Looking Beyond Label Noise: Shifted Label Distribution Matters in Distantly Supervised Relation Extraction

Qinyuan Ye*
University of Southern California
qinyuany@usc.edu

Liyuan Liu*
University of Illinois, Urbana-Champaign
ll2@illinois.edu

Maosen Zhang
Purdue University
maosenzhang.milo@gmail.com

Xiang Ren
University of Southern California
xiangren@usc.edu

Abstract

In recent years there is a surge of interest in applying distant supervision (DS) to automatically generate training data for relation extraction (RE). In this paper, we study the problem *what limits the performance of DS-trained neural models*, conduct thorough analyses, and identify a factor that can influence the performance greatly, *shifted label distribution*. Specifically, we found this problem commonly exists in real-world DS datasets, and without special handing, typical DS-RE models cannot automatically adapt to this shift, thus achieving deteriorated performance. To further validate our intuition, we develop a simple yet effective adaptation method for DS-trained models, *bias adjustment*, which updates models learned over the source domain (*i.e.*, DS training set) with a label distribution estimated on the target domain (*i.e.*, test set). Experiments demonstrate that bias adjustment achieves consistent performance gains on DS-trained models, especially on neural models, with an up to 23% relative F1 improvement, which verifies our assumptions.

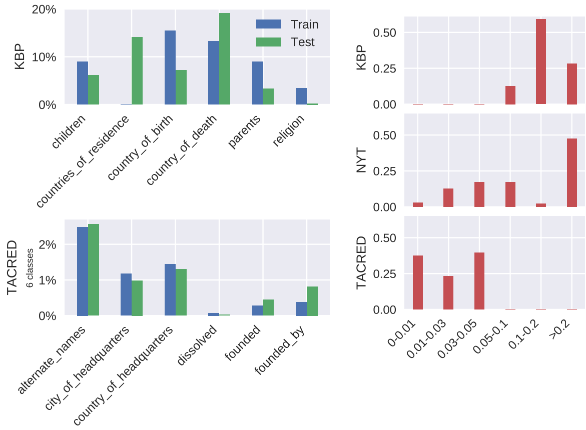

Figure 1: Left: Label distributions of KBP (distantly supervised dataset) are shifted, while those of TACRED (human-annotated dataset, 6 classes for illustration) are consistent. Full distributions for NYT and TACRED can be found in Appendix B. **Right:** Each relation r_i is categorized into intervals along x-axis according to $|p(r_i|\mathcal{D}_{train}) - p(r_i|\mathcal{D}_{test})|$; height of the bars indicates proportion of instances that fall into each category, revealing that the shift is severe in DS datasets such as KBP and NYT.

1 Introduction

Aiming to identify the relation among an entity pair, relation extraction (RE) serves as an important step towards text understanding and is a long-standing pursuit by many researchers. To reduce the reliance on human-annotated data, especially for data-hungry neural models (Zeng et al., 2014; Zhang et al., 2017), there have been extensive studies on leveraging *distant supervision* (DS) in conjunction with external knowledge bases to automatically generate large-scale training data (Mintz et al., 2009; Zeng et al., 2015). While recent DS-based relation extraction methods focus on handling *label noise* (Riedel et al., 2010; Hoffmann et al., 2011; Lin et al., 2016), *i.e.*, false labels introduced by the error-prone DS process, other factors may have been overlooked. Here, we observe model behaviors to be different on DS datasets and clean dataset, which implies existence of other challenges that restrict DS-RE performance. In this paper, we conduct thorough analyses over both real-world and synthetic datasets to explore the question — *what limits the performance of DS-trained neural models*.

Our analysis starts with a performance comparison among recent relation extraction methods on both DS datasets (*i.e.*, KBP (Ellis et al., 2012), NYT (Riedel et al., 2010)) and human-annotated dataset (*i.e.*, TACRED (Zhang et al., 2017)), with the goal of seeking models that can consistently yield strong results. We observe that, on human-annotated dataset, neural relation extraction mod-

*Equal contribution.

els outperform feature-based models by notable gaps, but these gaps diminish when the same models are applied to DS datasets—neural models merely achieve performance comparable with feature-based models. We endeavor to analyze the underlying problem that leads to this unexpected "diminishing" phenomenon.

Inspired by two heuristic threshold techniques that prove to be effective on DS datasets, and further convinced by comprehensive analysis results on synthetic datasets, we reveal an important factor of DS datasets—*shifted label distribution*, the issue that label distribution of training set does not align with that of test set. There often exists a large margin between label distributions of *distantly-supervised training set* and that of *human-annotated test set*, as shown in Figure 1. Intuitively, label distribution shift is mainly caused by false-positive and false-negative labels generated by the error-prone DS process, and the imbalanced data distribution of external knowledge bases.

To some extent, such distortion is a special case of domain shift —*i.e.*, training the model on a source domain and applying the learned model to a different target domain. To further verify our assumption, we develop a simple domain adaption method, *bias adaptation*, to address the shifted label distribution. It modifies the bias term in classifiers and explicitly fits models along the shift. Specifically, the proposed method estimates the label distribution of target domain with a small development set sampled from test set, and derives the adapted predictions under reasonable assumptions. In our experiments, we observe consistent performance improvement, which validates that model performance may be severely hindered by label distribution shift.

In the rest of the paper, we will first introduce the problem setting and report the inconsistency of model performance with human annotations and distant supervision. Then, we will present the two threshold techniques which are found to be effective on DS datasets and further lead us to the discovery of shifted label distribution. We further explore its impact on synthetic datasets in Section 4, and introduce the bias adjustment method in Section 5. In addition, comparison of denoising method, heuristic threshold and bias adjustment is conducted in Section 6.

Dataset	Distantly Supervised KBP	Distantly Supervised NYT	Human-annotated TACRED
#Relation Types	7	25	42
#Train Sentences	23,784	235,982	37,311
#Test Sentences	289	395	6,277

Table 1: Statistics of Datasets Used in Our Study.

2 Experiment Setup

In this paper, we conduct extensive empirical analyses on distantly supervised relation extraction (DS-RE). For a meaningful comparison, we ensure the same setup in all experiments. In this section, we provide a brief introduction on the setting, while more details could be found in Appendix A. All implementations are available at `https://github.com/INK-USC/DS-RelationExtraction`.

2.1 Problem Setting

Following previous works (Ren et al., 2017; Liu et al., 2017), we conduct relation extraction at *sentence level*. Formally speaking, the basic unit is the relation mention, which is composed of one sentence and one ordered entity pair within the sentence. The relation extraction task is to categorize each relation mention into a given set of relation types, or a Not-Target-Type (NONE).

2.2 Datasets

We conduct experiments on three popular relation extraction benchmarks. Specifically, KBP (Ling and Weld, 2012) and NYT (Riedel et al., 2010) are distantly-supervised; TACRED (Zhang et al., 2017) is human-annotated. Statistics of these datasets are summarized in Table 1.

2.3 Pre-processing

We leverage pre-trained GloVe (Pennington et al., 2014) embedding[1], and use the StanfordNLP toolkit (Manning et al., 2014) to get part of speech (POS) tags, named-entity recognition (NER) tags, and dependency parsing trees.

For the development sets, we use the provided development set on TACRED, and randomly split 10% of training set on KBP and NYT. This development set is also refered to as *noisy dev*, as we will introduce a *clean dev* to deal with shifted label distribution in the later parts.

[1] `http://nlp.stanford.edu/data/glove.840B.300d.zip`

Method / Dataset		Distantly-supervised Wiki-KBP	NYT	Human-annotated TACRED
Feature-based	CoType-RM (Ren et al., 2017)	28.98 ± 0.76	40.26 ± 0.51	45.97 ± 0.34
	ReHession (Liu et al., 2017)	36.07 ± 1.06	46.79 ± 0.75	58.06 ± 0.54
	Logistic (Mintz et al., 2009)	37.58 ± 0.27	47.33 ± 0.44	51.67 ± 0.03
Neural	CNN (Zeng et al., 2014)	30.53 ± 2.26	46.75 ± 2.79	56.96 ± 0.43
	PCNN (Zeng et al., 2015)	31.58 ± 0.35	44.63 ± 2.70	58.39 ± 0.71
	Bi-GRU	**37.77 ± 0.18**	47.88 ± 0.85	65.38 ± 0.60
	Bi-LSTM	34.51 ± 0.99	48.15 ± 0.87	62.74 ± 0.23
	PA-LSTM (Zhang et al., 2017)	37.28 ± 0.81	46.33 ± 0.64	**65.69 ± 0.48**
	Bi-GRU-ATT (Lin et al., 2016)	37.50 ± 1.21	**49.67 ± 1.06**	-
	PCNN-ATT (Lin et al., 2016)	33.74 ± 2.19	46.82 ± 0.82	-

Table 2: Performance Comparison of Relation Extraction Models (F1 score ± std).

2.4 Models

We consider two classes of relation extraction methods, *i.e.*, feature-based and neural models.

Specifically, Feature-based models include Logistic Regression (Mintz et al., 2009), CoType-RM (Ren et al., 2017) and ReHession (Liu et al., 2017). Neural models include Bi-LSTMs and Bi-GRUs (Zhang et al., 2017), Position-Aware LSTM (Zhang et al., 2017), CNNs and PCNNs (Zeng et al., 2014, 2015).

For each relation mention, these models will first construct a representation vector $\mathbf{h}$, and then make predictions with softmax based on $\mathbf{h}$[2]:

$$p(y = r_i|\mathbf{h}) = \frac{\exp(\mathbf{r}_i^T \mathbf{h} + b_i)}{\sum_{r_j} \exp(\mathbf{r}_j^T \mathbf{h} + b_j)}, \quad (1)$$

where $\mathbf{r}_i$ and b_i are the parameters corresponding to i-th relation type.

More details on these models can be found in Appendix A.

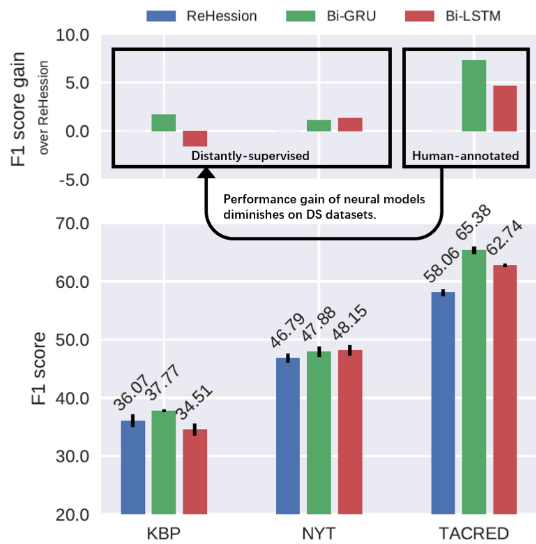

Figure 2: F1 score comparison among three models. Bi-GRU outperforms ReHession with a significant gap on TACRED, but only has comparable performance (with ReHession) on KBP and NYT. Similar gap diminishing phenomenon happens to Bi-LSTM.

3 The Diminishing Phenomenon

Neural models alleviate the reliance on hand-crafted features and have greatly advanced the state-of-the-art, especially on datasets with human annotations. Meanwhile, we observe such performance boost starts to diminish on distantly supervised datasets. Specifically, we list the performance of all models in Table 2 and summarize three popular models in Figure 2.

On TACRED, a human annotated dataset, complex neural models like Bi-LSTM and Bi-GRU significantly outperform feature-based ReHession, with an up to 13% relative F1 improvement. On the other hand, on distantly supervised datasets (KBP and NYT), the performance gaps between the aforementioned methods diminishes to within 5%. We refer to this observation as "diminishing" phenomenon. Such observation implies a lack of handling of the underlying difference between human annotations and distant supervision.

After a broad exploration, we found two heuristic techniques that we believe capture problems exclusive to distantly supervised RE, and are potentially related to "diminishing" phenomenon. We found they can greatly boost the performance on distantly supervised datasets, but fail to do so on human-annotated dataset. To get deeper insights, we analyze the diminishing phenomenon and the two heuristic methods.

[2]CoType-RM and logistic regression are exceptions as they don't adopt softmax to generate output.

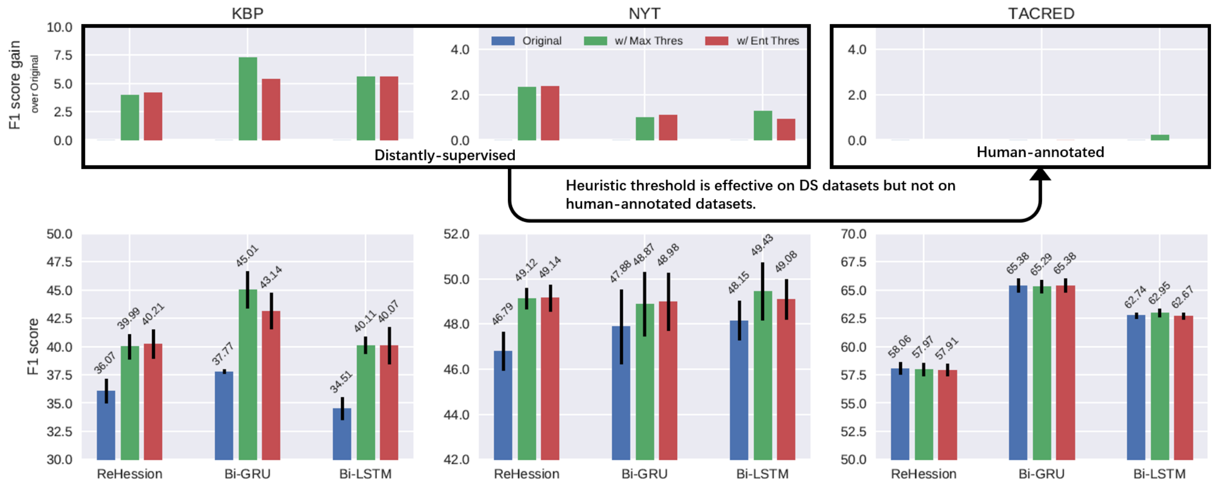

Figure 3: F1 scores on three datasets when (a) no threshold (original) (b) max threshold (c) entropy threshold are applied, respectively. A clear boost is observed on distantly supervised datasets (*i.e.*KBP and NYT) after applying threshold; however, performance n human-annotated dataset (*i.e.*TACRED) remains almost the same with thresholding. Heuristic threshold techniques may capture some important but overlooked problems in distantly supervised relation extraction.

3.1 Heuristic Threshold Techniques

Max threshold and entropy threshold are designed to identify the "ambiguous" relation mentions (*i.e.*, predicted with a low confidence) and label them as the NONE type (Ren et al., 2017; Liu et al., 2017). In particular, referring to the original predictions as $r^* = \arg\max_{r_i} p(y = r_i|\mathbf{h})$, we formally introduce these two threshold techniques:

- **Max Threshold** introduces an additional hyper-parameter T_m, and adjusts the prediction as (Ren et al., 2017):

$$predict(\mathbf{h}) = \begin{cases} r^*, & p(y = r^*|\mathbf{h}) > T_m \\ \text{NONE}, & \text{Otherwise} \end{cases}.$$

- **Entropy Threshold** introduces an additional hyper-parameter T_e. It first calculates the entropy of prediction:

$$e(\mathbf{h}) = -\sum_{r_k} p(y = r_k|\mathbf{h}) \log p(y = r_k|\mathbf{h}),$$

then it adjusts prediction as (Liu et al., 2017):

$$predict(\mathbf{h}) = \begin{cases} r^*, & e(\mathbf{h}) < T_e \\ \text{NONE}, & \text{Otherwise} \end{cases}.$$

To estimate T_e or T_m, 20% instances are sampled from the test set as an additional development set, and used to tune the value of T_e and T_m with grid search. After that, we would evaluate the model performance on the rest (80%) of the test set. We refer to this new dev set as *clean dev* and refer to the original dev set used for hyper-parameter tuning as *noisy dev*. We would like to highlight that tuning threshold on this clean dev is necessary as it acts as a bridge between distantly supervised train set and human-annotated test set. We found that, these two techniques only result in marginal differences on TACRED dataset, but lead to significant performance gains on both KBP and NYT datasets. This phenomenon implies that these technologies may capture some unique characteristics of distant supervision, and we will discuss more details as below.

3.2 Results and Discussion

Results of three representative models (ReHession, Bi-GRU and Bi-LSTM) are summarized in Figure 3. Full results are listed in Table 3.

We observe significant improvements on distantly supervised datasets (*i.e.*, KBP and NYT), with a up to 19% relative F1 improvement (Bi-GRU from 37.77% to 45.01% on KBP). However, on the human-annotated corpus, the performance gain can be hardly noticed. Such inconsistency implies that these heuristics may capture some important but overlooked factors for distantly supervised relation extraction, while we are still unclear about their underlying mechanisms.

Intuitively, distant supervision differs from human annotations in two ways: (1) **False Positive:** falsely annotating unrelated entities in one sentence as a certain relation type; (2) **False Negative:** neglecting related entities. These label noises distort the true label distribution of train corpora, creating a gap between the label distribution of train and test set (*i.e.*, shifted label distribution). With existing denoising methods, the effect of noisy training instances may be reduced; still, it would be infeasible to recover the original label, and thus label distribution shift remains an unsolved problem.

In our experiments, we notice that in distantly supervised datasets, instances labeled as NONE have a larger portion in test set than in train set. It is apparent that the strategy of rejecting "ambiguous" predictions would guide the model to predict more NONE types, leading the predicted label distribution towards a favorable direction. Specifically, in the train set of KBP, 74.25% instances are annotated as NONE and 85.67% in the test set. The original Bi-GRU model would annotate 75.72% instances to be NONE, which is close to 74.25%; after applying the max-threshold and entropy-threshold, this proportion becomes 86.18% and 88.30%, which are close to 85.67%.

Accordingly, we believe part of the underlying mechanism of heuristic threshold is to better handle the label distribution shift, and we try to further verify this hypothesis with experiments in next section.

4 Shifted Label Distribution

In this section, we will first summarize our observation on shifted label distribution, and then conduct empirical analysis to study its impact on model performance using synthetic datasets.

4.1 Shifted Label Distribution

Shifted label distribution refers to the problem that the label distribution of train set does not align with the test set. This problem is related to but different from "learning from imbalanced data", where the data distribution is imbalanced but consistent. Admittedly, one relation may appear more or less than another in natural language, creating distribution skews; however, this problem widely occurs in both supervised and distantly supervised settings, and is not our focus in this paper.

Our focus is the *label distribution difference between train and test set*. This problem is critical to distantly supervised relation extraction, where the train set is annotated with distant supervision and the test set is manually annotated. As previously mentioned in 3.2, distant supervision differs from human annotations by introducing false positive and false negative labels. The label distribution of train set is subject to existing entries in KBs, and thus there exists a gap between label distributions of train and test set.

We visualize the distribution of KBP (distantly-supervised dataset) and a truncated 6-class version of TACRED (human-annotated dataset) in Figure 1. Also, histograms on $|p(r_i|\mathcal{D}_{train}) - p(r_i|\mathcal{D}_{test})|$ of each relation r_i is shown. It is observed that KBP and NYT both have shifted label distribution; while TACRED has a relatively consistent label distribution.

4.2 Impact of Shifted Label Distribution

In order to quantitatively study the impact of label distribution shift, we construct synthetic datasets by sub-sampling instances from the human-annotated TACRED dataset. In this way, the only variable is the label distribution of synthetic datasets, and the impact of other factors such as label noise is excluded.

Specifically, we create five synthetic train sets, by sampling from original TACRED train set with label distributions S1-S5. S5 is a randomly generated label distribution (plotted in Fig 7 in Appendix B). S0 is TACRED's original train set label distribution. $\mathsf{S1} - \mathsf{S4}$ are created with linear interpolation between S0 and S5, *i.e.*, $\mathsf{Si} = \frac{5-i}{5}\mathsf{S0} + \frac{i}{5}\mathsf{S5}$. We ensure that the number of sampled instances in each train set is controlled within 10000 ± 3, and the label distribution of each set satisfies S1-S5 respectively.

We conduct experiments with three typical models (*i.e.*, Bi-GRU, Bi-LSTM and ReHession) and summarize the results in Fig 4. We observe that, from S1 to S5, the performance of all models consistently drops. This phenomenon verifies that shifted label distribution is making a negative influence on model performance. The negative effect expands as the train set label distribution becomes more twisted.

At the same time, we observe that feature-based ReHession is more robust to such shift. The gap between ReHession and Bi-GRU stably decreases, and eventually ReHession starts to outperform the other two at S4. This could be the reason accounting for "diminishing" phenomenon — Neural models such as Bi-GRU is supposed to outperform ReHession by a huge gap (as with S1); however on distantly supervised datasets, shifted label distribution seriously interfere the performance (as

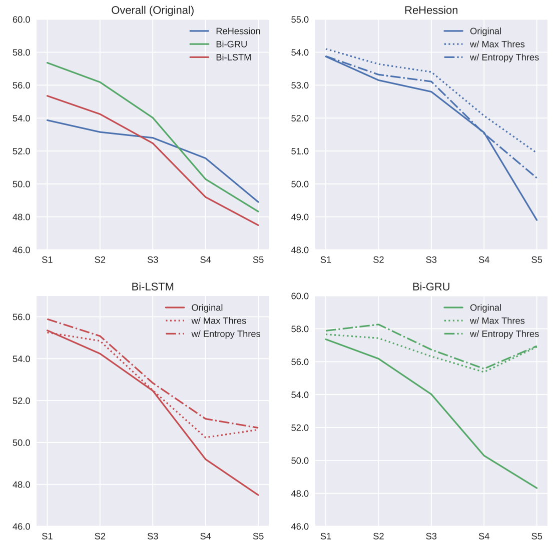

Figure 4: F1 scores on synthesized datasets S1-S5. We observe that (a) performance consistently drops from S1 to S5, demonstrating the impact of shifted label distributions; (b) ReHession is more robust to such distribution shift, outperforming Bi-LSTM and Bi-GRU on S4 and S5; (c) threshold is an effective way to handle such shift.

with S4 and S5), and therefore the performance gap diminishes.

We also applied the two aforementioned threshold techniques on the five synthetic datasets and summarize their performance in Fig 4. After applying threshold techniques, the three models become more robust to the label distribution shift. This observation verifies that the underlying mechanism of threshold techniques can help the model better handle label distribution shift.

5 Bias Adjustment: An Adaptation Method for Label Distribution Shift

Investigating the probabilistic nature of softmax classifier, we present a principled domain adaptation approach, *bias adjustment*, targeting label distribution shift. This approach explicitly fits the model along shift by adjusting the bias term in softmax classifier.

5.1 Bias Adjustment

We view corpora with different label distributions as different domains. Denoting distantly supervised corpus (train set) as $\mathcal{D}_d$ and human-annotated corpus (test set) as $\mathcal{D}_m$, our task becomes to calculate $p(y = r_i|\mathbf{h}, \mathcal{D}_m)$ based on $p(y = r_i|\mathbf{h}, \mathcal{D}_d)$.

We assume the only difference between $p(y = r_i|\mathbf{h}, \mathcal{D}_m)$ and $p(y = r_i|\mathbf{h}, \mathcal{D}_d)$ to be the label distribution, and the semantic meaning of each label is unchanged. Accordingly, we assume $p(\mathbf{h}|r_i)$ is universal, *i.e.*,

$$p(\mathbf{h}|r_i, \mathcal{D}_m) = p(\mathbf{h}|r_i, \mathcal{D}_d) = p(\mathbf{h}|r_i) \tag{2}$$

As distantly supervised relation extraction models are trained with $\mathcal{D}_d$, its prediction in Equation 1 can be viewed as $p(y = r_i|\mathbf{h}, \mathcal{D}_d)$, *i.e.*,

$$\begin{aligned} p(y = r_i|\mathbf{h}, \mathcal{D}_d) &= p(y = r_i|\mathbf{h}) \\ &= \frac{\exp(\mathbf{r}_i^T \mathbf{h} + b_i)}{\sum_{r_j} \exp(\mathbf{r}_j^T \mathbf{h} + b_j)} \end{aligned} \tag{3}$$

Based on the Bayes Theorem, we have:

$$p(y = r_i|\mathbf{h}, \mathcal{D}_m) = \frac{p(\mathbf{h}|r_i, \mathcal{D}_m) p(r_i|\mathcal{D}_m)}{\sum_{r_j} p(\mathbf{h}|r_j, \mathcal{D}_m) p(r_j|\mathcal{D}_m)} \tag{4}$$

Based on the definition of conditional probability and Equation 2, we have:

$$\begin{aligned} p(\mathbf{h}|r_i, \mathcal{D}_m) &= p(\mathbf{h}|r_i) = p(\mathbf{h}|r_i, \mathcal{D}_d) \\ &= \frac{p(\mathbf{h}, r_i, \mathcal{D}_d)}{p(r_i, \mathcal{D}_d)} \\ &= \frac{p(r_i|\mathbf{h}, \mathcal{D}_d) \cdot p(\mathbf{h}|\mathcal{D}_d)}{p(r_i|\mathcal{D}_d)} \end{aligned} \tag{5}$$

With Equation 3, 4 and 5, we can derive that

$$\begin{aligned} & p(y = r_i|\mathbf{h}, \mathcal{D}_m) \\ =& \frac{p(y = r_i|\mathbf{h}, \mathcal{D}_d) \dfrac{p(r_i|\mathcal{D}_m)}{p(r_i|\mathcal{D}_d)}}{\sum_{r_j} p(y = r_j|\mathbf{h}, \mathcal{D}_d) \dfrac{p(r_j|\mathcal{D}_m)}{p(r_j|\mathcal{D}_d)}} \\ =& \frac{\exp(\mathbf{r}_i^T \mathbf{h} + b'_i)}{\sum_j \exp(\mathbf{r}_j^T \mathbf{h} + b'_j)}. \end{aligned} \tag{6}$$

where

$$b'_i = b_i + \ln p(r_i|\mathcal{D}_m) - \ln p(r_i|\mathcal{D}_d) \tag{7}$$

With this derivation we now know that, under certain assumptions (Equation 2), we can adjust the prediction to fit a target label distribution given $p(r_i|\mathcal{D}_d)$ and $p(r_i|\mathcal{D}_m)$. Accordingly, we use Equation 6 and 7 to calculate the adjusted prediction as:

$$r^* = \arg\max_{r_i} \exp(\mathbf{r}_i^T \mathbf{h} + b'_i) \tag{8}$$

Label distribution estimation. As to the source domain (train) label distribution, $p(r_i|\mathcal{D}_d)$ can be easily estimated on train set. In our experiment,

Dataset	Model	Original	Max-thres	Δ	Ent-thres	Δ	BA-Set	Δ	BA-Fix	Δ
KBP	ReHession	36.07 ± 1.06	39.99 ± 1.09	3.92	40.21 ± 1.28	4.14	37.18 ± 1.59	1.11	37.86 ± 1.64	1.79
	Bi-GRU	37.77 ± 0.18	45.01 ± 1.61	7.24	43.14 ± 1.59	5.37	40.63 ± 1.79	2.86	42.28 ± 2.05	4.51
	Bi-LSTM	34.51 ± 0.99	40.11 ± 0.73	5.60	40.07 ± 1.61	5.56	37.43 ± 1.72	2.92	38.28 ± 1.66	3.77
	PA-LSTM	37.28 ± 0.81	44.04 ± 1.09	6.76	43.17 ± 1.25	5.89	40.47 ± 2.55	3.19	42.44 ± 1.32	5.16
	CNN	30.53 ± 2.26	31.23 ± 1.68	0.70	31.39 ± 2.43	0.86	30.73 ± 3.96	0.20	36.12 ± 2.65	5.59
	PCNN	31.58 ± 0.35	32.98 ± 0.89	1.40	32.19 ± 0.76	0.61	32.79 ± 1.95	1.21	38.78 ± 2.63	7.20
NYT	ReHession	46.79 ± 0.75	49.12 ± 0.47	2.33	49.14 ± 0.57	2.35	48.50 ± 1.23	1.71	48.14 ± 1.08	1.35
	Bi-GRU	47.88 ± 0.85	48.87 ± 1.40	0.99	48.98 ± 1.27	1.10	48.40 ± 1.52	0.52	48.74 ± 1.94	0.86
	Bi-LSTM	48.15 ± 0.87	49.43 ± 1.27	1.28	49.08 ± 0.88	0.93	49.72 ± 1.24	1.57	49.90 ± 1.44	1.75
	PA-LSTM	46.33 ± 0.64	46.70 ± 1.43	0.37	48.65 ± 1.24	2.32	47.77 ± 1.43	1.44	48.65 ± 1.24	2.32
	CNN	46.75 ± 2.79	47.87 ± 1.89	1.12	46.54 ± 2.70	-0.21	48.83 ± 1.66	2.08	48.17 ± 1.88	1.42
	PCNN	44.63 ± 2.70	48.31 ± 0.40	3.68	47.04 ± 2.60	2.41	49.08 ± 0.88	4.45	48.72 ± 1.72	4.09
TACRED	ReHession	58.06 ± 0.54	57.97 ± 0.57	-0.09	57.91 ± 0.52	-0.15	58.59 ± 0.66	0.53	58.61 ± 0.99	0.55
	Bi-GRU	65.38 ± 0.60	65.29 ± 0.58	0.01	65.38 ± 0.60	0.10	63.72 ± 0.64	-1.56	64.70 ± 0.46	-0.68
	Bi-LSTM	62.74 ± 0.23	62.95 ± 0.35	0.21	62.67 ± 0.29	-0.07	62.73 ± 0.60	-0.01	63.44 ± 0.54	0.70

Table 3: F1 score of RE Models with Threshold and Bias Adaptation. Five-time average and standard deviation are reported. Δ denotes the F1 improvement over original. On DS datasets, the four methods targeting label distribution shift achieve consistent performance improvement, with averagely 3.83 F1 improvement on KBP and 1.72 on NYT. However, the same four methods fail to improve performance on human-annotated TACRED.

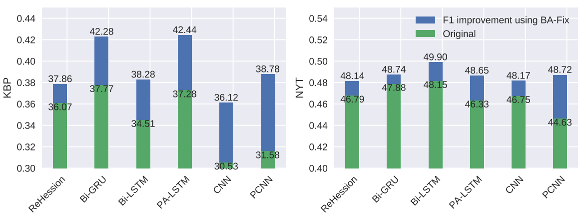

Figure 5: F1 Improvement using BA-Fix. BA-Fix consistently improves performance in compared models.

we use a simple way to estimate target domain distribution $p(r_i|\mathcal{D}_m)$, *i.e.*, using maximum likelihood estimation on a held-out *clean dev* set, which is a 20% sample from test set. This setting is similar to heuristic threshold techniques.

Implementation details. The bias adjustment model are implemented in two ways:

- **BA-Set** directly replaces the bias term in Equation 1 with b'_i in Equation 7 during evaluation. It does not require any modification to model training.

- **BA-Fix** fixes the bias term in Equation 1 as $b_i = \ln p(r_i|\mathcal{D}_d)$ during training and replaces it with $b'_i = \ln p(r_i|\mathcal{D}_m)$ during evaluation. Intuitively, BA-Fix would encourage the model to fit our assumption better (Equation 2); still, it needs special handling during model training, which is a minor disadvantage of BA-Fix compared with BA-Set.

5.2 Results and Discussion

We conduct experiments to explore the effectiveness of BA-Set and BA-Fix and summarize their performance in Table 3. We find that these two technologies bring consistent improvements to all methods on both distantly supervised datasets. Especially, in the case of PCNN on KBP dataset, a 23% relative F1 improvement is observed. At the same time, the same technology fails to achieve performance improvements on TACRED, the human annotated dataset. In other words, we found that, by explicitly adapt the model along the label distribution shift, consistent improvements can be achieved on distant supervision but not on human annotations. Therefore, this observation supports our assumption that shifted label distribution is an unique factor for distantly supervised RE that needs special handling.

Comparison with Heuristic Techniques. Comparing these two technologies (*i.e.*, BA-Set and BA-Fix) with heuristic techniques, we find bias adjustment technologies are more explainable but all four techniques achieve similar performance improvements. Specifically, we observe performance improvements on distant supervision are relative significant. But their performance gains are comparatively marginal on human annotations. This observation verifies our intuition that part of the underlying mechanism of these two heuristic techniques is to better the label distribution shift.

Relation Extraction Model Analysis. Our results can also shed insights on the relation extaction models. Noting that only bias terms in classifier are modified and only a small piece of extra information is used, it is implied that shifted label distribution is severely hindering model performance and may be limiting the power of neural models. Hidden representations $\mathbf{h}$ learned by neural models indeed capture semantic meanings more accurately than feature-based model, while the bias in classifier becomes the major obstacle towards better performance.

6 Comparison with Denoising Methods

In this section, we conduct analyses about shifted label distribution and its relation with label noise. Specifically, we apply a popular label noise reduction method–selective attention (Lin et al., 2016), which groups all sentences with the same entity pair into one bag, conducts multi-instance training and tries to place more weight on high-quality sentences within the bag. This method, along with threshold techniques and bias adjustment introduced in previous sections, is applied with two different models (*i.e.*, PCNN and Bi-GRU).

We summarize their improvements over the original model in Figure 6. We can find that selective attention indeed improves the performance; meanwhile, heuristic threshold and bias adaption approaches boost the performance, and in some cases the boost is even more significant than that of selective attention. This observation is reasonable since both heuristics and bias adaption approaches are able to access additional information of test set. Still, it is surprising that such small piece of information brings about huge difference, demonstrating the importance of handling shifted label distribution. It shows that there exists much space for improving distantly supervised RE models from a shifted label distribution perspective.

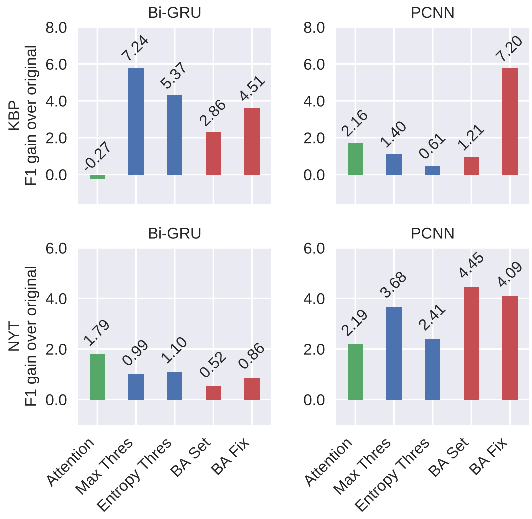

Figure 6: Comparison among selective attention, threshold heuristics and bias adaption approaches. Threshold heuristics and bias adaption approaches bring more significant improvements in some cases, indicating that shifted label distribution is a non-negligible problem.

7 Related Work

There exist two aspects of related work regarding the topic here, which are relation extraction and distant supervision.

7.1 Relation Extraction

Relation extraction is to identify the relationship between a pair of entity mentions within a sentence. Recent approaches follow the supervised learning paradigm and rely on considerable amounts of labeled instances to train effective models. Zeng et al. (2014) proposed using CNN for relation extraction, which could automatically capture features from texts. Zeng et al. (2015) further extended it with piecewise max-pooling, *i.e.*, splitting the sentence into three pieces with the object and subject entity, doing max-pooling over the three pieces separately, and finally concatenating the hidden representations. Lin et al. (2016) apply sentence selective attention for learning from multiple instances. This method orgainize sentences into bags and assign lower weights to those less relavant sentences in the bag. Zhang et al. (2017) combined attention mechanism with position information.

7.2 Distant Supervision

In supervised relation extraction paradigm, one longstanding bottleneck is the lack of large-scale labeled training data. In order to alleviate the dependency on human supervision, Mintz et al. (2009) proposed distant supervision, namely constructing large datasets automatically by heuristically aligning text to an existing Knowledge Base (*e.g.*, Freebase). Though this strategy lightens annotation burdens, distant supervision inevitably introduces label noises. Since the relation types are annotated merely according to entity mentions in the sentence, the local context could be annotated with labels that are not expressed in the sentence, In recent years, there are several existing methods being proposed to deal with the label noises: Riedel et al. (2010) use multi-instance single-label learning paradigm; Hoffmann et al. (2011); Surdeanu et al. (2012) propose multi-instance multi-label learning paradigm. Recently, with the advance of neural network techniques, deep learning methods (Zeng et al., 2015; Lin et al., 2016) are applied to distantly supervised datasets, with powerful automatic feature extraction and advanced label noised reducing techniques such as selective attention. Also, distantly supervised relation extraction is formulated into a reinforcement learning problem by (Feng et al., 2018).

8 Conclusion and Future Work

In this paper, we first present the observation of inconsistent performance when models are trained with human annotations and distant supervision in the task of relation extraction. It leads us to explore the underlying challenges for distantly supervised relation extraction. Relating two effective threshold techniques to label distribution, we reveal an important yet long-overlooked factor – *shifted label distribution*. The impact of this factor is further demonstrated with experiments on five synthetic datasets. We also consider this issue from a domain adaptation perspective, introducing a bias adjustment method to recognize and highlight label distribution shift. All of these findings support our argument that shifted label distribution can severely hinder model performance and should be handled properly in future research.

Based on these observations, we suggest that in addition to label noise, more attention be paid to the shifted label distribution in distantly supervised relation extraction research. We hope that the analysis presented will provide new insights into this long-overlooked factor and encourage future research of creating models robust to label distribution shift. We also hope that methods such as threshold techniques and bias adjustment become useful tools in future research.

References

Joe Ellis, Xuansong Li, Kira Griffitt, Stephanie Strassel, and Jonathan Wright. 2012. Linguistic resources for 2013 knowledge base population evaluations. In *TAC*.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*.

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: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, volume 1, pages 2124–2133.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *AAAI*, volume 12, pages 94–100.

Liyuan Liu, Xiang Ren, Qi Zhu, Shi Zhi, Huan Gui, Heng Ji, and Jiawei Han. 2017. Heterogeneous supervision for relation extraction: A representation learning approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 46–56.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Gábor Melis, Chris Dyer, and Phil Blunsom. 2017. On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024. International World Wide Web Conferences Steering Committee.

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. Springer.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465. Association for Computational Linguistics.

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

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

A Detailed Experiment Setup

For a fair and meaningful comparison, we use the same experimental setup in all experiments.

A.1 Details on Datasets

We select three popular relation extraction datasets as benchmarks. Specifically, two of them are distantly supervised and one is human-annotated.

KBP (Ling and Weld, 2012) uses Wikipedia articles annotated with Freebase entries as train set, and manually-annotated sentences from 2013 KBP slot filling assessment results (Ellis et al., 2012) as test set.

NYT (Riedel et al., 2010) contains New York Times news articles and has been already heuristically annotated. Test set is constructed manually by (Hoffmann et al., 2011).

TACRED (Zhang et al., 2017) is a large-scale crowd-sourced dataset, and is sufficiently larger than previous manually annotated datasets.

A.2 Model Details

We consider two popular classes of relation extraction methods here, *i.e.*, feature-based and neural models. For each relation mention, these models will first construct a representation $\mathbf{h}$, and then make predictions based on $\mathbf{h}$. [3]

$$p(y = r_i|\mathbf{h}) = \frac{\exp(\mathbf{r}_i^T \mathbf{h} + b_i)}{\sum_{r_j} \exp(\mathbf{r}_j^T \mathbf{h} + b_j)}$$

where $\mathbf{r}_i$ and b_i are the parameters corresponding to i-th relation type.

A.2.1 Feature-based model

We included three feature-based models, *i.e.*, CoType-RM (Ren et al., 2017), ReHession (Liu et al., 2017), and multi-class logistic regression. For each relation mention z, these methods would first extract a list of features, $\mathcal{F} = \{f_1, f_2, ..., f_m\}$. These features require additional resources like POS-taggers and brown clustering. Detailed description of these features are listed in Table 4.

CoType-RM is a variant of CoType (Ren et al., 2017), a unified learning framework to get both the feature embedding and the label embedding. It leverages a partial-label loss to handle the label noise, and uses cosine similarity to conduct inference. Here, we only use its relation extraction part.

ReHession (Liu et al., 2017) directly maps each feature to an embedding vector, treats their average as the relation mention representation $\mathbf{h}$, and uses a softmax to make predictions. This method was initially proposed for heterogeneous supervision, and is modified to fit our distantly supervised relation extraction task. Specifically, for a relation mention annotated with a set of relation $Y = \{r_{l_1}, \cdots, r_{l_m}\}$, it would first calculate a cross entropy as the loss function:

$$\mathcal{L} = -\sum_{r_i} q(y = r_i|Y, \mathbf{h}) \log p(y = r_i|\mathbf{h}) \quad (9)$$

where $p(.|\mathbf{h})$ is defined in Equation 1, and $q(.|Y, \mathbf{h})$ is used to encode supervision information in a self-adapted manner:

$$q(y = r_i|Y, \mathbf{h}) = \frac{\exp(\mathbf{r}_i^T \mathbf{h} + b_i) \cdot \mathbb{I}(r_i \in Y)}{\sum_{r_j \in Y} \exp(\mathbf{r}_j^T \mathbf{h} + b_j)}$$

We can find that when $|Y| = 1$ (only one label is assigned to the relation mention), $q(.|Y, \mathbf{h})$ would be one-hot and Equation 9 would become the classical cross entropy loss.

Logistic Regression is applied over the extracted features as a baseline method.[4]

A.2.2 Neural Models

We employed several popular neural structure to calculate the sentence representation $\mathbf{h}$. As to the objective function, we use Equation 9 for all the following neural models.

Bi-LSTMs and Bi-GRUs use Bidirectional RNNs to encode sentences and concatenate their final states in the last layer as the representation. Following previous work (Zhang et al., 2017), we use two types of RNNs, Bi-LSTMs and Bi-GRUs. Both of them have 2 layers with 200d hidden state in each layer.

Position-Aware LSTM computes sentence representation with an attention over the outputs of LSTMs. It treats the last hidden state as the query and integrates a position encoding to calculate the attention (Zhang et al., 2017).

CNNs and PCNNs use convolution neural networks as the encoder. In particular, CNN directly appends max-pooling after the convolution layer (Zeng et al., 2014); PCNN uses entities to split each sentence into three pieces, and does

[3] CoType and Logistic Regression are exceptions as they don't adopt softmax to generate output.

[4] We use liblinear package from https://github.com/cjlin1/liblinear

Feature Name	Description	Example
Brown cluster	Brown cluster ID for each token	"*BROWN_010011001*"
Part-of-speech (POS) tag	POS tags of tokens between two EMs	"*VBD*" , "*VBN*" , "*IN*"
Entity Mention Token	Tokens in each entity mention	"*TKN_EM1_Hussein*"
Entity mention (EM) head	Syntactic head token of each entity mention	"*HEAD_EM1_HUSSEIN*"
Entity mention order	whether EM 1 is before EM 2	"*EM1_BEFORE_EM2*"
Entity mention distance	number of tokens between the two EMs	"*EM_DISTANCE_3*"
Entity mention context	unigrams before and after each EM	"*EM1_AFTER_was*"
Tokens Between two EMs	each token between two EMs	"*was*" , "*born*" , "*in*"
Collocations	Bigrams in left/right 3-word window of each EM	"*Hussein was*" , "*in Amman*"

Table 4: Text features used in feature-based models. ("*Hussein*", "*Amman*", "*Hussein was born in Amman*") is used as an example.

max-pooling respectively on these three pieces after the convolution layer. Their outputs are concatenated as the final output (Zeng et al., 2015).

A.3 Model Training Details

We run each model for 5 times and report the average F1 and standard variation.

Optimization. We use Stochastic Gradient Descent (SGD) for all models. Learning rate is set at 1.0 initially, and is dynamically updated during training, *i.e.*, once the loss on the dev set has no improvement for 3 consecutive epochs, the learning rate will be factored by 0.1.

Hyper-parameters. For ReHession, dropout is applied on input features and after average pooling. We tried the two dropout rates in $\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

For Position Aware LSTM, Bi-LSTM and Bi-GRU, dropout (Srivastava et al., 2014) is applied at the input of the RNN, between RNN layers and after RNN before linear layer. Following (Melis et al., 2017) we tried input and output dropout probability in $\{0.4, 0.5, 0.6, 0.7, 0.8\}$, and intra-layer dropout probability in $\{0.1, 0.2, 0.3, 0.4\}$. We consider them as three separate hyper-parameters and tune them greedily. Following previous work (Zhang et al., 2017), dimension of hidden states are set at 200.

Following previous work (Lin et al., 2016), the number of kernels is set to 230 for CNNs or PCNNs, and the window size is set at 3. Dropout is applied after pooling and $\tanh$ activation . We tried the dropout rates in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$.

B Additional Figures and Tables

Figure 7 shows the full label distribution of TACRED dataset, and the simulated TACRED S5 dataset with a shifted distribution.

Figure 8 shows the full label distribution of NYT dataset. NYT is constructed with distant supervision and has a shifted distribution.

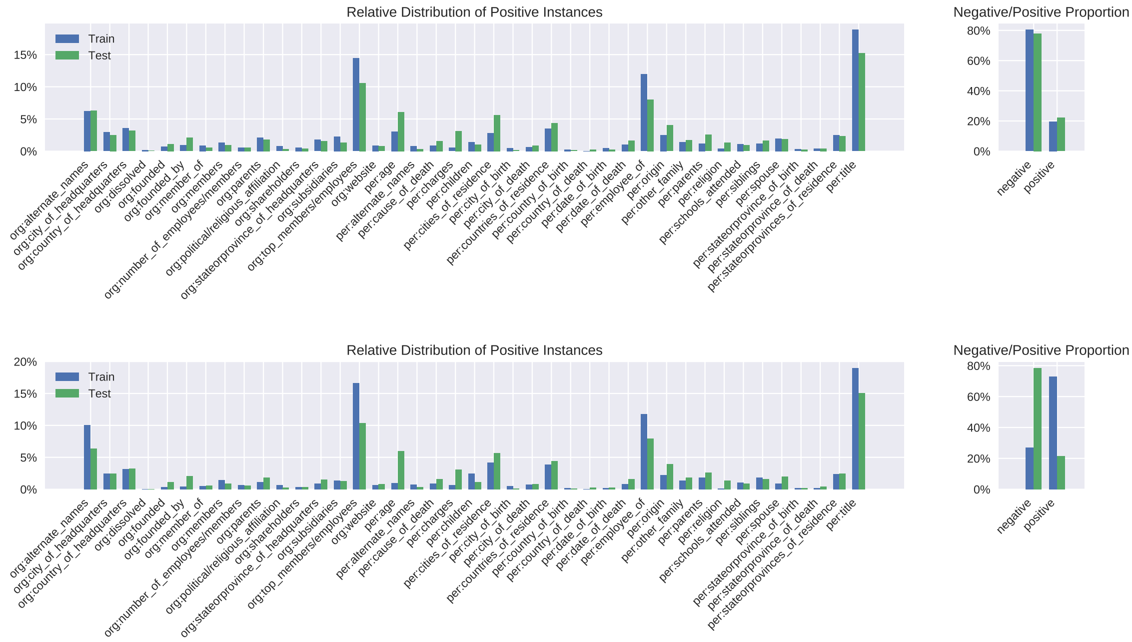

Figure 7: Top: Label distribution of original TACRED; **Bottom:** Using a randomly generated distribution for S5 train set, and keeping original test set. Label distribution of other synthesized datasets (S1-S4) are generated with linear interpolation of these two train set distributions.

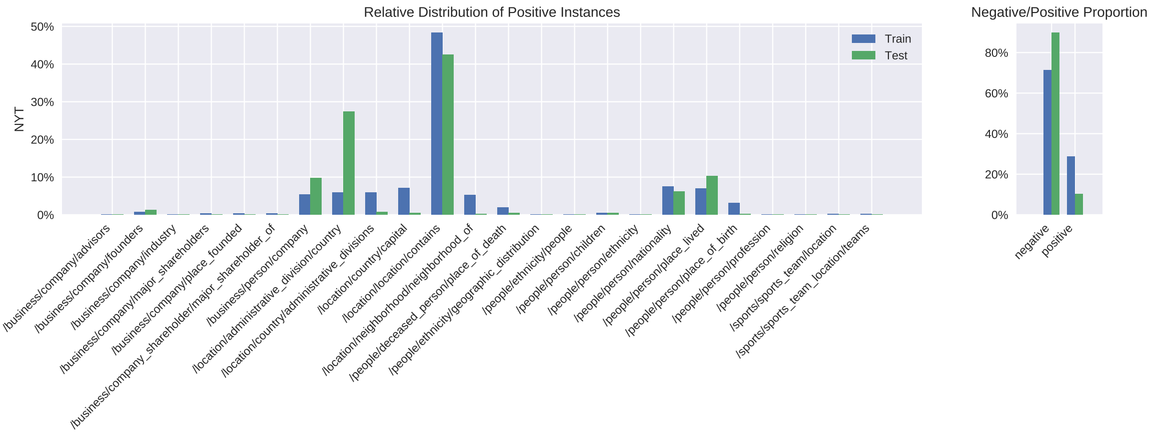

Figure 8: Label Distribution of original NYT. Similar to KBP, the distribution is shifted as well.