Progressive Multitask Learning with Controlled Information Flow for Joint Entity and Relation Extraction

Kai Sun,[1,2] **Richong Zhang,**[1,2*] **Samuel Mensah,**[1,2] **Yongyi Mao,**[3] **Xudong Liu**[1,2]
[1]Beijing Advanced Institution for Big Data and Brain Computing, Beihang University, Beijing, China
[2]SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China
[3]School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
sunkai@buaa.edu.cn, zhangrc@act.buaa.edu.cn, samensah@buaa.edu.cn, ymao@uottawa.ca, liuxd@act.buaa.edu.cn

Abstract

Multitask learning has shown promising performance in learning multiple related tasks simultaneously, and variants of model architectures have been proposed, especially for supervised classification problems. One goal of multitask learning is to extract a good representation that sufficiently captures the relevant part of the input about the output for each learning task. To achieve this objective, in this paper we design a multitask learning architecture based on the observation that correlations exist between outputs of some related tasks (e.g. entity recognition and relation extraction tasks), and they reflect the relevant features that need to be extracted from the input. As outputs are unobserved, our proposed model exploits task predictions in lower layers of the neural model, also referred to as early predictions in this work. But we control the injection of early predictions to ensure that we extract good task-specific representations for classification. We refer to this model as a Progressive Multitask learning model with Explicit Interactions (PMEI). Extensive experiments on multiple benchmark datasets produce state-of-the-art results on the joint entity and relation extraction task.

Introduction

Multitask learning (MTL) is an important methodology that simultaneously co-models multiple related tasks in a single model. One of the earliest proposed MTL architectures learns a shared representation for multiple tasks, where this shared representation is utilized by task-specific structures independently to learn task-specific representations for supervision (Collobert and Weston 2008). Thus, it induces an inductive bias that enhances the model's ability to generalize well on new inputs. This is a basic MTL structure which has been used successfully in several natural language processing (NLP) tasks (Liu et al. 2019b; Hu et al. 2019).

Although impressive performance has been achieved by considering the basic MTL architecture, some models have considered incorporating early predictions (predictions of the input in lower layers of the neural network) of the task-specific structures to improve the task-specific representations (He, Lee, and and‘ Daniel Dahlmeier 2019; Zhao et al. 2019). However, their approach applies a deterministic model (e.g. multilayer perceptron) on both the early predictions and shared representations to output the task-specific representations. Obviously, deterministic models do not take into account the randomness of the early predictions, as the quality of these predictions is heavily dependent on the quality of early classifiers. A more natural approach is to consider a stochastic model that possesses some inherent randomness. Besides, it is important to additionally control the flow of information from the early predictions since they are just an approximation to the ground truth, and not the ground truth itself. That way we are sure to reduce the noise that comes with early predictions to extract more expressive task-specific representations.

Moreover, the previous MTL models (Liu et al. 2019b; Hu et al. 2019) only exploit the implicit interaction that is captured by the shared representation. Our understanding of task relatedness informs us that correlations actually exist between the outputs of some related tasks, e.g. the entity recognition and relation classification tasks. Since we can obtain access to early predictions, we can model such correlations explicitly to improve task-specific representations.

Motivated by these findings, the goal of this paper is to develop a multi-task learning architecture that incorporates early predictions of task-specific structures to improve the learning of task-specific representations. Specifically, we follow an approach used to extract the minimal sufficient statistics of an input about an output using neural networks (Alemi et al. 2017), and develop stochastic maps that consider the shared representations and the interaction of early predictions of task-specific networks to extract good task-specific representations for supervision.

To verify our proposed multitask learning approach, we choose the joint entity and relation extraction as our target task due to its popularity in NLP. So far, many works (Miwa and Sasaki 2014; Gupta, Schütze, and Andrassy 2016; Fu, Li, and Ma 2019) have focused on leveraging multitask learning to solve this joint task by taking entity recognition (ER) as one task, and relation classification (RC) as another task. The ER aims to extract all entities in the sentence, and the RC aims to classify the relations between all word pairs in the sentence. However, most of the previous works do not consider leveraging the early prediction to improve the task-specific representation nor do they explicitly model the interactions between the two tasks. We show that our proposed

[*] Corresponding author: zhangrc@act.buaa.edu.cn

approach can improve the performance for this task.

Our contribution is summarized as follows:

- We propose a progressive multi-task learning model (PMEI) which leverages interactions of early predictions to improve the task-specific representation.
- Our model employs stochastic maps to encode both the shared representations between tasks and the early predictions from tasks. In particular, we ensure the information flow from early predictions is controlled to reduce the noise that comes with it.
- We take the joint entity and relation extraction as a concrete example and apply our proposed method on this joint task. Extensive experiments on several benchmark datasets show the effectiveness of the proposed method.

Related Work

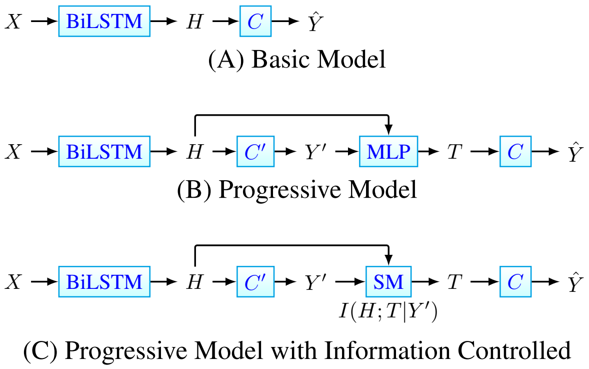
Multitask Learning

The existing multitask learning architectures proposed so far can be categorized according to their topological structure. We have those with a flat structure, a graph structure, and a hierarchical structure (Sun et al. 2020). A suitable structure depends on the relatedness of the tasks. The most commonly used multitask learning architecture is the flat structure. In this architecture, task-specific networks are fed with a shared representation, and then each task-specific network learns in isolation without interaction. This structure has been successfully applied in a variety of tasks, including relation extraction (Fu, Li, and Ma 2019) and natural language understanding (Liu et al. 2019b). However, as pointed by (Liu, Qiu, and Huang 2017), the shared representation exploited by one task may contain noise brought by other tasks, contaminating the task-specific representations. To address this weakness, the hierarchical structure places different tasks at different layers of the network according to the complexity of the tasks. The graph structure model interactions dynamically among the tasks to learn task-specific representations. Our proposed multitask learning architecture can be categorized under the graph structure.

Joint Entity and Relation Extraction

Traditional approaches proposed to solve the entity and relation extraction task use a two-step pipeline-based approach (Zelenko, Aone, and Richardella 2003). However, these approaches are faced with error propagation from the entity recognition task to the relation classification task, and cannot leverage the interactions between the two tasks. Recent approaches consider to treat the tasks jointly. Among these works, we have those that consider a sequence-to-sequence approach (Zeng et al. 2018, 2019; Zeng, Zhang, and Liu 2020), but these approaches fail to effectively deal with the overlapping relation problem (Wei et al. 2020). Other works have considered a sequence labelling approach to address the problem (Zheng et al. 2017b; Dai et al. 2019; Takanobu et al. 2019; Wei et al. 2020).

More recently, multitask learning methods have been proposed to address the joint task due to its ability to exploit interactions among related tasks to learn good task-specific representations (Miwa and Sasaki 2014; Miwa and Bansal 2016; Gupta, Schütze, and Andrassy 2016; Fu, Li, and Ma 2019; Zeng, Zhang, and Liu 2020). Although the aforementioned MTL methods show satisfactory performance, they exploit only the implicit interactions that is captured in the shared representations of the related tasks. Besides, they do not exploit early predictions of the ER and RC tasks as seen in other NLP tasks (He, Lee, and and‘ Daniel Dahlmeier 2019; Zhao et al. 2019). Without modeling such information, these methods are limited in performance.



(A) Basic Model

(B) Progressive Model

(C) Progressive Model with Information Controlled

Figure 1: Our models for single task learning.

Method

In this section, we gradually present our model architecture. We first introduce a progressive classification model on a single task to show the rationale behind our proposed approach. Then we introduce the progressive classification model in the context of multitask learning. Finally, we demonstrate the application of the proposed multitask learning methods on the joint ER and RC tasks.

Progressive Classification on Single Task Learning

In this section, we describe our models for single-task learning. The overview of our models are shown in Figure 1.

Consider the basic model in Figure 1(A). Let $X \in \mathcal{X}$ be an input random variable (e.g, a sentence), and $Y \in \mathcal{Y}$ be an output random variable (e.g. class label). We employ a bidirectional LSTM (BiLSTM) to extract a contextual representation $H \in \mathcal{H}$ from X. A classification model is defined as the map $C : \mathcal{H} \to p(\mathcal{Y})$. The function C takes H as input and outputs the probability distribution $p(\hat{Y}) = C(H)$ over the output space. This is a basic model and requires that the Markov relation $Y \to X \to H$ is satisfied. Thus, for the joint distribution $p(x, y, h)$ which factors as follows:

$$p(x, y, h) = p(h|x, y)p(x, y), \tag{1}$$

it assumes that the conditional distribution $p(h|x, y) = p(h|x)$ under the Markov constraint. This means that H is a function of X and it is defined by X exclusively. In other words, H cannot provide any new information about Y, except for what is contained in X. Suppose H has access to Y, the classification task becomes easy, but this is impossible since Y is unobserved. This observation leads us to design a model where some knowledge of Y, denoted as Y', is used as additional information to improve the representation of H.

To this end, we progressively improve the representation of H as shown in Figure 1(B). Specifically, we employ a classifier C' which takes H as input and produces the prediction Y'. Here, Y' can be interpreted as an early prediction for the task, and it is likely to approximate the output Y. Thus, these early predictions provide some information about Y which can be used as additional information to H to extract a more expressive representation T. In this model architecture, a multilayer perceptron (MLP) is applied on both H and Y' to learn the representation T.

Although employing early predictions have proved to be useful in several tasks, previous works (He, Lee, and and‘ Daniel Dahlmeier 2019; Zhao et al. 2019) merely pass Y' and H into an MLP to extract T. A key observation from these works show that Y' can indeed improve the representation of H. However, these approaches ignore the fact that Y' may not necessarily be the ground truth, so not all the information contained in Y' may be beneficial to the model performance. Thus, we argue that it is necessary to control the information flow of Y'. Specifically, we construct a stochastic map (SM) to model the mutual information between H and T conditioned on Y', denoted as $I(H;T|Y')$. In this way, we can control the information flow of Y' by controlling $I(H;T|Y')$ during optimization. A small value of $I(H;T|Y')$ means T is largely determined by Y', while a large value means T is largely determined by H. The mutual information (MI) $I(H;T|Y')$ is given by

$$\begin{aligned} I(H;T|Y') &= \int dh\, dt\, dy'\, p(h,t,y')\, log\, \frac{p(t|h,y')}{p(t|y')} \\ &\leq \int dh\, dt\, dy'\, p(h,t,y')\, log\, \frac{p(t|h,y')}{r(t|y')}, \end{aligned} \tag{2}$$

where $r(t|y')$ is a variational approximation to $p(t|y')$, inducing an upper bound on $I(H;T|Y')$. Minimizing the upper bound of $I(H;T|Y')$ is the same as minimizing the KL-divergence between $p(t|h,y')$ and $r(t|y')$.

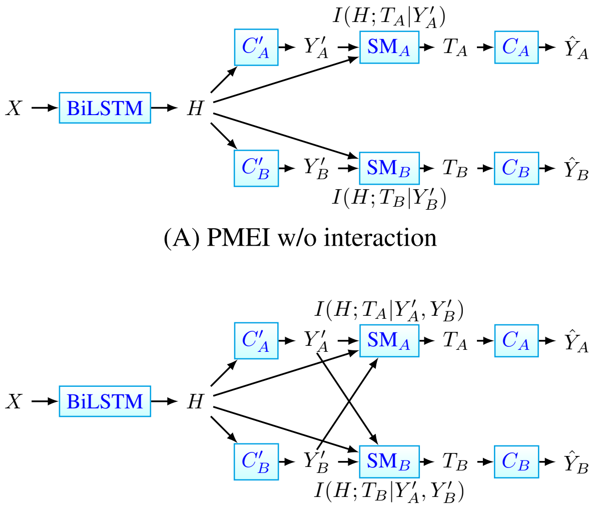
$$I(H;T|Y') \leq \text{KL}(p(t|h,y')||r(t|y')) \tag{3}$$

As the KL-divergence approaches zero, $p(t|h,y')$ approximates $r(t|y')$. And in this case, t is determined by y' to a great extent. Thus, by controlling the KL-divergence between $p(t|h,y')$ and $r(t|y')$, we can control the injection of y' into t.

Progressive Classification on Multitask Learning

We design a multitask learning architecture based on our proposed single-task learning method which considers the control of information flow. Figure 2 shows the architecture of our models.

Suppose $\mathcal{Y}_A$ and $\mathcal{Y}_B$ are the output spaces of two different but related tasks A and B. The two tasks have different outputs, but share the same input in our setting. Let H be the shared representation of both tasks modeled by BiLSTM. We employ classifiers C'_A and C'_B corresponding to tasks A and B to make early predictions Y'_A and Y'_B. Following the approach proposed in the single-task learning method in Figure 1(C), we can control the injection of Y'_A (or Y'_B) in H to extract task-specific representations T_A (or T_B).



(A) PMEI w/o interaction

(B) PMEI

Figure 2: Our models on the multi-task learning

It is possible that correlations exist between the learning tasks in the multitask learning architecture. But the model depicted in Figure 2(A) does not model interactions explicitly, but exploit only the implicit interactions in H. Without modeling such explicit interactions, as shown in these works (Lan et al. 2017; He, Lee, and and‘ Daniel Dahlmeier 2019; Zhao et al. 2019; Dankers et al. 2019; Liu et al. 2019a), the multitask learning model cannot properly distinguish the relevant features for the individual tasks.

As a solution, we consider one observation: correlations exist between outputs of several tasks (He, Lee, and and‘ Daniel Dahlmeier 2019; Zhao et al. 2019). Assuming we have early predictions for multiple tasks, we can exploit these interactions to improve task-specific representations. Thus, a natural idea should be that the conditional MI term should be under the condition of both Y'_A and Y'_B, i.e. $I(H;T_A|Y'_A,Y'_B)$ and $I(H;T_B|Y'_A,Y'_B)$, so that the model effectively exploits the interactions between the two tasks. We therefore construct an upper bound over $I(H;T_A|Y'_A,Y'_B)$ as

$$I(H;T_A|Y'_A,Y'_B) \leq \text{KL}(p(t_A|h,y'_A,y'_B)||r(t_A|y'_A,y'_B)) \tag{4}$$

An upper bound over $I(H;T_B|Y'_A,Y'_B)$ will follow the same formulation as $I(H;T_A|Y'_A,Y'_B)$.

Our Model for the Joint ER and RC Tasks

In this section, we demonstrate the application of the proposed multitask learning method in Figure 2(B) for the joint extraction of entities and relations.

Given a sentence $s = \{w^1, w^2, \cdots, w^n\}$ consisting of n words, and a set of l pre-defined relation types $R =$

$\{\rho^1, \cdots, \rho^l\}$, the joint task aims to extract all relational facts in sentence s. In this paper, a relational triple is represented in the form $\langle e^i, \rho, e^j \rangle$, where e^i, e^j are entity words (i.e, entities written as a single word) or heads of multi-token entities corresponding to $w^i, w^j \in s$, and the relation $\rho \in R$. Given a word pair (w^i, w^j), the goal is to predict the probability $\hat{y}_{(i,j)}$ that the relational triple $\langle w^i, \rho, w^j \rangle$ is factual. Besides, the entity recognition task which takes each word $w^i \in s$ and predicts a probability $\hat{y}_i$ over BIOES labels (Fu, Li, and Ma 2019) can be used to identify the head and tail words of multi-token entities for the extracted relational triple.

Learning a shared representation When addressing this problem, we first map the word sequence s to a set of vectors $x = \{x^1, x^2, \ldots, x^n\}$, where $x^i \in \mathbb{R}^d$ is a word embedding (Pennington, Socher, and Manning 2014) with a dimension size of d. Denote X, a random variable corresponding to the initial vectors of sentence s. We construct a shared representation H for the ER and RC tasks by means of a BiLSTM.

Learning task-specific representations In our model for the joint task, the tasks A and B correspond to the ER and RC tasks. Let C'_e and C'_r be classification models for the ER and RC tasks which takes H as input and produce early predictions Y'_e and Y'_r. In fact, there are correlations between the outputs of the ER and RC tasks. For example, the output of the ER task can provide information on whether the words w^i, w^j in a relational triple $\langle w^i, \rho, w^j \rangle$ are entity words or multi-tokens. Moreover, it provides information on which pairs of words to focus on in the RC task, since not all words in the sentence are entity words or involved in multi-token entities. Meanwhile, the relation r of the extracted relational triple in the RC task provides information on the entity type for w^i, w^j (typically when w^i, w^j are entity words).

As already mentioned, the correlations between the outputs of the ER and RC tasks, as well as its ability to increasingly improve predictions makes it necessary to exploit H, Y'_e and Y'_r for the extraction of task-specific representations T_e and T_r. Employing stochastic maps for the respective tasks, we control the information flow to T_e and T_r by minimizing the mutual information $I(H; T_e|Y'_e, Y'_r)$ and $I(H; T_r|Y'_e, Y'_r)$,

$$I(H; T_e|Y'_e, Y'_r) \leq \mathrm{KL}(p(t_e|h, y'_e, y'_r)||r(t_e|y'_e, y'_r)) \quad (5)$$
$$I(H; T_r|Y'_e, Y'_r) \leq \mathrm{KL}(p(t_r|h, y'_e, y'_r)||r(t_r|y'_e, y'_r)) \quad (6)$$

Both (5) and (6) are solved similarly. We therefore focus on how we solve (5). To find solutions to the distributions $p(t_e|h, y'_e, y'_r)$ and $r(t_e|y'_e, y'_r)$, we follow an approach proposed in (Alemi et al. 2017). Specifically, each of the distribution is modeled by two neural networks $f^\mu(\cdot)$ and $f^\sigma(\cdot)$, which are respectively used to compute the mean μ and standard deviation σ of t_e. The representation t_e is then sampled from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. At this point, it is important to note that $y'_e \in \mathbb{R}^{n\times 5}$ is a probability distribution over BIEOS labels (Fu, Li, and Ma 2019), and $y'_r \in \mathbb{R}^{n\times n\times l}$ is a probability distribution over the distinct relations. As a consequence, the dimension of y'_e and y'_r are unequal, and this must be considered when integrating both information in a neural network.

Now for $r(t_e|y'_e, y'_r)$, we employ two multilayer perceptrons (MLPs) f^μ and f^σ. Both take in as input a concatenation $y' = [y'_e; f^t(y'_r)]$ to compute a mean and a standard deviation for t_e. In this case, the function f^t is a max pool function to transform y'_r to the dimension space $\mathbb{R}^{n\times l}$ to ease the concatenation.

We now discuss how we model $p(t_e|h, y'_e, y'_r)$. Here, instead of employing simple MLPs, we consider gated recurrent unit cells (GRUCells) to fully model the interactions between the two tasks. GRUCell$_\mu$ is to model the mean of t_e, and GRUCell$_\sigma$ is to model the standard deviation of t_e. Both GRUCells have similar network structures. The operation of a GRUCell$_\mu$ is as follows:

$$\begin{aligned} z &= \sigma\left(W_{\mathrm{z}}(h \oplus y')\right) \\ u &= \sigma\left(W_{\mathrm{u}}(h \oplus y')\right) \\ \check{h} &= \tanh\left(W_{\mathrm{o}}((u * h) \oplus y')\right) \\ \mu &= (1 - z) * h + z * \check{h} \end{aligned} \quad (7)$$

where y' is the concatenation of $f^t(y'_r)$ and y'_e, $\oplus$ is a concatenation operator, and W_z, W_u, W_o are learnable parameters.

Task-specific classification Let C_e and C_r be classification models that take the respective inputs $T_e \in \mathcal{T}_e$ and $T_r \in \mathcal{T}_r$, modeled by the mutual information $I(H; T_e|Y'_e, Y'_r)$ and $I(H; T_r|Y'_e, Y'_r)$, and outputs the corresponding probability distributions $C_e(T_e) = p(\hat{Y}_e) \in p(\mathcal{Y}_e)$ and $C_r(T_r) = p(\hat{Y}_r) \in p(\mathcal{Y}_r)$. We can define the classification model for the ER task as the map

$$C_e : \mathcal{T}_e \rightarrow p(\mathcal{Y}_e), \quad (8)$$

and the classification model for the RC task as the map

$$C_r : \mathcal{T}_r \rightarrow p(\mathcal{Y}_r), \quad (9)$$

We take C_e and C_r as neural networks with its own set of parameters. Now let $t_e = \{t_e^1, t_e^2, ..., t_e^n\}$ be an instance of the random variable T_e, and $t_r = \{t_r^1, t_r^2, ..., t_r^n\}$ be an insstance of the random variable T_r. C_e takes as input the features t_e and makes a prediction $C_e(t_e)$ over BIEOS labels for each $t_e^i \in t_e$. Specifically, for the feature vector t_e^i corresponding to the i-th word in the sentence, the probability distribution $\hat{y}_i \in C_e(t_e)$ is computed as follows:

$$\hat{y}_i = \mathrm{softmax}(W_{\mathrm{e}} t_e^i + b_{\mathrm{e}}), \quad (10)$$

where $\theta_{\mathrm{E}} = \{W_{\mathrm{e}}, b_e\}$ are trainable model parameters. Hence the set of predictions $C_e(t_e) = \{\hat{y}_i | t_e^i \in t_e\}$. Also, the classification model C_r takes as input the features t_r and makes a prediction $C_r(t_r)$ for each pair of feature vectors in t_r. More specifically, given $t_r^i, t_r^j \in t_r$, where $t_r^i \neq t_r^j$, the prediction $\hat{y}_{(i,j)} \in C_r(t_r)$ is defined as follows:

$$\begin{aligned} m &= \phi\left(W_{\mathrm{m}}(t_r^i \oplus t_r^j)\right) \\ \hat{y}_{(i,j)} &= \sigma\left(W_{\mathrm{r}} m + b_r\right) \end{aligned} \quad (11)$$

where $\oplus$ is a concatenation operator, $\phi(\cdot)$ is the ReLU activation function, $\sigma(\cdot)$ is the sigmoid activation function. $\theta_{\mathrm{R}} = \{W_{\mathrm{m}}, W_{\mathrm{r}}, b_r\}$ are learnable model parameters. Hence the set of predictions $C_r(t_r) =: \{\hat{y}_{(i,j)} | t_r^i, t_r^j \in t_r, t_r^i \neq t_r^j\}$.

Training Objective

The final training objective of our model is in three parts: (1) the supervision loss of C_e and C_r. (2) the supervision loss of C'_e and C'_r. (3) the loss produced by the MI terms $I(H; T_e | Y'_e, Y'_r)$ and $I(H; T_r | Y'_e, Y'_r)$.

The supervision loss of C_e and C_r is computed as follows:

$$\begin{aligned} L_{\mathrm{e}}(w^i) &= \mathrm{CrossEntropy}\left(y_i, \hat{y}_i\right) \\ L_{\mathrm{r}}(\langle w^i, \rho, w^j \rangle) &= \mathrm{CrossEntropy}\left(y_{(i,j)}, \hat{y}_{(i,j)}\right) \end{aligned} \tag{12}$$

where y_i and $y_{(i,j)}$ are the respective ground truth values of word w and relational triple $\langle w^i, \rho, w^j \rangle$, and $\hat{y}_i$ and $\hat{y}_{(i,j)}$ are the predictions from the C_e and C_r.

The supervision loss L of C_e and C_r over all words and relational triples for all sentences is then calculated as follows.

$$L = \sum_s \left(\sum_{w^i \in s} L_{\mathrm{e}}(w^i) + \sum_{w^i, w^j \in s, \rho \in R} L_{\mathrm{r}}(\langle w^i, \rho, w^j \rangle) \right) \tag{13}$$

The supervision loss L' of C'_e and C'_r has a similar computation as L. The loss L_{IC_e} of $I(H; T_e | Y'_e, Y'_r)$ is computed as

$$L_{IC_e} = \sum_s \mathrm{KL}[p(t_e | h_s, y'_e, y'_r), r(t_e | y'_e, y'_r)] \tag{14}$$

The loss of $I(H; T_r | Y'_e, Y'_r)$ denoted as L_{IC_r} has a similar computation as L_{IC_e}. Hence the total loss is given by

$$L_{total} = \beta_e \, L_{IC_e} + \beta_r \, L_{IC_r} + \alpha \, L' + L \tag{15}$$

where β_e, β_r and α are positive parameters to control the weight of loss.

Experiment

Datasets

Recent works (Zeng, Zhang, and Liu 2020; Wei et al. 2020) on the joint entity and relation extraction task mainly evaluate on NYT (Riedel, Yao, and McCallum 2010), WebNLG (Gardent et al. 2017), NYT10 (Riedel, Yao, and McCallum 2010) and NYT11 (Hoffmann et al. 2011) datasets. We directly use the preprocessed NYT and WebNLG datasets released by (Zeng et al. 2018), and the preprocessed NYT10 and NYT11 datasets released by (Takanobu et al. 2019). Note that NYT and WebNLG datasets mark only the tail word of an entity. We take a further step to tag entities with the conventional BIOES tagging scheme. Table 1 and 2 show the statistics of the datasets.

Dataset	Train	Dev	Test
NYT	56195	5000	5000
WebNLG	5019	500	703
NYT10	70339	-	4006
NYT11	62648	-	369

Table 1: Statistics of the datasets.

Dataset	NYT		WebNLG	
	Train	Test	Train	Test
Multi-token entities	39.1%	38.9%	64.2%	63.8%
Single-token entities	60.9%	61.1%	35.9%	36.2%
Relations	24	24	170	170

Table 2: Percentages of multi-token entities and single-token entities, and the number of relations on NYT and WebNLG.

Evaluation Protocols

We follow the evaluation protocols of previous works (Zeng, Zhang, and Liu 2020; Wei et al. 2020) and report the Precision, Recall and micro-F1 performance of our models on the datasets. Each reported result is the average performance over three runs using different random seeds. Best performance in boldfaced. Evaluation is performed on the *partial match* task and the *exact match* task. The partial match task requires the relation, and the heads of both the entities in the extracted relational triple to be correct. The exact match task strictly requires the relation, the head and tail of both entities in the extracted relational triple to be correct.

Implementation Details

We initialize word embeddings with either Glove (Pennington, Socher, and Manning 2014) or BERT (Devlin et al. 2019). BERT-based models directly use BERT embeddings as H. We use a batch size of 50 for Glove models, and a mini-batch of 6 for BERT models. We use Adam optimizer with an initial learning rate $1e^{-3}$ for Glove models, and $1e^{-5}$ for BERT models. We empirically fine-tune the hyperparameters of the model on the development set. Since NYT10 and NYT11 have no development set, we randomly select 10% of samples from the training set as the development set. Due to the space limit, we list the hyperparameter values used for all models on the datasets. We search the word embedding size in [100, 300], BiLSTM embeddings in [100, 200], dropout for word embeddings in [0.1, 0.2, ... , 0.8], β_e and β_r in [e^{-3}, e^{-4}, ... , e^{-9}], and α in [1.0, e^{-1}, ... , e^{-6}]. We implement our model using PyTorch on a Linux machine with a GPU device NVIDIA V100 NVLINK 32GB. The code is available in our Github repository.[1]

Performance Comparison

We compare our models with recent works including the sequence-to-sequence (seq2seq) models such as OneDecoder (Zeng et al. 2018), MultiDecoder (Zeng et al. 2018), OrderRL (Zeng et al. 2019), and the sequence labeling models such as NovelTagging (Zheng et al. 2017b), ReHession (Liu et al. 2017), LSTM-CRF (Zheng et al. 2017a),

[1] https://github.com/BDBC-KG-NLP/Progressive_AAAI2021

	NYT			WebNLG		
Model	P	R	F1	P	R	F1
OneDecoder	59.4	53.1	56.0	32.2	28.9	30.5
MultiDecoder	61.0	56.6	58.7	37.7	36.4	37.1
OrderRL	77.9	67.2	72.1	63.3	59.9	61.6
CASREL	84.2	83.0	83.6	86.9	80.6	83.7
CASREL$_{BERT}$	89.7	89.5	89.6	**93.4**	90.1	91.8
MTL	83.9	83.1	83.5	84.9	86.3	85.6
PMEI	**88.7**	**86.8**	**87.8**	**88.7**	**87.6**	**88.1**
MTL$_{BERT}$	89.4	**89.9**	89.7	89.4	92.0	90.7
PMEI$_{BERT}$	**90.5**	89.8	**90.1**	91.0	**92.9**	**92.0**
NovelTagging	62.4	31.7	42.0	52.5	19.3	28.3
GraphRel$_{1p}$	62.9	57.3	60.0	42.3	39.2	40.7
GraphRel$_{2p}$	63.9	60.0	61.9	44.7	41.1	42.9
CopyMTL-One	72.7	69.2	70.9	57.8	60.1	58.9
CopyMTL-Mul	75.7	68.7	72.0	58.0	54.9	56.4
MTL	77.4	76.4	76.9	76.7	74.8	75.7
PMEI	**84.5**	**84.0**	**84.2**	**78.8**	**77.7**	**78.2**
MTL$_{BERT}$	87.0	88.7	87.8	**80.9**	82.0	81.4
PMEI$_{BERT}$	**88.4**	**88.9**	**88.7**	80.8	**82.8**	**81.8**

Table 3: Results on NYT and WebNLG with partial match (top) and exact match (bottom).

PA-LSTM-CRF (Dai et al. 2019), HRL (Takanobu et al. 2019), CASREL (Wei et al. 2020), as well as the multitask learning models such as SPTree (Miwa and Bansal 2016), GraphRel (Fu, Li, and Ma 2019), CopyMTL (Zeng, Zhang, and Liu 2020). As a baseline, we include a basic MTL model (MTL) which directly pass H into the classifiers C_e and C_r for classification. Table 3 shows the results on NYT and WebNLG datasets. Table 4 shows the results on NYT10 and NYT11.

Glove embedding results We compare our Glove models (MTL, PMEI) with recent works. Although MTL has a simple architecture, it significantly outperforms some recent seq2seq models including OrderRL, CopyMTL-one and CopyMTL-Mul. As noted in (Wei et al. 2020), seq2seq architectures may not be ideal to address the joint task, especially for the overlapping relation problem. The low performance of the seq2seq models when compared to our MTL is consistent with the findings by (Wei et al. 2020). We also find that PMEI significantly outperforms CASREL on NYT, WebNLG, and even outperforms CASREL$_{BERT}$ on NYT11, while showing competitive performance with CASREL$_{BERT}$ on NYT10. Additionally, we realize that the F1 performance of PMEI significantly drops on the exact match task on WebNLG as compared to the partial match task. Note that 60% of entities in WebNLG are multi-token entities, therefore the exact match task is particularly difficult on this dataset.

BERT embedding results We see an improvement when using BERT embeddings on the exact and partial match tasks, suggesting that incorporating prior knowledge induced by BERT in the joint ER and RC tasks is an effective approach. Comparing our BERT models with other recent BERT models, we find that our model PMEI$_{BERT}$ outperforms the CASREL$_{BERT}$ on NYT10 and NYT11, and show competitive performance with CASREL$_{BERT}$ on NYT and WebNLG.

	NYT10			NYT11		
Model	P	R	F1	P	R	F1
MultiDecoder	56.9	45.2	50.4	34.7	53.4	42.1
CASREL$_{BERT}$	77.7	68.8	73.0	50.1	58.4	53.9
MTL	75.0	65.9	70.2	51.9	57.0	54.3
PMEI	**79.1**	**67.2**	**72.6**	**56.0**	**58.6**	**57.2**
MTL$_{BERT}$	77.9	69.9	73.7	54.3	59.7	56.9
PMEI$_{BERT}$	**79.1**	**70.4**	**74.5**	**55.8**	**59.7**	**57.7**
NovelTagging	59.3	38.1	46.4	46.9	48.9	47.9
MultiDecoder	56.9	45.2	50.4	34.7	53.4	42.1
ReHession	-	-	-	41.2	57.3	48.0
LSTM-CRF	-	-	-	69.3	31.0	42.8
SPTree	49.2	55.7	52.2	52.2	54.1	53.1
PA-LSTM-CRF	-	-	-	49.4	59.1	53.8
HRL	71.4	58.6	64.4	53.8	53.8	53.8
MTL	72.0	59.0	64.8	50.7	55.4	53.0
PMEI	**75.4**	**65.8**	**70.2**	**55.3**	**57.8**	**56.5**
MTL$_{BERT}$	**77.9**	67.8	72.5	**55.1**	57.3	56.2
PMEI$_{BERT}$	77.3	**69.7**	**73.3**	54.9	**58.9**	**56.8**

Table 4: Results on NYT10 and NYT11 with partial match (top) and exact match (bottom).

	NYT			NYT10		
Model	P	R	F1	P	R	F1
MTL$_{BERT}$	89.4	**89.9**	89.7	77.9	69.9	73.7
PMEI$_{BERT}$	**90.5**	89.8	**90.1**	**79.1**	**70.4**	**74.5**
MTL$_{BERT^*}$	73.6	67.6	70.4	60.1	46.2	52.3
PMEI$_{BERT^*}$	**80.3**	**68.8**	**74.1**	**67.2**	**48.4**	**56.2**

Table 5: Results on NYT and NYT10 with partial match. Models with fixed BERT parameters are marked "$*$".

Bias of BERT We notice that the PMEI significantly outperforms the MTL model, while the improvement of PMEI$_{BERT}$ over MTL$_{BERT}$ seems to be marginal especially on the NYT and NYT10 datasets. We believe that the inductive bias brought by the pre-trained BERT explains these results. To verify this assumption, we conduct an experiment on MTL$_{BERT}$ and PMEI$_{BERT}$ on NYT and NYT10 datasets where we freeze the parameters of BERT during training. Table 5 shows the results. We mark models with frozen BERT parameters with the symbol "$*$".

Considering the F1 performance in Table 5, we find that PMEI$_{BERT}$ surpasses MTL$_{BERT}$ only by 0.4% on NYT and 0.8% on NYT10. Meanwhile, PMEI$_{BERT^*}$ surpasses MTL$_{BERT^*}$ by a great margin, 3.7% on NYT and 3.9% on NYT10. The results suggest that when the pre-trained BERT is frozen, PMEI$_{BERT^*}$ has a better inductive bias relative to MTL$_{BERT^*}$ for the learning task. However, when further tuning is allowed in the pre-trained BERT, MTL$_{BERT}$ also has an inductive bias appropriate for the task, thus the advantage of PMEI$_{BERT}$ over MTL$_{BERT}$ becomes less significant.

	NYT			WebNLG		
Model	P	R	F1	P	R	F1
PMEI	**84.5**	**84.0**	**84.2**	**78.8**	**77.7**	**78.2**
MTL	77.4	76.4	76.9	76.7	74.8	75.7
MTL_{SM}	79.6	76.0	77.8	77.5	76.0	76.7
PMEI w/o SM	83.5	81.9	82.7	78.4	75.4	76.9
PMEI w/o interaction	79.7	77.8	78.7	78.0	77.5	77.7
PMEI w/o GRU	83.2	83.6	83.4	78.5	76.7	77.5

Table 6: Performance of ablated model architectures on the exact match task.

Ablation Study

We perform ablation studies to note the importance of several model components. Ablated models include **MTL**: directly passes H into the classifiers C_e and C_r for classification. **MTL_{SM}**: directly passes H through two independent stochastic maps to extract task-specific representations T_e and T_r. **PMEI w/o SM**: uses a deterministic map (i.e., GRUCell) instead of a stochastic map to extract T_e and T_r. **PMEI w/o interaction**: disregards explicit interactions between the tasks. **PMEI w/o GRU**: in this model GRUCell is replaced by an MLP to model μ and σ of the encoder $p(t_e|h, y'_e, y'_r)$.

We observe that PMEI w/o SM significantly surpasses PMEI w/o interaction on NYT, suggesting that the interaction between tasks on the NYT plays a more important role than the information control. On the other hand, PMEI w/o interaction outperforms PMEI w/o SM on WebNLG. Given the statistics shown in Table 2, it is easy to tell that the exact match task is indeed difficult on WebNLG as compared to NYT. This implies that the quality of Y'_e and Y'_r in WebNLG is significantly lower than NYT. Modeling explicit interactions with low quality predictions Y'_e and Y'_r will hurt performance, especially if we do not control the information flow from Y'_e and Y'_r to the task-specific representations T_e and T_r. We can also see that PMEI w/o interaction outperforms MTL_{SM}, suggesting the importance of leveraging early predictions to extract a better task-specific representation for classification. Lastly, we observe that PMEI w/o GRU underperforms PMEI, which suggests that employing the GRU can bring about performance improvement.

Impact of α, β_e, and β_r

To recall, α is a weight that controls the supervision loss of C'_e and C'_r, while β_e and β_r are weights to control the information flow of Y'_e and Y'_r. A high value for α means C'_e and C'_r are likely to overfit during training. A high value for β_e (or β_r) means we increase the flow of Y'_e (or Y'_r) to update the task-specific representation T_e (or T_r). In this experiment, $\beta = \beta_e = \beta_r$.

We investigate the impact of α and β on the performance of PMEI. Note that Y'_e and Y'_r are predictions of the classifiers C'_e and C'_r. Hence the quality of these predictions is subject to the classifiers' fitness, and may influence our model's performance. By controlling the supervision on C'_e and C'_r, we can control the fitness. In Figure 3, the sub-figure to the left (Figure 3(A)) shows the performance of our model on varying values of α, where β_e and β_r are fixed. The subfigure to the right (Figure 3(B)) shows the performance of our model on varying values of β, where the value of α is fixed.

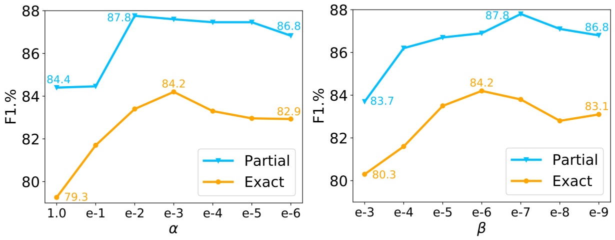


Figure 3: F1 performance curves of our model with different α (left) and β (right) values on the NYT dataset. $\beta = \beta_e = \beta_r$

We find that as we decrease the value of α or β, the performance of PMEI improves to a point, afterwards the performance deteriorates in a general sense. Specifically, in Figure 3(A), we find that PMEI achieves the best performance at $\alpha = e^{-2}$ on partial match and $\alpha = e^{-3}$ on exact match task for a fixed $\beta = e^{-7}$. Meanwhile, in Figure 3(B) PMEI achieves the best performance at $\beta = e^{-7}$ on partial match and $\beta = e^{-6}$ on exact match task for a fixed $\alpha = e^{-2}$. In particular, the results with varying values for β shows that it is important to control the flow of early predictions.

Conclusion

We build a multitask learning model for the joint entity and relation extraction task. The core of our model is the way we learn representations for the data to be classified, which is typically a core task in every supervised learning framework. In this paper, we acknowledge the correlations that exist between the outputs of the related tasks, and exploit these correlations through the interaction of early predictions in the individual tasks. Previous works have considered this approach to improve representation learning, but they do so by passing these early predictions, as well as the input representation through a deterministic map. In our approach, we consider a stochastic map as a natural way to capture the task-specific representations. Meanwhile, we control the information flow of early predictions to ensure that good task-specific representations can be extracted for supervision. In this way, we progressively make predictions on the individual tasks. Extensive experiments on several benchmark datasets show the effectiveness of our approach.

Acknowledgments

This work is supported partly by the National Natural Science Foundation of China (No. 61772059, 61421003), by the Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), by the Fundamental Research Funds for the Central Universities, by the Beijing S&T Committee (No. Z191100008619007) and by the State Key Laboratory of Software Development Environment (No. SKLSDE-2020ZX-14).

References

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2017. Deep Variational Information Bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL https://openreview.net/forum?id=HyxQzBceg.

Collobert, R.; and Weston, J. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, 160–167. doi:10.1145/1390156.1390177. URL https://doi.org/10.1145/1390156.1390177.

Dai, D.; Xiao, X.; Lyu, Y.; Dou, S.; She, Q.; and Wang, H. 2019. Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 6300–6308. doi:10.1609/aaai.v33i01.33016300. URL https://doi.org/10.1609/aaai.v33i01.33016300.

Dankers, V.; Rei, M.; Lewis, M.; and Shutova, E. 2019. Modelling the interplay of metaphor and emotion through multitask learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2218–2229. doi:10.18653/v1/D19-1227. URL https://doi.org/10.18653/v1/D19-1227.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. URL https://www.aclweb.org/anthology/N19-1423/.

Fu, T.; Li, P.; and Ma, W. 2019. GraphRel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 1409–1418. URL https://www.aclweb.org/anthology/P19-1136/.

Gardent, C.; Shimorina, A.; Narayan, S.; and Perez-Beltrachini, L. 2017. Creating Training Corpora for NLG Micro-Planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 179–188. doi:10.18653/v1/P17-1017. URL https://doi.org/10.18653/v1/P17-1017.

Gupta, P.; Schütze, H.; and Andrassy, B. 2016. Table Filling Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, 2537–2547. URL https://www.aclweb.org/anthology/C16-1239/.

He, R.; Lee, W. S.; and and‘ Daniel Dahlmeier, H. T. N. 2019. An Interactive Multi-Task Learning Network for End-to-End Aspect-Based Sentiment Analysis. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 504–515. URL https://www.aclweb.org/anthology/P19-1048/.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L. S.; and Weld, D. S. 2011. Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, 541–550. URL https://www.aclweb.org/anthology/P11-1055/.

Hu, M.; Peng, Y.; Huang, Z.; Li, D.; and Lv, Y. 2019. Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 537–546. doi:10.18653/v1/p19-1051. URL https://doi.org/10.18653/v1/p19-1051.

Lan, M.; Wang, J.; Wu, Y.; Niu, Z.; and Wang, H. 2017. Multi-task Attention-based Neural Networks for Implicit Discourse Relationship Representation and Identification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 1299–1308. doi:10.18653/v1/d17-1134. URL https://doi.org/10.18653/v1/d17-1134.

Liu, L.; Ren, X.; Zhu, Q.; Zhi, S.; Gui, H.; Ji, H.; and Han, J. 2017. Heterogeneous Supervision for Relation Extraction: A Representation Learning Approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 46–56. doi:10.18653/v1/d17-1005. URL https://doi.org/10.18653/v1/d17-1005.

Liu, P.; Fu, J.; Dong, Y.; Qiu, X.; and Cheung, J. C. K. 2019a. Learning Multi-Task Communication with Message Passing for Sequence Learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 4360–4367. doi:10.1609/aaai.v33i01.33014360. URL https://doi.org/10.1609/aaai.v33i01.33014360.

Liu, P.; Qiu, X.; and Huang, X. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1–10. doi:10.18653/v1/P17-1001. URL https://doi.org/10.18653/v1/P17-1001.

Liu, X.; He, P.; Chen, W.; and Gao, J. 2019b. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 4487–4496. doi:10.18653/v1/p19-1441. URL https://doi.org/10.18653/v1/p19-1441.

Miwa, M.; and Bansal, M. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. URL https://www.aclweb.org/anthology/P16-1105/.

Miwa, M.; and Sasaki, Y. 2014. Modeling Joint Entity and Relation Extraction with Table Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1858–1869. URL https://www.aclweb.org/anthology/D14-1200/.

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

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling Relations and Their Mentions without Labeled Text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, 148–163. doi:10.1007/978-3-642-15939-8_10. URL https://doi.org/10.1007/978-3-642-15939-8_10.

Sun, T.; Shao, Y.; Li, X.; Liu, P.; Yan, H.; Qiu, X.; and Huang, X. 2020. Learning Sparse Sharing Architectures for Multiple Tasks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 8936–8943. URL https://aaai.org/ojs/index.php/AAAI/article/view/6424.

Takanobu, R.; Zhang, T.; Liu, J.; and Huang, M. 2019. A Hierarchical Framework for Relation Extraction with Reinforcement Learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 7072–7079. doi:10.1609/aaai.v33i01.33017072. URL https://doi.org/10.1609/aaai.v33i01.33017072.

Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; and Chang, Y. 2020. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 1476–1488. URL https://www.aclweb.org/anthology/2020.acl-main.136/.

Zelenko, D.; Aone, C.; and Richardella, A. 2003. Kernel Methods for Relation Extraction. *J. Mach. Learn. Res.* 3: 1083–1106. URL http://jmlr.org/papers/v3/zelenko03a.html.

Zeng, D.; Zhang, H.; and Liu, Q. 2020. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 9507–9514. URL https://aaai.org/ojs/index.php/AAAI/article/view/6495.

Zeng, X.; He, S.; Zeng, D.; Liu, K.; Liu, S.; and Zhao, J. 2019. Learning the Extraction Order of Multiple Relational Facts in a Sentence with Reinforcement Learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 367–377. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1035. URL https://www.aclweb.org/anthology/D19-1035.

Zeng, X.; Zeng, D.; He, S.; Liu, K.; and Zhao, J. 2018. Extracting Relational Facts by an End-to-End Neural Model with Copy Mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 506–514. doi:10.18653/v1/P18-1047. URL https://www.aclweb.org/anthology/P18-1047/.

Zhao, S.; Liu, T.; Zhao, S.; and Wang, F. 2019. A Neural Multi-Task Learning Framework to Jointly Model Medical Named Entity Recognition and Normalization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 817–824. doi:10.1609/aaai.v33i01.3301817. URL https://doi.org/10.1609/aaai.v33i01.3301817.

Zheng, S.; Hao, Y.; Lu, D.; Bao, H.; Xu, J.; Hao, H.; and Xu, B. 2017a. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing* 257: 59–66. doi:10.1016/j.neucom.2016.12.075. URL https://doi.org/10.1016/j.neucom.2016.12.075.

Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; and Xu, B. 2017b. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1227–1236. doi:10.18653/v1/P17-1113. URL https://doi.org/10.18653/v1/P17-1113.