representation,

[1] Code is available in the supplementary materials.

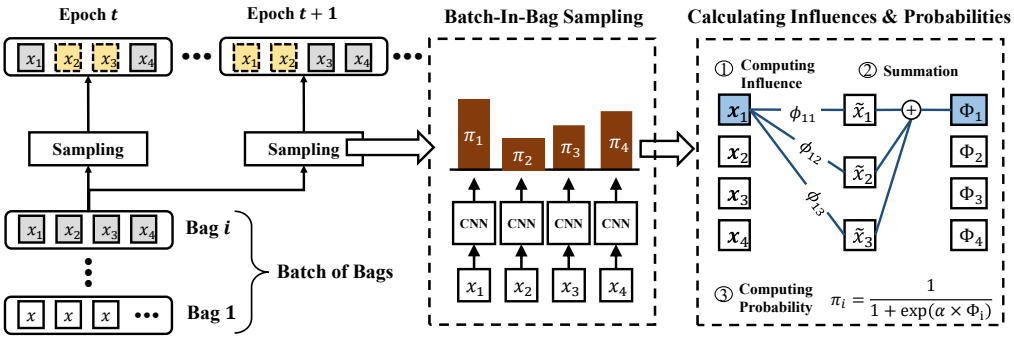


Figure 2: The flowchart of the instance-level subsampling method, where x is training sentence; $\tilde{x}$ is the validation sample; ϕ is the computed influence; and a dotted box means the instance is dropped after subsampling.

2) convolution layers, and 3) instance selection.

Word Representation. Inputs of the encoder are raw sentences, which are usually represented by indices of words, e.g., a sentence x_* with l words $x_* = \{x_{*,1}, \ldots, x_{*,l}\}$. Similar to Zeng et al. (2015), we transform them into dense real-valued representation vectors as $\boldsymbol{w}_* = \{\boldsymbol{w}_{*,1}, \ldots, \boldsymbol{w}_{*,l}\}$, by concatenating the word embedding from $\boldsymbol{V} \in \mathbb{R}^{d^a \times |V|}$ (where $|V|$ denotes the size of the vocabulary and d^a is the dimension of word embedding) and position embedding with dimension d^p together. As there are two position embeddings with each measuring the relative distance to one of the two entities, each word vector in $\boldsymbol{w}$ has dimension $d^a + 2 \times d^p$.

Convolution Layers. Convolution operations are further conducted on the obtained word representations, which can be briefly described as

$$\boldsymbol{x}_* = \mathrm{CNN}(\boldsymbol{w}_*). \tag{1}$$

The CNN model takes the representation vectors $\boldsymbol{w}_*$ as inputs, and outputs the processed feature vectors $\boldsymbol{x}_* \in \mathbb{R}^{d \times l}$. Details of the above CNN structure can be referred to (Zeng et al., 2015). The probability for relation prediction, taking $\boldsymbol{x}_*$ as input, is given by

$$P(y = k|\boldsymbol{x}_*) = \frac{\exp(\boldsymbol{\beta}^{(k)\top} \boldsymbol{x}_*)}{\sum_{k'} \exp(\boldsymbol{\beta}^{(k')\top} \boldsymbol{x}_*)}, \tag{2}$$

where $\boldsymbol{\beta} = \{\boldsymbol{\beta}^{(1)} \ldots \boldsymbol{\beta}^{(K)}\} \in \mathbb{R}^{d \times K}$ is the weight matrix of the last fully-connected layer, and K is the total number of relations.

Batch-In-Bag Instance Sampling (BiB). The original IFS performs sampling in a *post-hoc* paradigm. That means, IFS samples from the full training set, then retrains model on the obtained subsamples. However, we argue that this paradigm is not suitable for DS. In post-hoc sampling, all instances are gathered together, the subsamples may be dominated by majority relations with lots of training instances, resulting in severe class imbalance. In an extreme case, minority relations may completely disappear after subsampling. On contrast, in-bag sampling ensures the class ratio being aligned with the full set. Besides, the post-hoc sampling entails an additional iteration for going across all instances in each training epoch, where BiB is more efficient.

In BiB, the subsampling is conducted within bags. Given a bag $X = \{x_1, \ldots, x_n\}$ which contains n sentences, we try to sample a subset X_{sub} that $|X_{sub}| < n$ from X. To this end, we calculate the influences $\Phi_i \; \forall i = 1, \ldots, n$, then get the sampling probabilities π_i by

$$\pi_i = \pi(\Phi_i) := \frac{1}{1 + \exp(\alpha \times \Phi_i)}, \tag{3}$$

where π_i is the probability of x_i being selected and α is a hyper-parameter. Afterwards, we execute sampling to obtain the favorable subset X_{sub}. Consequently, the training objective function $J(\theta)$ is

$$J(\theta) = \frac{1}{|X_{sub}|} \sum_{x_i \in X_{sub}} \ell_i(\theta), \tag{4}$$

where $\ell(\theta)$ is the abbreviation of loss function $\ell(x, y; \theta)$ for notation simplicity.

3.2 Measuring Instance Influence via Influence Function

The core step of above framework is measuring instance's influence Φ. The fundamental intuition behind IFS is that an adverse instance, which causes our model's validation loss increasing, should be given low probability being sampled. We next present Φ's property and substantiate this intuition in a more rigorous way.

Consider a classification problem where we attempt to obtain a model $f_\theta : \mathcal{X} \to \mathcal{Y}$, which is

parametrized by θ, that can make prediction from an input space $\mathcal{X}$ (e.g., sentences) to an output space $\mathcal{Y}$ (e.g., relations). Given a set of training data $\{x_i\}_{i=1}^n$ and the corresponding labels $\{y_i\}_{i=1}^n$, the optimal $\hat{\theta}$ defined by

$$\hat{\theta} := \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell_i(\theta) \tag{5}$$

is the empirical risk minimizer. We usually evaluate the learned model $f_{\hat{\theta}}$ on an additional validation set $\{(x_j^v, y_j^v)\}_{j=1}^m$ such as

$$L(\hat{\theta}) := \frac{1}{m} \sum_{j=1}^{m} \ell_j^v(\hat{\theta}) \tag{6}$$

where $\ell_j^v(\hat{\theta})$ is the validation loss on x_j^v.

In order to quantitatively measure the i-th training sample's influence over model's validation loss, we can perturb the training loss $\ell_i(\theta)$ by a small ϵ, then retrain a perturbed risk minimizer $\tilde{\theta}$ as

$$\tilde{\theta} := \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell_i(\theta) + \epsilon \times \ell_i(\theta). \tag{7}$$

As a result, we are able to compute the validation loss change of the validation sample x_j^v, as

$$\delta_j(\epsilon) := \ell_j^v(\tilde{\theta}) - \ell_j^v(\hat{\theta}) \tag{8}$$

which can be regarded as how much x_i *influences* the prediction on x_j^v. That means, if the $\epsilon = -1/n$, according to Eq. (7), x_i's loss $\ell_i(\theta)$ is actually *removed* from the objective function. In this situation, $\delta_j(\epsilon) > 0$, i.e., $\ell_j^v(\tilde{\theta}) - \ell_j^v(\hat{\theta}) > 0$, implies that removing x_i causes the validation loss on x_j^v increasing, or equivalent

$$\delta_j(-\frac{1}{n}) > 0 \rightarrow x_i \text{ is good for } x_j^v. \tag{9}$$

The influence function $\phi_{i,j} := \phi(x_i, x_j^v; \hat{\theta})$ can be used to linearly approximate $\delta_j(\epsilon)$ by

$$\delta_j(\epsilon) = \ell_j^v(\tilde{\theta}) - \ell_j^v(\hat{\theta}) \simeq \epsilon \times \phi_{i,j} \tag{10}$$

where the closed-form expression of ϕ is given by Koh and Liang (2017) as

$$\phi_{i,j} := -\nabla_\theta \ell_j^v(\hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta \ell_i(\hat{\theta}) \tag{11}$$

and $H_{\hat{\theta}} := \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 \ell_i(\hat{\theta})$ is the Hessian matrix.

Bear in mind that in Eq. (10), if $\delta_j(-1/n) > 0$, i.e., $\phi_{i,j} < 0$, we can compute x_i's influence over the whole validation set by summation as

$$\Phi_i = \sum_{j=1}^{m} \phi_{i,j} = -\sum_{j=1}^{m} \nabla_\theta \ell_j^{v\top}(\hat{\theta}) H_{\hat{\theta}}^{-1} \nabla_\theta \ell_i(\hat{\theta}), \tag{12}$$

such that $\Phi_i < 0$ indicates that x_i is *good* for the whole validation set in average. And, if Φ_i is smaller, then x_i is more likely to be a favorable sample, vice versa. Hence, Φ can be regarded as a reasonable measure of the sample's influence.

3.3 Probabilistic Sigmoid Subsampling: A Theoretical Perspective

With the influence measure Φ, there are two genres of performing IFS: deterministic (Wang et al., 2018) and probabilistic (Wang et al., 2020). The deterministic method simply drops all *unfavorable* samples that have $\Phi > 0$. However, Wang et al. (2020) argued that using 0 as the threshold usually results in failure to the out-of-sample test, because it is sensitive to distribution shift. Instead, they advocated to design a probabilistic sampling function $\pi(\Phi)$ for subsampling. In this work, we follow the probabilistic sampling, along with constructive analysis for the probabilistic sampling, which suggests the use of sigmoid sampling.

Our analysis centers around the deviation of the induced validation loss by inaccurate estimate of influence. Let's denote the validation loss with inaccurate influence by $\ell^v(\tilde{\theta}; \hat{\Phi})$, thus

$$\Delta^2(L) := \frac{1}{m} \sum_{j=1}^{m} (\ell_j^v(\tilde{\theta}; \hat{\Phi}) - \ell_j^v(\tilde{\theta}))^2 \tag{13}$$

indicates the robustness of the model under $\hat{\Phi}$. We then give the following proposition on $\Delta^2(L)$ with respect to sampling function π. Proof can be found in Appendix A.

Proposition 1 (Robustness of Probabilistic Sampling under Inaccurate Influence)**.** *Let $\pi'(\Phi_i)$ be the derivative of $\pi(\cdot)$ function when taking Φ_i as its input, we have*

$$\begin{aligned} \sup_{\Phi, \hat{\Phi}} \Delta^2(L) &= \gamma \sum_{i=1}^{n} (\pi(\hat{\Phi}_i) - \pi(\Phi_i))^2 \sum_{j=1}^{m} \phi_{i,j}^2 \\ &\simeq \gamma \sum_{i=1}^{n} \Big((\hat{\Phi}_i - \Phi_i) \pi'(\Phi_i) \Big)^2 \sum_{j=1}^{m} \phi_{i,j}^2 \end{aligned} \tag{14}$$

where γ is a constant.

Algorithm 1 Finding Influential Instances for DS on RE by Influence Subsampling.

Require: Training and validation data $\mathcal{D}_{tr}, \mathcal{D}_{va}$; Hyper-parameters: r and α;

for epoch $t = 1 \to T$ **do**
  repeat
    Initialize the selected instances set $X_{sub} = \emptyset$;
    Sequentially sample a batch of bags $\{X_1, \ldots, X_B\}$ from $\mathcal{D}_{tr}$;
    for bag $b = 1 \to B$ **do**
      Obtain instance-level loss as $\vec{\ell} \leftarrow (\ell_1(\hat{\theta}_t), \ldots, \ell_{|X_b|}(\hat{\theta}_t))^\top$;
      Compute influences $\Phi_i \leftarrow s_t^\top \nabla_\theta \ell_i(\hat{\theta}_t)\ \forall i = 1, \ldots, |X_b|$;
      Compute sampling probability $\pi_i \leftarrow 1/(1 + \exp(\alpha \times \Phi_i))\ \forall i$;
      Sample $r \times |X_b|$ instances from X_b to get $\tilde{X}_b$, and $X_{sub} \leftarrow X_{sub} \cup \tilde{X}_b$;
    end for
    Update $\hat{\theta}_t$ using the selected subset X_{sub} by gradient descent;
  until going through all bags in $\mathcal{D}_{tr}$.
  Get validation loss by $L(\hat{\theta}_t) \leftarrow \frac{1}{m}\sum_{j=1}^{m} \ell_j^v(\hat{\theta}_t)$ on $\mathcal{D}_{va}$;
  Obtain $s_t \leftarrow H_t^{-1} \nabla_\theta L(\hat{\theta}_t)$ by stochastic estimation as done in Eq. (19);
end for

It can be viewed that $\Delta^2(L)$ is controlled by the derivative of sampling function $\pi'(\Phi)$. For the sigmoid sampling in Eq. (3), it is easy to derive that

$$\pi'(\Phi) = -\alpha\pi(\Phi)(1 - \pi(\Phi)), \tag{15}$$

which means $\max|\pi'(\Phi)| = \frac{1}{4}\alpha$ when $\Phi = 0$. $\Delta^2(L)$ is hence controlled by the hyper-parameter α. When $|\Phi|$ increases, $|\pi'(\Phi)|$ reduces sharply, which ensures the variance's upper bound being tight all the time, thus enhancing the robustness of the subsampling process. By contrast, in deterministic sampling, $\Delta^2(L)$ is sensitive to inaccurate $\hat{\Phi}$ because it is "hard", or more rigorously, because $\Delta^2(L)$ is probably large due to large $|\pi(\Phi) - \pi(\hat{\Phi})|$ caused by an improper dropout threshold.

4 Efficient Implementation

Recall that in Eq. (12), for computing Φ_i for $i = 1, \ldots, n$, it first computes $\phi_{i,j}$ for $j = 1, \ldots, m$ on all validation samples, based on the influence function given by Eq. (11), then sums $\phi_{i,j}$ up to obtain Φ_i, resulting in time complexity of $\mathcal{O}(mn)$. Moreover, for deep neural networks with massive parameters, computing the layer-wise gradients $\nabla_\theta \ell(\theta)$ is complicated. Such that, the original IFS is not applicable for deep learning based RE models.

To address it, we here propose an efficient implementation of our REIF framework. In particular, we demonstrate how to reduce the complexity of calculating influences from $\mathcal{O}(mn)$ to $\mathcal{O}(n)$, then to $\mathcal{O}(1)$. In addition, we present how to compute the influence function by stochastic estimation, and conclude the whole algorithm at last.

4.1 Computing Influences in Linear Time

We argue that in Eq. (12), it is unnecessary to calculate $\phi_{i,j}$ separately, since here we actually only care about their summation. Specifically, since the summation is only related to the subscript j, we can cast it to

$$\begin{aligned}\Phi_i &= -\nabla_\theta \ell_i^\top(\hat{\theta}) H_{\hat{\theta}}^{-1} \sum_{j=1}^{m} \nabla_\theta \ell_j^v(\hat{\theta}) \\ &= -\nabla_\theta \ell_i^\top(\hat{\theta}) H_{\hat{\theta}}^{-1} \nabla_\theta \sum_{j=1}^{m} \ell_j^v(\hat{\theta}) \\ &= -m\nabla_\theta \ell_i^\top(\hat{\theta}) H_{\hat{\theta}}^{-1} \nabla_\theta L(\hat{\theta}),\end{aligned} \tag{16}$$

where $L(\hat{\theta})$ comes from Eq. (6). By this derivation, we can calculate $L(\hat{\theta})$ rather than all $l_j(\hat{\theta})$, then take derivative of $L(\hat{\theta})$. Since $L(\hat{\theta})$ only needs to be calculated once and it is shared in calculating all Φ_is, this process only requires $\mathcal{O}(n)$ time, without loss of accuracy.

4.2 Linear Approximation for $\mathcal{O}(1)$ Complexity

The term $\nabla_\theta \ell(\hat{\theta})$ in Eq. (16) usually has complicated expression when $f_\theta(\cdot)$ is a neural network, hence the previous works implemented it by the auto-grad systems like TensorFlow (Abadi et al., 2016) and PyTorch (Paszke et al., 2019). However, when the number of alternative training instances

is large, even $\mathcal{O}(n)$ is not satisfactory enough, because additional differential operations need to be done on each $\ell_i(\hat{\theta})$ sequentially. Moreover, when faced with complex neural networks with massive parameters, computing the Hessian matrix $H_{\hat{\theta}}$ and its inverse is intractable. Considering these issues, we propose a linear approximation approach to reduce the complexity to $\mathcal{O}(1)$, and avoid operating on all parameters of the neural network.

Suppose the cross entropy loss function is used:

$$\ell(\theta) = -\sum_{k=1}^{K} \mathbb{I}\{y = k\} \log P(y = k|x; \theta) \quad (17)$$

where $\mathbb{I}(\cdot)$ is an indicator function. Let $\boldsymbol{y}, \hat{\boldsymbol{y}} \in \mathbb{R}^K$ be the one-hot label vector, e.g., $(1, 0, 0)^\top$, and prediction vector, e.g., $(0.8, 0.1, 0.1)^\top$, respectively. We replace $\nabla_\theta \ell(\theta)$ in Eq. (11) with the derivatives on $\boldsymbol{\beta}$ (the weight of the last fully-connected layer):

$$\nabla_\theta \ell(\theta) \Rightarrow \nabla_{\boldsymbol{\beta}} \ell(\theta) = (\hat{\boldsymbol{y}} - \boldsymbol{y})\boldsymbol{x}^\top \in \mathbb{R}^{d \times K} \quad (18)$$

where $\boldsymbol{x}$ is the input of the last fully-connected layer. This closed-form expression allows computing batch gradients in $\mathcal{O}(1)$ time. Although the calculated influence might be inaccurate, it is still reliable for measuring instances' *relative quality* in general. We will validate this claim in our experiments.

4.3 Algorithm

Algorithm 1 shows the details of REIF, with two hyper-parameters: the sampling ratio r and the sigmoid sampling parameter α. The optimal value of r depends on quality of the dataset, since the higher quality it is, the more favorable instances it might have. For α, keeping $\alpha = 1$ is satisfactory in most scenarios.

In particular, on line #14 of Algorithm 1, we compute the product between the inverse Hessian matrix and a gradient vector via the stochastic estimation procedure by Koh and Liang (2017). Denoting the vector $\nabla_\theta L(\hat{\theta})$ by v, it first initializes the approximate inverse Hessian-Vector-Product (HVP) by $\tilde{H}_0^{-1} v \leftarrow v$, then repeatedly samples n_b training instances and updates as

$$\tilde{H}_t^{-1} v \leftarrow v + (I - \frac{1}{n_b} \sum \nabla_\theta^2 \ell(\hat{\theta})) \tilde{H}_{t-1}^{-1} v \quad (19)$$

until $\tilde{H}_t^{-1} v$ converges. In our algorithm, we only need to do this once after each epoch, to get the precomputed inverse HVP $s = H_{\hat{\theta}}^{-1} \nabla_\theta L(\hat{\theta})$. Therefore, during training, we directly compute $\nabla_\theta \ell_i(\hat{\theta})$ for each instance according to Eq. (18), then multiply it with the precomputed s.

5 Experiments

Our proposed REIF is model-agnostic, thus can be incorporated into the majority of RE models, e.g., PCNN (Zeng et al., 2015). We concentrate on the following research questions:

RQ1. How does our REIF perform as compared with classical baselines?

RQ2. How does the sampling ratio r influence the performance of the REIF?

RQ3. Does the sigmoid function lead to more robust sampling than the deterministic sampling?

RQ4. How does the proposed batch-in-bag sampling perform compared with the post-hoc sampling used in original IFS methods?

5.1 Datasets

In our experiments, we use two versions of widely used NYT datasets, the NYT-SMALL and NYT-LARGE. The small version is released in Riedel et al. (2010), by aligning Freebase with the New York Times corpus. In particular, we use the filtered version of the NYT-SMALL released by Zeng et al. (2015). The large version was released in (Lin et al., 2016) with further augmentation in training set. Data statistics can be found in Appendix B.

5.2 Experimental Setups

We pick PCNN (PCNN+ONE) (Zeng et al., 2015) as the backbone in our experiments, and include several baselines for comparison: the attention-based PCNN (PCNN+ATT) and the naive average method (PCNN+AVE) (Lin et al., 2016). Note that our REIF method is model-agnostic, hence it is applicable for other deep learning based backbones as well, e.g., CNN and RNN. Setups of models can be found in Appendix C.

We sample a clean validation set from training set by a rule-based approach used in (Jia et al., 2019), in order to obtain the inverse HVP required for calculating influences. During subsampling, we set $\alpha = 1$ and $r = 10\%$[2] for our REIF.

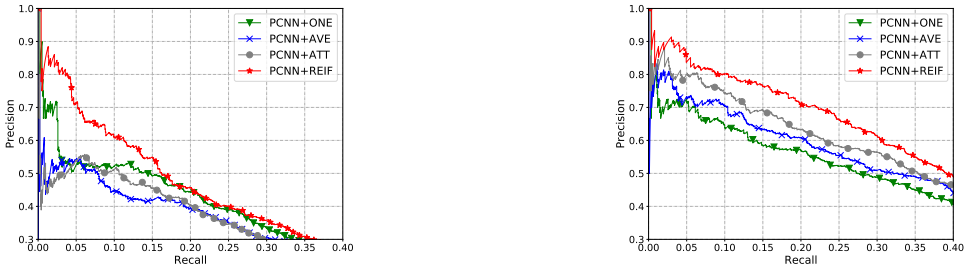
5.3 Effects of Influence Subsampling (RQ1)

Fig. 3 shows the precision-recall curve in held-out evaluation of ONE, AVE, ATT, and our REIF, and Table 1 illustrates the corresponding P@N of all methods. Our REIF performs the best among all

[2]The ceiling function is used for rounding.

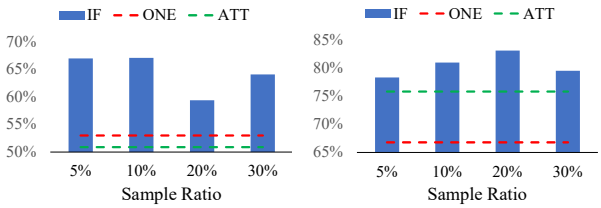
Table 1: P@N for relation extraction results, on NYT-SMALL and NYT-LARGE, where the best ones are in bold.

Dataset	NYT-SMALL				NYT-LARGE			
P@N (%)	100	200	300	Mean	100	200	300	Mean
PCNN + ONE	54.0%	52.7%	52.2%	53.0%	70.4%	66.4%	63.6%	66.8%
PCNN + AVE	52.7%	50.8%	47.3%	50.3%	73.0%	71.2%	67.8%	70.6%
PCNN + ATT	52.7%	50.7%	49.5%	50.9%	79.7%	76.0%	71.6%	75.8%
PCNN + REIF (Proposed)	**75.2%**	**65.1%**	**60.8%**	**67.0%**	**85.0%**	**80.2%**	**77.7%**	**80.9%**



(a) NYT-SMALL (b) NYT-LARGE

Figure 3: Aggregated precision-recall (P-R) curves obtained by PCNN+ONE, PCNN+AVE, PCNN+ATT, and the proposed PCNN+REIF on NYT-SMALL (left) and NYT-LARGE (right) datasets.



(a) NYT-SMALL (b) NYT-LARGE

Figure 4: Mean P@N (average of P@100/200/300) varies with sampling ratio of REIF (IF) method.

methods. In details, on NYT-SMALL, our REIF improves 14% over ONE, and 16.1% over ATT; on NYT-LARGE, the improvements are 14.1% and 5.1%, respectively, in terms of the mean P@N. It should be noted that our REIF only leverages 10% instances during training, while ATT involves all instances but performs badly on NYT-SMALL, while ONE only picks one instance per bag. It indicates that neither picking too many nor too few instances gains satisfactory performance in distant supervision. On contrast, our REIF can detect and pick those favorable ones from the noisy dataset, thus achieving a better model. In distant supervision, our method is effective for achieving nice trade-off between efficiency and effectiveness.

5.4 Effects of Sampling Ratio (RQ2)

We keep sampling ratio $r = 10\%$ in above experiments, which means only 10% of instances in each bag are picked in each epoch. We evaluate the performance of REIF with respect to different r. Results are reported in Fig. 4. REIF keeps stable when sampling ratio ranges from 5% to 30%, such that adding more instances does not make much difference, which might be due to high noise in the NYT dataset, i.e., focusing on those favorable instances is enough for training a satisfactory RE model.

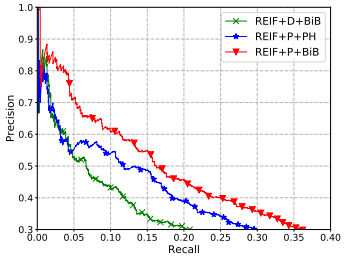
5.5 Effects of Sigmoid Sampling & Batch-In-Bag Sampling (RQ3, RQ4)

Our REIF is engaged with the proposed probabilistic sigmoid sampling and batch-in-bag sampling, namely REIF+P+BiB. We would like to validate these two techniques compared with the deterministic sampling (REIF+D+BiB), and the post-hoc sampling (REIF+P+PH). Our main observations from Fig. 5 are as follows:

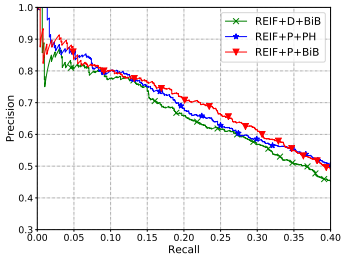
(1) The probabilistic sigmoid sampling is crucial for robust subsampling, as the REIF+D+BiB performs the worst in both datasets. As mentioned in Proposition 1, drawbacks of REIF+D mainly come from inaccurate estimate of influence $\hat{\Phi}$, due to the non-convexity of neural networks and the use of linear approximations. That is, we could not determine the instances that have $\hat{\Phi}$ around the threshold with very high confidence, e.g., deterministic ranking and selecting, since this causes high variance of the resulting test loss, as indicated in Eq. (14). By contrast, we should assign them similar probabilities to be sampled, as done in REIF-P, to avoid sharp variation of the test loss caused by inaccurate influences in deterministic selection.

Table 2: Examples of influences calculated with the relation *children*, on the NYT-LARGE corpus. The words in bold are entities. The *Att Scores* (Lin et al., 2016) are standardized into $[0, 1]$ by softmax, and *Influences* are the smaller the better.

Instances	Influences	Att Scores
... because of **art rooney**, the legendary steelers' owner ... be a family under his oldest son, **dan rooney**.	-2.23E-02	1.11E-04
... mother of **joseph** paula and walter eva, grandmother of david, lauren, **jacob**, miriam and leah.	-1.07E-04	2.61E-09
... the suspense novelists **mary higgins clark** and **carol higgins clark** signed books and posed for photographs for five hours here ...	1.50E-05	1.44E-07
... daughter jamie baldinger and her husband, **joseph**; son david goldring and his wife **rachel** ...	7.81E-04	1.39E-09



(a) NYT-SMALL



(b) NYT-LARGE

Figure 5: precision-recall curve of compared REIF variants, where the REIF+P+BiB is the REIF with probabilistic sigmoid sampling and batch-in-bag sampling, +D means deterministic sampling and +PH means post-hoc sampling.

(2) Our bag-in-batch sampling method generally performs better than the post-hoc sampling in DS, especially on the tail instances. When recall is high, REIF+BiB performs better on the minor relations, thus has higher precision than REIF+PH. In BiB, more minor relation instances are maintained, which facilitates the model's capacity of mining minor relation instances. Considering efficiency and the overall effectiveness, we shall prefer BiB in practice.

6 Case Study

Table 2 reports an example of calculating influences that support instance selection in our method. Picking a relation *children* as the example, influences and attention scores (Lin et al., 2016) are computed, from which we can identify that the influences quantitatively measure their individual quality. Recall in Section 3.2 that the smaller influences indicate better data quality. The first and the last instances are clearly right and wrong, respectively, in terms of indicating the relation *children* between their entities. By contrast, the second one tends to be right because it implies that *Joseph* is the parent of *Jacob*. Although two entities in the third instance are very similar, no evidence shows they are relatives. Therefore, sampling probabilities can be obtained via these influences for the further subsampling process.

On the other hand, the instance's attention score for the bag's relation is uninformative, thus are not capable of discriminating these instances' quality. Our IFS method thus shows its superiority for interpretable instance selection for building reliable knowledge bases.

7 Conclusion & Discussion

In this work, we employed the influence function for measuring instance quality, then proposed an efficient subsampling scheme to find the influential instances for training, namely REIF. Our method is model-agnostic, therefore it can be engaged in the majority of RE models. We argue that our method of finding influential instances is not merely limited in relation extraction, but can be generalized to other tasks that confront noisy data or distant supervision. For instance, in other weak supervision scenarios where data are not with high quality or high confidence, such as crowdsourcing, our method can be an effective approach to build data pipeline from data quality measure to data selection. We leave this as our future work.

Acknowledgments

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distant supervision for relation extraction via instance-level adversarial training. *arXiv preprint arXiv:1805.10959*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 541–550.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. ARNOR: Attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1399–1408.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2124–2133.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1011. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018a. DSGAN: Generative adversarial training for distant supervision relation extraction. *arXiv preprint arXiv:1805.09929*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018b. Robust distant supervision relation extraction via deep reinforcement learning. *arXiv preprint arXiv:1805.09927*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163.

Boris Sharchilev, Yury Ustinovsky, Pavel Serdyukov, and Maarten de Rijke. 2018. Finding influential training samples for gradient boosted decision trees. *arXiv preprint arXiv:1802.06640*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 721–729.

Tianyang Wang, Jun Huan, and Bo Li. 2018. Data dropout: Optimizing training data for convolutional neural networks. In *IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 39–46.

Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, and Shao-Lun Huang. 2020. Less is better: Unweighted data subsampling via influence function. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yi Wu, David Bamman, and Stuart Russell. 2017. Adversarial training for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1778–1783.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169.

Yujin Yuan, Liyuan Liu, Siliang Tang, Zhongfei Zhang, Yueting Zhuang, Shiliang Pu, Fei Wu, and Xiang Ren. 2019. Cross-relation cross-bag attention for distantly-supervised relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 419–426.

Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Peng Zhou, Jiaming Xu, Zhenyu Qi, Hongyun Bao, Zhineng Chen, and Bo Xu. 2018. Distant supervision for relation extraction with hierarchical selective attention. *Neural Networks*, 108:240 – 247.

A Proof of Proposition 1

Proposition 1 (Robustness of Probabilistic Sampling under Inaccurate Influence)**.** *Let $\pi'(\Phi_i)$ be the derivative of $\pi(\cdot)$ function when taking Φ_i as its input, we have*

$$\sup_{\Phi,\hat{\Phi}} \Delta^2(L) = \gamma \sum_{i=1}^{n} (\pi(\hat{\Phi}_i) - \pi(\Phi_i))^2 \sum_{j=1}^{m} \phi_{i,j}^2 \simeq \gamma \sum_{i=1}^{n} \left((\hat{\Phi}_i - \Phi_i)\pi'(\Phi_i) \right)^2 \sum_{j=1}^{m} \phi_{i,j}^2 \tag{A.1}$$

where γ is a constant.

Proof.

$$\Delta^2(L) \propto \sum_{j=1}^{m} (\ell_j^v(\tilde{\theta}; \hat{\Phi}) - \ell_j^v(\tilde{\theta}))^2 \tag{A.2}$$

$$= \sum_{j=1}^{m} (\ell_j^v(\tilde{\theta}; \hat{\Phi}) - \ell_j^v(\hat{\theta}) + \ell_j(\hat{\theta}) - \ell_j^v(\tilde{\theta}))^2 \tag{A.3}$$

$$\propto \sum_{j=1}^{m} \left(\sum_{i=1}^{n} \pi(\hat{\Phi}_i)\phi_{i,j} - \pi(\Phi_i)\phi_{i,j} \right)^2 \tag{A.4}$$

$$\leq \sum_{i=1}^{n} (\pi(\hat{\Phi}_i) - \pi(\Phi_i))^2 \sum_{j=1}^{m} \phi_{i,j}^2 \tag{A.5}$$

Eq. (A.4) is obtained by definition of probabilistic subsampling because

$$\ell_j^v(\tilde{\theta}) - \ell_j^v(\hat{\theta}) \simeq \sum_{i=1}^{n} \epsilon_i \phi_{i,j} \propto \sum_{i=1}^{n} \pi(\Phi_i)\phi_{i,j}. \tag{A.6}$$

Details can be referred to Wang et al. (2020). Taking linear Taylor expansion of the $\pi(\hat{\Phi}_i) - \pi(\Phi_i)$ at the last line yields the final result. □

B Dataset Statistics

Table 1: Data statistics of used two NYT datasets. "# Pos", "# Ins", "# Rel": number of postive bags, instances and relations, respectively.

	NYT-SMALL		NYT-LARGE	
	Train	Test	Train	Test
# Bags	65,726	93,574	281,270	96,678
# Pos	4,266	1,732	18,252	1,950
# Ins	112,941	152,416	522,611	172,448
# Rel	26	26	53	53

C General Setups for Training PCNN

Following the configurations of previous works, we employ word2vec[3] to extract the word embeddings, to process the raw data. Parameters of PCNN are set according to (Zeng et al., 2015): window size $d^w = 3$, sentence embedding size $d^s = 230$, word dimension $d^a = 50$ and position dimension $d^p = 5$ for fair comparison. During training, we fix the batch size $B = 128$, dropout ratio $p = 0.5$, and use the ADADELTA (Zeiler, 2012) with parameters $\rho = 0.95$ and $\varepsilon = 10^{-6}$ for optimization.

[3] https://code.google.com/p/word2vec/