Distilling Knowledge from Well-informed Soft Labels for Neural Relation Extraction

Zhenyu Zhang, Xiaobo Shu, Bowen Yu, Tingwen Liu*, Jiapeng Zhao, Quangang Li, Li Guo
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{zhangzhenyu1996,shuxiaobo,yubowen,liutingwen,zhaojiapeng,liquangang,guoli}@iie.ac.cn

Abstract

Extracting relations from plain text is an important task with wide application. Most existing methods formulate it as a supervised problem and utilize one-hot hard labels as the sole target in training, neglecting the rich semantic information among relations. In this paper, we aim to explore the supervision with soft labels in relation extraction, which makes it possible to integrate prior knowledge. Specifically, a bipartite graph is first devised to discover type constraints between entities and relations based on the entire corpus. Then, we combine such type constraints with neural networks to achieve a knowledgeable model. Furthermore, this model is regarded as teacher to generate well-informed soft labels and guide the optimization of a student network via knowledge distillation. Besides, a multi-aspect attention mechanism is introduced to help student mine latent information from text. In this way, the enhanced student inherits the dark knowledge (e.g., type constraints and relevance among relations) from teacher, and directly serves the testing scenarios without any extra constraints. We conduct extensive experiments on the TACRED and SemEval datasets, the experimental results justify the effectiveness of our approach.

Introduction

Relation extraction (RE), defined as the task of detecting semantic relations among two entities in a sentence, is a key component of many natural language processing (NLP) applications, such as knowledge base population (Zhang et al. 2018; Distiawan et al. 2019) and question answering (Deng et al. 2019; Mitra et al. 2019).

Most existing work solves the RE task by fitting the outputs of a model to *hard labels* (i.e, one-hot vectors) (Zhang et al. 2017; Zhang, Qi, and Manning 2018; Guo, Zhang, and Lu 2019), regardless of the rich semantic correlations among relations. For example, the relation *cities of residence* is often considered similar to *city of birth* rather than *founded by*, yet such information can not be carried by hard labels (Lopez-Paz et al. 2016). With this in mind, using *soft labels* (i.e, probability distributions containing the relevance over relations) as additional supervision for RE is a natural choose to leverage the rich correlated information among relations. In fact, exploring the supervision with soft labels has been concerned in many machine learning areas, especially computer vision (Park et al. 2019; Yang et al. 2019), these successful prior work have confirmed that *soft labels are more informative than hard labels*. But to the best of our knowledge, there are few references to explore *the supervision with soft labels in the RE task*.

As suggested in (Hinton, Vinyals, and Dean 2015; Yim et al. 2017), knowledge distillation is an effective way to explore and incorporate soft labels, which involves a teacher network to provide soft training signals for a student network. However, the performance of teacher typically determines the upper bound of student (Mishra and Marr 2018; Clark et al. 2019). In light of this, it is necessary to train a prominent teacher by introducing extra knowledge to generate better-informed soft labels. Inspired by previous work which improves RE with type constraints (Vashishth et al. 2018; Lei et al. 2018), we decide to enhance the teacher network with such type knowledge. To be specific, every relation puts some constraints on the type of subject and object entities, and vice versa (Koch et al. 2014). For instance, the relation *cities of residence* can only occur between a *person* and a *city*. Nevertheless, most of prevalent approaches acquire such constraints from knowledge base (e.g. Freebase), making it hard to obtain appropriate rules and apply in real testing scenarios, because the target relations may not be found in existing knowledge bases in some cases. For example, all relations in the SemEval dataset cannot be retrieved from existing knowledge base (Hendrickx et al. 2010). To overcome this limitation, in this paper we acquire type constraints directly from the corpus.

Methodologically, we explore soft labels to improve RE from two perspectives: excavate type constraints from the entire corpus to acquire *soft rules* (the *global* perspective), and combine soft rules with the teacher network to generate *well-informed soft labels* for each instance (the *local* perspective). For a sentence with a relational fact {*subject, relation, object*}, we combine the entity type of *subject* and *object* to achieve a pattern. To exploit the type constraints with a global point of view, we first count the co-occurrence number of patterns and relations across the corpus, and then nor-

*Corresponding author: Tingwen Liu (liutingwen@iie.ac.cn)

malize the global statistics such that each pattern has a valid probability distribution over all relations. Finally, a pattern-to-relation bipartite graph is constructed, with one node set is patterns and another is relations. In this way, the weight between pattern p_i and relation r_j can be formulated as the probability that the relation of one sentence with pattern p_i is r_j, and the distribution among all relations of one pattern can be further regarded as the *soft rule* of mapping entity to relation. To capture the relevance among relations locally, we incorporate knowledge distillation into training procedure. Specifically, we first combine soft rules with a Graph Convolution Networks (GCN) model to achieve three versions of knowledgeable teachers with different fusion methods, then employ the teacher to generate *well-informed soft labels* and guide the optimization of a student. In addition, we present a multi-aspect attention (MAA) mechanism for the student network to mine some latent knowledge in the text by imitating the global branch of teacher. By this means, the student network inherits all the knowledge from teacher and requires no external information any more.

In summary, our contributions are as follows:

- In this paper, for the first time, we propose to supervise RE with soft labels, which is capable of capturing more dark knowledge than one-hot hard labels.
- By distilling the knowledge in well-informed soft labels which contain type constraints and relevance among relations, we free the testing scenarios from a heavy reliance on external knowledge.
- The extensive experiments on two public datasets justify the effectiveness of our approach. The source code can be obtained from https://github.com/zzysay/KD4NRE.

Preliminaries

In this section, we first describe how we collect and arrange global type constraints to obtain *soft rules*, then briefly recall some basics of knowledge distillation.

Type Constraints

In RE task, relations have expected types for each argument. Entity types, whether coarse-grained (e.g., from NER tags) or fine-grained (e.g., from Freebase), are important knowledge for making decisions (Koch et al. 2014). When we focus on individual instance, the hard constraint of entity type could help us to filter some irrelevant relations, while the rest relations are treated equally. But in fact, when we zoom out to consider the entire corpus, and count the co-occurrence number of entity types and relations, we will have a more comprehensive view of type constraints: the constraints between entity types and relations can then be represented by its co-occurrence number with relations.

To organize the type constraints between entities and relations globally and structurally, we construct a bipartite graph $\mathcal{G}$. Specifically, for each entity pair (s, o) in the sentence, we combine their types to achieve a *pattern* p. From this step, we obtain the pattern set $\mathcal{P} = \{p_i\}$ and formulate a support set $\mathcal{S}(p_i)$ for each p_i. The support set of a pattern contains all entity pairs corresponding to this pattern. In addition, we also get a set of target relations $\mathcal{R} = \{r_j\}$, and the support set $\mathcal{S}(r_j)$ denoting the set of entity pairs having relation r_j. The co-occurrence number of pattern p_i and relation r_j is defined as $w_{ij} = |\mathcal{S}(p_i) \cap \mathcal{S}(r_j)|$. In other word, every relational fact (s, r_j, o) with pattern p_i is counted as a co-occurrence of p_i and r_j.

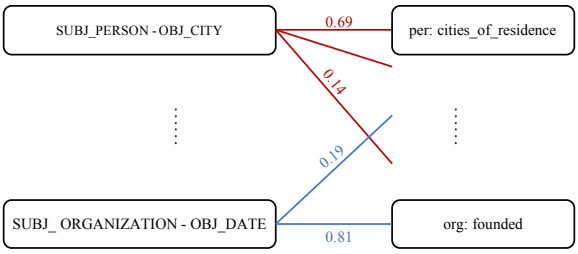

Figure 1: Toy illustration of our bipartite graph, in which left nodes are patterns, right nodes are relations, and edges are weighted by normalized co-occurrence statistics.

However, it is quite difficult to directly utilize the raw co-occurrence counts as soft rules, since these counts have a heavily skewed distribution that spans several orders of magnitude. Following (Su et al. 2018b), for each pattern we normalize its co-occurrence counts to form a valid probability distribution over relations. In the end, the bipartite pattern-to-relation graph $\mathcal{G}$ is constructed, with one node set being the patterns, the other node set being the relations, and the weighted edges $\bar{w}_{ij} = P(r_j|p_i) = w_{ij}/\sum_{j'} w_{ij'}$ representing the normalized global co-occurrence statistics. Figure 1 shows an example for clarity.

Knowledge Distillation

knowledge distillation is an effective framework to transfer knowledge from a neural network to another, which typically consists of two branches: a teacher T, which is usually a complex model or accompanied by some extra knowledge, and a student S, which is a small network that learns from the teacher (Hinton, Vinyals, and Dean 2015). In standard knowledge distillation model, the teacher network T is trained with ground-truth (i.e., hard labels) and outputs soft labels $\tilde{P}_T = \mathit{softmax}(\tilde{Z}_T/\tau)$, in which $\tilde{Z}_T$ is the pre-softmax logits and τ is the temperature parameter that is normally set to 1. Similarly, one can define $\tilde{P}_S = \mathit{softmax}(\tilde{Z}_S/\tau)$ for the student network S. In the training stage, S is required to match not only the ground-truth one-hot labels, but also the probability outputs of the teacher model:

$$\mathcal{L}_S = (1-\lambda)\mathcal{L}_{GT}^S + \lambda\mathcal{L}_{KD} \tag{1}$$

where $\mathcal{L}_{GT}^S$ is the ground-truth loss using one-hot labels, $\mathcal{L}_{KD}$ is the knowledge distillation loss using teacher's soft labels and λ is the coefficient to trade off such two terms. Typically, $\mathcal{L}_{GT}$ is often the cross entropy loss in classification problems, and $\mathcal{L}_{KD}$ is the Kullback-Leibler divergence to quantify the difference of output distribution from student to teacher:

$$\mathcal{L}_{GT}^S = \mathit{CE}(\tilde{G}, \tilde{P}_S) = -\sum\nolimits_i \tilde{G}(i)\log\tilde{P}_S(i) \tag{2}$$

$$\mathcal{L}_{KD} = \mathit{KL}(\tilde{P}_T||\tilde{P}_S) = \sum\nolimits_i \tilde{P}_T(i)\log(\tilde{P}_T(i)/\tilde{P}_S(i)) \tag{3}$$

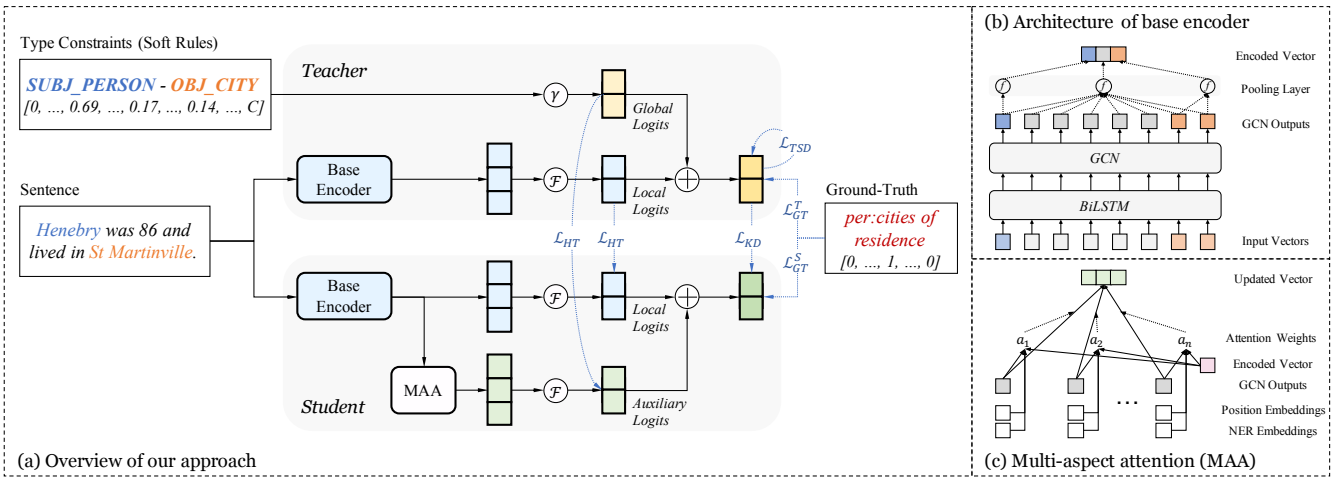

Figure 2: Plate representation of our proposed model. (a): The overview of our approach, in which the blue dashed lines indicate loss functions. (b): The architecture of base encoder, in which the dependency tree of sentence is transformed into an undirected graph. (c): The diagram of multi-aspect attention, it utilizes information from positions, NER tags, entities and sentences.

where $\tilde{G}$ is the one-hot distribution of ground-truth and $\tilde{G}(i)$ is the i-th element of $\tilde{G}$. The soft distribution $\tilde{P}_T$ outputted by teacher model is more smooth by assigning non-zero probabilities to more than one class and yields smaller variance in gradients, which contain hidden information (also known as dark knowledge) about the relationship between different classes. By learning from soft labels, the student network inherits such dark knowledge and often has a faster convergence speed (Chen et al. 2017).

Methodology

Figure 2(a) indicates the architecture of our approach. We divide our approach into three components as follows:

- **The Base Encoder** captures both sequence-based and dependency-based information to encode raw text. In our model, it is the basic module of the teacher and student networks to learn sentence representations.
- **The Teacher Network** aims to combine prior soft type constraints with neural networks to achieve knowledgeable teacher, and there are three different fusion methods.
- **The Student Network** can only access to raw text, and a multi-aspect attention (MAA) mechanism is introduced to imitate the global branch of teacher network. During training, this network is forced to produce vectorized outputs that are similar to the outputs of teacher network.

Theoretically, each component can be instantiated with any learning structure in the deep learning literature. We present our particular implementations in the following.

The Base Encoder

Following Zhang, Qi, and Manning (2018), we implement our base encoder with a GCN-based model, its structure is presented in Figure 2(b). It is worth mentioning that the base encoder can easily use other RE methods which may be left as future work.

Let $\mathcal{X} = [x_1, ..., x_n]$ denote a sentence, where x_i is the i-th token. A subject entity s and an object entity o correspond to two spans in the sentence: $\mathcal{X}_s = [x_{s_1}, ..., x_{s_n}]$ and $\mathcal{X}_o = [x_{o_1}, ..., x_{o_n}]$. Given $\mathcal{X}$, $\mathcal{X}_s$, and $\mathcal{X}_o$, the goal of base encoder is to build a sentence representation for the RE task. For each word x_i, we transform it to a vector $\mathbf{x}_i \in \mathbb{R}^{d_w}$ using a word embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d_w}$, where $|\mathcal{V}|$ is the vocabulary and d_w is the dimension of word embeddings.

Firstly, a BiLSTM layer is adopted to capture the contextual information for each word. We denote all the input vectors as $\mathbf{X} = [\mathbf{x}_1; ...; \mathbf{x}_n]$. The contextualized word representation is obtained as follows:

$$\mathbf{H}_{lstm} = \text{BiLSTM}(\mathbf{X}) \tag{4}$$

To further capture the dependency structure over input sentence, we adopt the graph convolution operation to model dependency trees by converting each tree into an undirected graph with an adjacency matrix $\mathbf{A}$, where $A_{ij} = A_{ji} = 1$ if there is a dependency edge between tokens x_i and x_j:

$$\mathbf{H}_{gcn} = \text{GCN}(\mathbf{H}_{lstm}, \mathbf{A}) \tag{5}$$

Note that we add self-loops to each node in the graph, so that the information of corresponding node in the former layer will carry over to the later one directly. Next, the sentence representation is expressed as follows:

$$\mathbf{h}_{sent} = f(\mathbf{H}_{gcn}) \tag{6}$$

Here $f: \mathbb{R}^{d_h \times n} \rightarrow \mathbb{R}^{d_h}$ is a max pooling function that maps n hidden vectors to a sentence vector, where d_h is the dimension of hidden states. We also obtain subject representation $\mathbf{h}_s$ from $\mathbf{H}_{gcn}$: $\mathbf{h}_s = f([\mathbf{h}^{gcn}_{s_1}; ...; \mathbf{h}^{gcn}_{s_n}])$, as well as object representation $\mathbf{h}_o$ similarly.

Finally, we concatenate the sentence and entity representations as the output vector of our base encoder, which can be used for RE directly with a linear layer followed by softmax function:

$$\mathbf{h}_{base} = [\mathbf{h}_{sent}, \mathbf{h}_s, \mathbf{h}_o] \tag{7}$$

The Teacher Network

The teacher network focuses on improving the performance of RE with soft rules yield from the bipartite graph $\mathcal{G}$, which is constructed in the section of type constraints.

In the training stage, for the input sentence $\mathcal{X}$ with specified subject and object entities, we first obtain its pattern p using the type of these two entities, and further retrieve the corresponding global probability distribution $\mathcal{G}(p) \in \mathbb{R}^{d_r}$ from $\mathcal{G}$, where d_r is the number of relations. Note that the probability of NA (i.e., no relation) in $\mathcal{G}(p)$ is always 0, which means that no pattern will point to NA. To eliminate this limitation, we set $\mathcal{G}(p)$[NA] to a constant C. In this way, then $\mathcal{G}(p)$ can be viewed as *global* knowledge, while the output of base encoder can be seen as *local* knowledge. To acquire a knowledgeable teacher with global awareness, we combine them in a regulable manner:

$$\mathbf{h}_{tea} = \mathcal{F}(\mathbf{h}_{base}) + \gamma \cdot \mathcal{G}(p) \tag{8}$$

where $\mathcal{F}$: $\mathbb{R}^{d_h} \rightarrow \mathbb{R}^{d_r}$ is a linear function and d_r is the number of relations. γ is the weight of global probability, which has three different forms of implementation: γ_s, γ_v and γ_r:

- $\gamma_s \in \mathbb{R}$ is a fixed *scalar*, which is the most straightforward method to unite the local and global knowledge, although some manual adjustments are needed.
- $\gamma_v \in \mathbb{R}^{d_r}$ is a trainable *vector*. It gives different weight to different dimension of $\mathcal{G}(p)$ and is optimized via gradient back-propagation automatically.
- $\gamma_r \in \mathbb{R}^{d_r}$ is a trainable *relation-related vector* corresponding to current gold relation label r, $\gamma_r = \mathbf{\Gamma}[r]$ and $\mathbf{\Gamma} \in \mathbb{R}^{d_r \times d_r}$ is a relation embedding matrix. Unlike the above two label-independent methods, each relation has its unique weight vector, since $\mathbf{\Gamma}$ fully considers the distinctions between relations.

In that case, the teacher network has three different versions according to the different implementations of γ, we denote them as Teacher-*S*, Teacher-*V* and Teacher-*R*, respectively. Although Teacher-*R* relies on gold relation labels and is not suitable for testing scenarios, it is still an ingenious way to generate well-informed soft labels.

The Student Network

Recall that the mission of teacher is to generate soft labels, while student serves testing scenarios where extra knowledge and annotations are missing. Therefore, we hope student could tap the deep potential of raw text with the guidance of knowledgeable teacher. As shown in Figure 2(c), we present a multi-aspect attention (MAA) mechanism for the student network, which evaluates relative contribution of each word with the consideration of multiple aspects and further generates updated sentence representation.

Here, we define four aspects to measure the importance of each word. Following Zhang et al. (2017), we obtain *position* embeddings $\mathbf{p}_i = [\mathbf{p}_i^s; \mathbf{p}_i^o]$ for each word x_i using a shared position embedding matrix $\overline{\mathbf{P}}$, based on the relative distances from x_i to subject and object entities. Besides, named entity recognition (NER) tag is also a very useful shallow grammatical information, which can be treated as coarse-grained entity type. We obtain *NER tag* embedding $\mathbf{ner}_i$ for x_i using a NER tag embedding matrix $\overline{\mathbf{N}}$. In addition, we define a summary vector $\mathbf{q} = \mathbf{h}_{base}$, which encodes information about the entire *sentence* and *two entities*. Finally, we use above information from different four angles to develop the multi-aspect attention and further acquire the updated sentence representation:

$$\mathbf{h}_{maa} = \mathcal{MAA}(\mathbf{H}^{gcn}, \mathbf{P}, \mathbf{NER}, \mathbf{q}) \tag{9}$$

where $\mathbf{P} = [\mathbf{p}_1; ...; \mathbf{p}_n]$ and $\mathbf{NER} = [\mathbf{ner}_1; ...; \mathbf{ner}_n]$ are the embedding sequences of position and NER tag respectively. The function $\mathcal{MAA}(\cdot)$ refers to the multi-aspect attention, which can be formulated as follows:

$$u_i = \mathbf{v}^\top \tanh(\mathbf{W}_h \mathbf{h}_i^{gcn} + \mathbf{W}_p \mathbf{p}_i + \mathbf{W}_n \mathbf{ner}_i + \mathbf{W}_q \mathbf{q}) \tag{10}$$

$$a_i = \frac{\exp(u_i)}{\sum_{j=1}^{n} \exp(u_j)} \tag{11}$$

$$\mathbf{h}_{maa} = \sum\nolimits_{i=1}^{n} a_i \mathbf{h}_i^{gcn} \tag{12}$$

Here $\mathbf{W}_h \in \mathbb{R}^{d_h \times d_h}$,$\mathbf{W}_p \in \mathbb{R}^{d_h \times 2d_p}$,$\mathbf{W}_n \in \mathbb{R}^{d_h \times d_n}$,$\mathbf{W}_q \in \mathbb{R}^{d_h \times d_q}$ and $\mathbf{v} \in \mathbb{R}^{d_h}$ are learnable parameters, where d_p and d_n are the dimensions of position and NER tag embeddings respectively. Additional parameters of the network include embedding matrices $\overline{\mathbf{P}} \in \mathbb{R}^{(2l-1) \times d_p}$ and $\overline{\mathbf{N}} \in \mathbb{R}^{|\mathcal{N}| \times d_n}$, where l is the maximum sentence length and $\mathcal{N}$ is the set of NER tags generated from the Stanford CoreNLP toolkit (Manning et al. 2014). These two embedding matrices are initialized randomly.

Finally, $\mathbf{h}_{maa}$ is fed into a linear layer and integrated with the original logits:

$$\mathbf{h}_{stu} = \mathcal{F}(\mathbf{h}_{base}) + \mathcal{F}(\mathbf{h}_{maa}) \tag{13}$$

We name $\mathcal{F}(\mathbf{h}_{maa})$ as auxiliary logits, it is designed to imitate the global logits $\gamma \cdot \mathcal{G}(p)$ of the teacher network in the training stage. In other words, we hope that MAA has the ability to capture some dark knowledge related to the type constraints between entities and relations.

Objective Functions

This subsection illustrates the objective functions of teacher and student networks, we introduce two additional loss functions to help our model efficiently transfer dark knowledge (i.e., the global type constraints and the relevance among relations) from teacher to student.

The Teacher Network In vanilla knowledge distillation, the teacher network is trained to fit one-hot labels. However, the ultimate goal of teacher is to provide better guidances for student, rather than achieve high accuracy simply. Inspired by Yang et al. (2019), we introduce a top score difference (TSD) loss to make the teacher's distribution softer. More concretely, we first pick up K classes with the highest confidence scores from the teacher's output, and then compute the gap between the confidence scores of the primary class and other K-1 classes:

$$\mathcal{L}_{TSD} = \rho_1 - \frac{1}{k-1} \sum\nolimits_{k=2}^{K} \rho_k \tag{14}$$

where ρ_k refers to the value of k-th largest element in the output distribution of teacher. Based on the global statistics, K is set to 3 empirically. We add the penalty term to standard ground-truth loss $\mathcal{L}_{GT}^{T}$ when training the teacher, facilitating it to distribute confidence to a few secondary relations:

$$\mathcal{L}_T = \mathcal{L}_{GT}^{T} + \mathcal{L}_{TSD} \tag{15}$$

The Student Network Typically, knowledge distillation transfers dark knowledge from the final output of teacher. Chen et al. (2017) demonstrate that using the intermediate representation of teacher as hint can stabilize the training process and improve the final performance of student. Here, we utilize the Kullback-Leibler divergence to measure the differences of corresponding branches between the teacher and student networks as hint learning loss:

$$\mathcal{L}_{HT} = KL(L_L^T||L_L^S) + KL(L_G^T||L_A^S) \tag{16}$$

where L_L^T and L_G^T are the local and global logits of the teacher network respectively, L_L^S and L_A^S are the local and auxiliary logits of the student network respectively. Intuitively, it encourages the results of MAA to be similar with the scaled global logits of the teacher network. The loss of knowledge distillation is calculated as the sum of $\mathcal{L}_{HT}$ and $\mathcal{L}_{KD}$ with a weight factor λ_{ht}. As a result, the updated loss of student network is defined as follows:

$$\mathcal{L}_S = (1 - \lambda_{kd})\mathcal{L}_{GT}^{S} + \lambda_{kd}\hat{\mathcal{L}}_{KD} \tag{17}$$

$$\hat{\mathcal{L}}_{KD} = \mathcal{L}_{KD} + \lambda_{ht}\mathcal{L}_{HT} \tag{18}$$

Experiments

Experimental Settings

Datasets We conduct experiments on two widely used benchmark datasets: (1) **TACRED** (Zhang et al. 2017): It is the currently largest benchmark dataset for supervised RE, which contains 41 relations and a specially *no relation* class. Mentions in TACRED are typed, in which subject entities falls into 2 categories, and object entities are categorized into 16 types. We report micro-averaged Precision, Recall and F1 scores on this dataset as is conventional. (2) **SemEval** (Hendrickx et al. 2010): The SemEval (i.e., SemEval 2010 task 8) dataset contains 18 directed relations and a *no relation* class, all relations in this dataset cannot be retrieved from existing knowledge base. On SemEval, we follow the convention and report the macro-averaged F1 scores.

Implementation Details We tune all hyper-parameters according to the results on dev sets. For the base encoder, we use the same configure with C-GCN (Zhang, Qi, and Manning 2018). Beyond that, we set the NA probability C to 0.2, the temperature of knowledge distillation τ to 1, the weight factor of hint learning λ_{ht} to 1.8, the weight factor of type constraints in Teacher-*S* γ_s to 0.8. The size of position embedding d_p and NER tag embedding d_n in MAA are both set to 30. Inspired by Clark et al. (2019), we adopt the teacher annealing strategy: Let λ_{kd} increases from 0 to 1 linearly throughout the training stage of student. GloVe (Pennington, Socher, and Manning 2014) vectors are used as the initialization for word embeddings. Following Zhang et al. (2017) and Zhang, Qi, and Manning (2018), we augment the input with part-of-speech (POS) and named entity recognition (NER) embeddings, which are initialized randomly. For the SemEval dataset, we use Stanford CoreNLP toolkit to generate dependency parse trees, POS and NER annotations.

Dataset	#Train	#Dev	#Test	#Relation
TACRED	68,124	22,631	15,509	42
SemEval	7,500	500	2,717	19

Table 1: Statisticses of the TACRED and SemEval datasets.

Baselines We compare our model with the following baseline models: (1) **SDP-LSTM** (Xu et al. 2015): It applies a neural sequence model along the shortest dependency path between target entities. (2) **PA-LSTM** (Zhang et al. 2017): It employs a position-aware attention mechanism over LSTM outputs, and outperforms several CNN-based models. (3) **C-GCN** (Zhang, Qi, and Manning 2018): It applies a combination of pruning strategy and graph convolutions to the dependency tree, which is the base encoder of our model. (4) **SA-LSTM** (Yu et al. 2019): It adopts a segment attention mechanism on top of the LSTM, and is capable of learning relational expressions. (5) **ERNIE** (Zhang et al. 2019): It is a pre-trained language model with rich knowledge information, and outperforms BERT in RE task. (6) **AG-GCN** (Guo, Zhang, and Lu 2019): It utilizes an attention guided graph convolutional networks, which is the recent state-of-the-art on the TACRED dataset.

Experimental Results

Table 2 and 3 summarize the comparison results on the two datasets. We utilize Student-*S*/*V*/*R* to designate three different versions of student network with Teacher-*S*/*V*/*R* respectively. Note that Teacher-*R* is not suitable for testing scenarios, we do not report the results in these two tables.

Performance with Type Constraints Firstly, we focus on the performance of teacher. From the results in Table 2 and 3, we observe consistent performance gains when comparing our teacher networks with base encoder (C-GCN), which demonstrates the effectiveness of introducing soft type constraints. In particular, by using scalar weight γ_s, Teacher-*S* achieves significant improvements (+1.5% on F1 for TACRED and +0.6% on F1 for SemEval). When we replace γ_s with γ_v, Teacher-*V* further outperforms the AG-GCN model and achieves a new state-of-the-art. This is sensible, since the values in soft rules and local logits are not in the same order of magnitude, γ_v scales the different dimensions of soft rules to different degrees adaptively. Furthermore, for models without γ_s, the performance fluctuates only slightly when the NA probability C varies between $(0, 1]$, which again confirms that γ with vector form can realize the autoregulation of soft rules.

Additionally, it is worth mentioning that our Teacher-*R* achieves 83.5% and 94.8% F1 in the dev set of TACRED and SemEval respectively. Although greatly promoting the performance, it leaks the target relation label and thus cannot be applied to the testing stage.

Model	P	R	F1
SDP-LSTM (Xu et al. 2015)	66.3	52.7	58.7
PA-LSTM (Zhang et al. 2017)	65.7	64.5	65.1
C-GCN (Zhang, Qi, and Manning 2018)†	69.9	63.3	66.4
SA-LSTM (Yu et al. 2019)	69.0	66.2	67.6
ERNIE (Zhang et al. 2019)	70.0	66.1	68.0
AG-GCN (Guo, Zhang, and Lu 2019)	**73.1**	64.2	68.2
Teacher-*S* (ours)	69.5	66.3	67.9
Teacher-*V* (ours)	71.6	66.0	68.7⋆
Student-*S* (ours)	68.5	67.6	68.1
Student-*V* (ours)	71.1	66.7	68.8
Student-*R* (ours)	71.4	**67.9**	**69.6**⋆

Table 2: Results on the TACRED dataset, bold marks highest number among all models. † marks the base encoder of our model. ⋆ marks statistically significant improvements over AG-GCN with $p < 0.01$ under a bootstrap test.

Model	F1
SDP-LSTM (Xu et al. 2015)	83.7
PA-LSTM (Zhang et al. 2017)	82.7
C-GCN (Zhang, Qi, and Manning 2018)†	84.8
AG-GCN (Guo, Zhang, and Lu 2019)	85.7
Teacher-*S* (ours)	85.4
Teacher-*V* (ours)	85.9⋆
Student-*S* (ours)	85.5
Student-*V* (ours)	85.9
Student-*R* (ours)	**86.8**⋆

Table 3: Results on the SemEval dataset, bold marks highest number among all models. † marks the base encoder of our model. ⋆ marks statistically significant improvements over AG-GCN with $p < 0.05$ under a bootstrap test.

Performance with Knowledge Distillation Next, we investigate the performance of our student network, which is hoped to inherit both local and global knowledge from the teacher network by knowledge distillation. From Table 2, we can see that: (1) Student-*R* outperforms other models in all settings, which justifies the philosophy for choosing well-informed soft labels as additional supervision for RE. (2) A better teacher can educate a better student, which is a respond to the point we mentioned earlier: incorporating additional knowledge when training teacher is an efficient approach to improve the performance of student. (3) Student-*S* and Student-*V* outperform Teacher-*S* and Teacher-*V* slightly, indicating that knowledge in soft labels has been passed on from the teacher network to the student network. Thanks to the teacher annealing strategy, the student can make further progress beyond teacher, since it ensures the student gets rich training signals early in training, but is not limited to only imitating the teacher. However, Student-*R* does not have such consistent gain over Teacher-*R*. The reason behind such phenomenon is still under investigation, we presume that the student network do not has enough capability to imitate such remarkable label knowledge.

Model	F1
Student-*R*	69.6
– TSD loss (TSD)	68.8
– Hint learning loss (HT)	68.4
– TSD & HT	67.9
– Type constraints (TC)	68.0
– Multi-aspect attention (MAA)	68.3
– TC & MAA	67.8
– Knowledge distillation (KD)	67.5

Table 4: Results on the TACRED dataset to investigate the influence of different model architectures.

Overall, our approach achieves quite impressive results (+1.4% on F1) compared with AG-GCN on the TACRED dataset. Similar results can be found on SemEval, we omit the specific analysis due to space limitations.

Ablation Study

To study the contribution of each component in Student-*R*, we run an ablation study on the TACRED dataset (see Table 4). From these ablations, we can observe that: (1) TSD loss is a necessary component that contributes 0.8% gain of F1 to the ultimate performance, we attribute this gain to the softer distribution of the teacher's output, and softer means more informative. (2) The use of hint learning is crucial, since the F1 drops drastically by 1.2% if it is removed, which can be interpreted that it provides an effective guidance to clone the teacher's structure for student. (3) The type constraints contributes about 1.6% F1, which indicates that it is important to let our model aware of the global type knowledge. (4) Removing MAA hurts the result by 1.3% F1, it shows that the participation of multi-aspect information can indeed help students to excavate some knowledge from raw texts. (5) When we remove the type constraints and MAA and use a vanilla knowledge distillation model, the score drops by 1.8%. Conversely, when we remove the knowledge distillation and let the student learn from hard labels, the score drops by 2.1%, which justifies the effectiveness of using knowledge distillation in RE.

Effectiveness of Well-informed Soft Labels

From cases in Figure 3, we attribute the performance gain to two design choices: (1) The introduction of soft type constraints. Intuitively, the global statistics over examples provide more signal than independent sentence. In this way, combining with the soft rules can help neural networks generate reasonable and high-confidence predictions for the instances with complicate semantics. (2) The application of knowledge distillation. It is effective because that the soft output of class distribution from the knowledgeable teacher may carry additional information (e.g., the global awareness and cross-category relationship), such knowledge in soft labels can be successfully transferred from the teacher to student by leveraging knowledge distillation. Overall, our main claims about using soft labels instead of hard labels is that a lot of helpful information can be carried in soft labels that could not possibly be encoded with one-hot vectors.

Sentences	Baseline		Student-*R*	
The Urban League[OBJ-ORGANIZATION], which bought the Colman School[OBJ-ORGANIZATION], is confronting the community issue directly.	no relation	[0.42]	org:subsidiaries	[0.65]
	org:subsidiaries	[0.35]	no relation	[0.26]
	org:parents	[0.14]	org:parents	[0.03]
Wayne A. Holst[SUBJ-PERSON] teaches at the University of Calgary and at St. David's United Church[OBJ-ORGANIZATION].	per:schools attended	[0.51]	per:employee of	[0.91]
	per:employee of	[0.41]	per:schools attended	[0.08]
	no relation	[0.07]	no relation	[0.01]
Cash Minerals Ltd.[OBJ-ORGANIZATION] released a report about the amount of uranium found in the Yukon, where John Graham[SUBJ-PERSON] was born and raised.	no relation	[0.96]	no relation	[0.95]
	per:employee of	[0.02]	per:employee of	[0.03]
	per:schools attended	[0.01]	per:schools attended	[0.01]

Figure 3: Outputs of Baseline (C-GCN) and Student-*R* on samples from TACRED, underscore marks the ground truth.

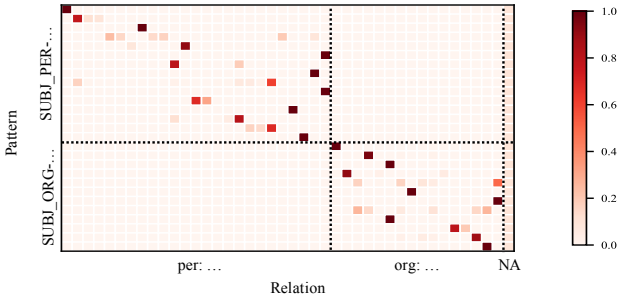

Figure 4: Visualization of soft rules on the TACRED dataset. Blocks are highlighted with different degrees according to the probabilities from pattern to relation.

Visualization of Soft Rules

We visualize the bipartite pattern-to-relation graph to further show how the soft type constraints help RE boost the performance. From Figure 4, we can clearly see that soft rules is the refinement of hard type constraint, which can also be regarded as some prior knowledge and help model rectify some wrong cases and keep the right ones. For example, the probability of the relation between a *person* and an *organization* being *employee of* is 0.92. It is intuitive, when we talk about a person and an organization, our immediate reaction is that the man is a employee of the organization, then make further judgments based on the specific statements.

Related Work

In this paper, we deal with the supervised relation extraction (RE) problem, which is also known as relation classification (RC). Recently, various models are proposed based on different neural architectures, such as convolutional neural networks (Zeng et al. 2014; 2015), recurrent neural networks (Xu et al. 2015; Zhang et al. 2017), graph convolutional networks (Zhang, Qi, and Manning 2018; Guo, Zhang, and Lu 2019) and transformers (Alt, Hübner, and Hennig 2019; Zhang et al. 2019). But overall, existing approaches mostly use hard labels (e.g., one-hot vectors) as the optimization objective in training, ignoring rich semantic relevance among relations. Although Liu et al. (2017) introduced a soft-label method, they aims to alleviate the wrong label problem during training and neglect the cross-category relationships too. Besides, there are also some efforts aimed at using external information to boost the performance of RE, including entity description (Ji et al. 2017), entity type (Vashishth et al. 2018) and relation-specific constraint (Lei et al. 2018). However, they tend to focus on the distantly supervised relation extraction (DSRE) and acquire information from knowledge base (KB), such heavy reliance on KB limits their scalability. For example, all relations in the SemEval dataset cannot be retrieved from existing KB (Hendrickx et al. 2010). To get rid of the limitation of KB, Su et al. (2018a) explored to consider the dependency of relation within an entity pair for DSRE. Su et al. (2018b) proposed to combat the wrong label problem in DSRE with global statistics. However, there is no prior study has explored to count soft labels from text corpus with a global perspective, which is a new opportunity to further improve the performance for RE.

Our work is also related to knowledge distillation (Hinton, Vinyals, and Dean 2015). Yim et al. (2017) concluded that it can bring three benefits: fast optimization, knowledge transfer and performance improvement. Recently, it shines brilliantly in various fields, especially for computer vision (Chen et al. 2017; Mishra and Marr 2018; Park et al. 2019). Nevertheless, in the natural language processing area, it is still in infancy stage and used for few tasks (Liu, Chen, and Liu 2019; Tan et al. 2019; Clark et al. 2019). In this paper, we employ knowledge distillation to help us mine soft labels and transfer knowledge for RE.

Conclusions and Future Work

In conclusion, we explore a new viewpoint of utilizing soft labels to boost the performance of RE. Specifically, we first construct a bipartite graph to discover soft rules between entity types and relations from entire corpus, and then combine soft rules with a GCN-based model to achieve knowledgeable teacher. Furthermore, we present a multi-aspect attention mechanism to help student mine the potential of raw text and adopt knowledge distillation to transfer dark knowledge from teacher to student. Finally, experimental results on two public datasets prove the effectiveness of our model. In the future, we plain to explore soft labels with other forms and adapt our model to other NLP tasks.

Acknowledgements

This research is supported by the National Key Research and Development Program of China (grant No. 2016YFB 0801003) and the Strategic Priority Research Program of Chinese Academy of Sciences (grant No. XDC02040400).

References

Alt, C.; Hübner, M.; and Hennig, L. 2019. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. In *Proc. of AAAI*.

Chen, G.; Choi, W.; Yu, X.; Han, T.; and Chandraker, M. 2017. Learning efficient object detection models with knowledge distillation. In *Proc. of NeurIPS*.

Clark, K.; Luong, M.-T.; Khandelwal, U.; Manning, C. D.; and Le, Q. 2019. Bam! born-again multi-task networks for natural language understanding. In *Proc. of ACL*.

Deng, Y.; Xie, Y.; Li, Y.; Yang, M.; Du, N.; Fan, W.; Lei, K.; and Shen, Y. 2019. Multi-task learning with multi-view attention for answer selection and knowledge base question answering. In *Proc. of AAAI*.

Distiawan, B.; Weikum, G.; Qi, J.; and Zhang, R. 2019. Neural relation extraction for knowledge base enrichment. In *Proc. of ACL*.

Guo, Z.; Zhang, Y.; and Lu, W. 2019. Attention guided graph convolutional networks for relation extraction. In *Proc. of ACL*.

Hendrickx, I.; Kim, S. N.; Kozareva, Z.; Nakov, P.; Ó Séaghdha, D.; Padó, S.; Pennacchiotti, M.; Romano, L.; and Szpakowicz, S. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proc. of SemEval@ACL*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. In *Proc. of NeurIPS*.

Ji, G.; Liu, K.; He, S.; and Zhao, J. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *Proc. of AAAI*.

Koch, M.; Gilmer, J.; Soderland, S.; and Weld, D. S. 2014. Type-aware distantly supervised relation extraction with linked arguments. In *Proc. of ACL*.

Lei, K.; Chen, D.; Li, Y.; Du, N.; Yang, M.; Fan, W.; and Shen, Y. 2018. Cooperative denoising for distantly supervised relation extraction. In *Proc. of COLING*.

Liu, T.; Wang, K.; Chang, B.; and Sui, Z. 2017. A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proc. of ACL*.

Liu, J.; Chen, Y.; and Liu, K. 2019. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *Proc. of AAAI*.

Lopez-Paz, D.; Bottou, L.; Schölkopf, B.; and Vapnik, V. 2016. Unifying distillation and privileged information. In *Proc. of ICLR*.

Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; and McClosky, D. 2014. The stanford corenlp natural language processing toolkit. In *Proc. of ACL (System Demonstrations)*.

Mishra, A., and Marr, D. 2018. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *Proc. of ICLR*.

Mitra, A.; Clark, P.; Tafjord, O.; and Baral, C. 2019. Declarative question answering over knowledge bases containing natural language text with answer set programming. In *Proc. of AAAI*.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In *Proc. of CVPR*.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*.

Su, S.; Jia, N.; Cheng, X.; Zhu, S.; and Li, R. 2018a. Exploring encoder-decoder model for distant supervised relation extraction. In *Proc. of IJCAI*.

Su, Y.; Liu, H.; Yavuz, S.; Gur, I.; Sun, H.; and Yan, X. 2018b. Global relation embedding for relation extraction. In *Proc. of NAACL*.

Tan, X.; Ren, Y.; He, D.; Qin, T.; Zhao, Z.; and Liu, T.-Y. 2019. Multilingual neural machine translation with knowledge distillation. In *Proc. of ICLR*.

Vashishth, S.; Joshi, R.; Prayaga, S. S.; Bhattacharyya, C.; and Talukdar, P. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. In *Proc. of EMNLP*.

Xu, Y.; Mou, L.; Li, G.; Chen, Y.; Peng, H.; and Jin, Z. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *Proc. of EMNLP*.

Yang, C.; Xie, L.; Qiao, S.; and Yuille, A. L. 2019. Training deep neural networks in generations: A more tolerant teacher educates better students. In *Proc. of AAAI*.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proc. of CVPR*.

Yu, B.; Zhang, Z.; Liu, T.; Wang, B.; Li, S.; and Li, Q. 2019. Beyond word attention: Using segment attention in neural relation extraction. In *Proc. of IJCAI*.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proc. of COLING*.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proc. of EMNLP*.

Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; and Manning, C. D. 2017. Position-aware attention and supervised data improve slot filling. In *Proc. of EMNLP*.

Zhang, C.; Li, Y.; Du, N.; Fan, W.; and Yu, P. S. 2018. On the generative discovery of structured medical knowledge. In *Proc. of SIGKDD*.

Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019. Ernie: Enhanced language representation with informative entities. In *Proc. of ACL*.

Zhang, Y.; Qi, P.; and Manning, C. D. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proc. of EMNLP*.