Extracting Entities and Relations with Joint Minimum Risk Training

Changzhi Sun[2], Yuanbin Wu[1, 2], Man Lan[1, 2], Shiliang Sun[2],
Wenting Wang[3], Kuang-Chih Lee[3], and Kewen Wu[3]

[1]Shanghai Key Laboratory of Multidimensional Information Processing
[2]Department of Computer Science and Technology, East China Normal University
[3]Alibaba Group
{changzhisun}@stu.ecnu.edu.cn
{ybwu,mlan,slsun}@cs.ecnu.edu.cn
{nentiao.wwt,kuang-chih.lee,kewen.wukw}@alibaba-inc.com

Abstract

We investigate the task of joint entity relation extraction. Unlike prior efforts, we propose a new lightweight joint learning paradigm based on minimum risk training (MRT). Specifically, our algorithm optimizes a global loss function which is flexible and effective to explore interactions between the entity model and the relation model. We implement a strong and simple neural network where the MRT is executed. Experiment results on the benchmark ACE05 and NYT datasets show that our model is able to achieve state-of-the-art joint extraction performances.

1 Introduction

Detecting entities and relations is usually the first step towards extracting structured knowledge from plain texts. Its goal is to identify text spans representing typed objects (*entities*) and semantic relations among those text spans (*relations*). For example, in the following sentence,

> [Associated Press]$_{\text{ORG}}$ [writer]$_{\text{PER}}$ [Patrick McDowell]$_{\text{PER}}$ in [Kuwait City]$_{\text{GPE}}$.

"Associate Press" is an organization entity (ORG), "writer" is a person entity (PER), and the two entities have an affiliation relation (ORG-AFF).

Two types of models have been applied to the extraction task, the pipeline model and the joint model. In the pipeline setting, the task is broken down into independently learned components (an entity model and a relation model). Despite its flexibility, the pipeline ignores interactions between the two models. For example, the entity model doesn't look at relation annotations which are useful for identifying entities (e.g., if an ORG-AFF relation exists, the entity model can only assign ORG and AFF to its entities). The joint setting, on the other hand, extracts entities and relations in a unified model, which can explore shared information and alleviate error propagations between models. Here we will focus on joint models.

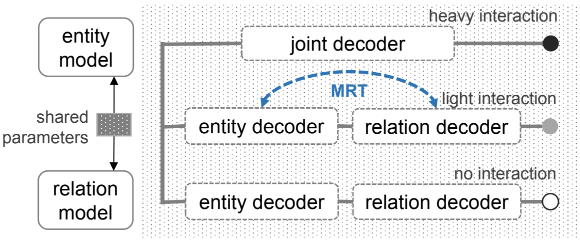

Figure 1: Paradigms of joint entity relation extraction.

One simple joint learning paradigm is through sharing parameters (Miwa and Bansal, 2016; Katiyar and Cardie, 2017). Typically, instead of training two independent models, the entity and relation model can share some input features or internal hidden states. It has an advantage that no additional constraint is required on the two sub-models. But the connections among sub-models are still not fully explored due to independent sub-model decoders. For example, to get signals from relation annotations, the entity model needs to wait for the relation model to update the shared parameters. To further utilize the interaction between decoders, some complex joint decoding algorithms (e.g., simultaneously decoding entities and relations in beam search) have been carefully studied (Li and Ji, 2014; Katiyar and Cardie, 2016; Zhang et al., 2017; Zheng et al., 2017). In this paradigm, it is important (and hard) to make a good balance between the exactness of the joint decoding algorithm and capacities of individual sub-models.

In this work, we propose a joint minimum risk training (MRT) (Och, 2003; Smith and Eisner, 2006) method for the entity and relation extraction

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2256–2265
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

task. It provides a lightweight way to strengthen connections between the entity model and the relation model, and keeps their capacities unaffected. Given an input $\mathbf{x}$ and a loss function $\Delta(\hat{\mathbf{y}}, \mathbf{y})$ (measuring the difference between model output $\hat{\mathbf{y}}$ and the true $\mathbf{y}$), MRT seeks a posterior $P(\hat{\mathbf{y}}|\mathbf{x})$ to minimize the expected loss $\mathbf{E}_{\hat{\mathbf{y}} \sim P(\hat{\mathbf{y}}|\mathbf{x})} \Delta(\hat{\mathbf{y}}, \mathbf{y})$. Comparing with prior joint decoding algorithms, the MRT-based algorithm is simple and can be applied to a broad range of entity relation extraction models without changing the original sub-models and decoders (Figure 1).

One advantage of the MRT-based method is that it can explicitly optimize a global sentence-level loss (e.g., F1 score) rather than local token-level losses. Therefore, it may catch more sentence-level information in the training time and match evaluation metrics better in the testing time. Furthermore, besides the handcrafted losses, we also try to directly learn a loss function from data during the joint MRT process. The automatically obtained loss would help MRT to calibrate its risk estimation with knowledge from the data distribution. On the other hand, comparing with preivous single task MRT, the joint MRT algorithm here will integrate messages from different sub-models, which is the key step for enhancing decoder interactions in the joint learning. As a result, the training of the entity model now can directly acknowledge the loss of the relation model (without waiting for shared parameters) and vice versa.

We compile the proposed joint MRT with a strong neural network-based model which uses recurrent neural networks (RNN) in the entity model and convolutional neural networks (CNN) in the relation model. On benchmark ACE05 and NYT datasets, we show that the new RNN+CNN structure outperforms previous neural network-based models. After adding the joint MRT, our model is able to achieve state-of-the-art performances.

To summarize, our main contributions include

1. proposing a new joint learning paradigm based on minimum risk training for the joint entity relation extraction task.

2. implementing a strong and simple neural-network-based entity relation extraction model which carries the proposed MRT algorithm. [1]

3. achieving state-of-the-art results on two benchmark datasets (ACE05 and NYT).

2 Related Work

In many pipelined entity relation extraction systems, one first learns an entity model, then learns a relation model based on entities generated by the entity model (Miwa et al., 2009; Chan and Roth, 2011; Lin et al., 2016). Such systems are often flexible to incorporate different data sources and different learning algorithms. However, they may also suffer from error propagation and data inefficiency. To tackle the problem, many recent studies try to develop joint extraction algorithms.

Parameter sharing is a basic strategy in joint learning paradigms. For example, in (Miwa and Bansal, 2016), the entity model is a sentence-level RNN, and the relation model is a dependency tree path RNN which takes hidden states of the entity model as features (i.e., the shared parameters). Our basic extraction model is similar to theirs but with a CNN-based relation model. Similarly, Katiyar and Cardie (2017) build a simplified relation model on the entity RNN using the attention mechanism.

To further explore interactions between the entity decoder and the relation decoder, some joint decoding algorithms were studied. For example, Katiyar and Cardie (2016) propose a CRF-based model which conducts joint decoding with augmented transition matrices. Zheng et al. (2017) propose to directly encode relations in the sequential labelling tag set. Both of them are exact decoding algorithms, but they need adding constraints on the relation model (e.g., Zheng et al. (2017) cannot handle entities which appear in multiple relations). On the other side, Li and Ji (2014) develop a joint decoding algorithm based on beam search. Zhang et al. (2017) study a globally normalized joint model. They retain capacities of sub-models, while their decoding algorithms are inexact. Here, we introduce MRT to the task, which is a more lightweight setting of joint learning.

Minimum risk training is a learning framework which tries to handle models with arbitrary discrepancy metrics (i.e., losses of a model output w.r.t. the true answer) (Och, 2003; Smith and Eisner, 2006; Gimpel and Smith, 2010). It has been successfully applied to many NLP tasks. Some recent work include (He and Deng, 2012; Shen et al., 2016) which apply MRT to (neural) ma-

[1] Our implementation is available at https://github.com/changzhisun/entrel-joint-mrt.

chine translation, (Xu et al., 2016) which develops a shift-reduce CCG parser to directly optimize F1, and (Ayana et al., 2016) which uses a MRT-based model for summarization. We note that most previous applications of MRT focus on a single job, while the joint entity relation extraction consists of two sub-tasks. Investigating MRT in joint learning scenarios is the main topic of this work.

Finally, the sampling algorithm of solving MRT is similar to the policy gradient algorithm in reinforcement learning (RL) (Sutton and Barto, 1998). Some recent NLP applications which share the key idea of MRT but are described with RL language also show promising results (e.g., dialog systems (Li et al., 2016), machine translation (Nguyen et al., 2017)). The idea of learning loss functions from data is similar to inverse reinforcement learning (Abbeel and Ng, 2004; Ratliff et al., 2006).

3 The Approach

We define the joint entity and relation extraction task following the setting of (Miwa and Bansal, 2016). Given an input sentence $s = w_1, \dots, w_{|s|}$ (w_i is a word), the task is to extract a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$. An entity $e \in \mathcal{E}$ is a sequence of words labelling with an entity type (e.g., person (PER), organization (ORG)). Let $\mathcal{T}_e$ be the set of possible entity types. A relation r is a triple (e_1, e_2, l), where e_1 and e_2 are two entities, l is a relation type describing the semantic relation between e_1 and e_2 (e.g., organization affiliation relation (ORG-AFF)). Let $\mathcal{T}_r$ be the set of possible relation types.

In our joint extraction method (Figure 2), we treat entity detection as a sequence labelling task (Section 3.1) and relation detection as a classification task (Section 3.2). Models of the two tasks share parameters and are trained jointly. Departing from previous joint learning algorithms (Miwa and Bansal, 2016; Katiyar and Cardie, 2017; Zhang et al., 2017), we introduce minimum risk training to the joint extraction model. It optimizes a global loss function and bridges the discrepancy between training and testing (Section 3.3).

3.1 Entity Detection

To represent entities in s, we assign a tag t_i to each word w_i following the BILOU tagging scheme: t_i takes a value in $\{\texttt{B-}*, \texttt{I-}*, \texttt{L-}*, \texttt{O}, \texttt{U-}*\}$, where B, I, L and O denote the begin, inside, end and outside of an entity, U denotes a single word entity and $* \in \mathcal{T}_e$ represents different entity types. For example, for a person (PER) entity "Patrick McDowell", we assign B-PER to "Patrick" and L-PER to "McDowell". Given an input sentence s, the entity model predicts the tags of words $\hat{\mathbf{t}} = \hat{t}_1, \hat{t}_2, \dots, \hat{t}_{|s|}$ by learning from the true tags $\mathbf{t} = t_1, t_2, \dots, t_{|s|}$.

We use a bidirectional long short term memory (bi-LSTM) network (Hochreiter and Schmidhuber, 1997) to solve the sequence labelling task. At each sentence position i, a forward LSTM chain computes a hidden state vector $\vec{\mathbf{h}}_i$ by recursively collecting information from the beginning of s to the current position i. Similarly, a backward LSTM chain collects information $\overleftarrow{\mathbf{h}}_i$ from the end of s to the position i.

$$\vec{\mathbf{h}}_i = \text{LSTM}(\mathbf{x}_i, \vec{\mathbf{h}}_{i-1}; \vec{\boldsymbol{\theta}}),$$
$$\overleftarrow{\mathbf{h}}_i = \text{LSTM}(\mathbf{x}_i, \overleftarrow{\mathbf{h}}_{i+1}; \overleftarrow{\boldsymbol{\theta}}).$$

The word representation $\mathbf{x}_i$ of w_i has two parts $\mathbf{x}_i = \mathbf{w}_i \oplus \mathbf{c}_i$ ($\oplus$ is the vector concatenation). $\mathbf{w}_i$ is a word embedding of word w_i (from an embedding look-up table $\mathbf{W}_e$). $\mathbf{c}_i$ is a character-based representation of w_i which is obtained by running a convolution neural network on the character sequence of w_i: $\mathbf{c}_i = \text{CNN}(\text{char}(w_i); \boldsymbol{\theta}_c)$.

To predict the tag $\hat{t}_i$, we combine the forward and the backward hidden vector to $\mathbf{h}_i = \vec{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i$, and apply a softmax function on $\mathbf{h}_i$ to get the posterior of $\hat{t}_i$,

$$P_{\text{ent}}(\hat{t}_i | s; \boldsymbol{\theta}_E) = \text{Softmax}(\mathbf{W}_E \cdot \mathbf{h}_i), \quad (1)$$

where $\boldsymbol{\theta}_E = \{\mathbf{W}_e, \boldsymbol{\theta}_c, \vec{\boldsymbol{\theta}}, \overleftarrow{\boldsymbol{\theta}}, \mathbf{W}_E\}$ are parameters of the entity model. Given an input sentence s and its ground truth tag sequence $\mathbf{t}$, the training objective is to minimize $\mathcal{L}_{\text{ent}}$, [2]

$$\mathcal{L}_{\text{ent}}(\boldsymbol{\theta}_E) = -\frac{1}{|s|} \sum_{i=1}^{|s|} \log P_{\text{ent}}(\hat{t}_i = t_i | s; \boldsymbol{\theta}_E).$$

3.2 Relation Detection

Given a set of detected entities $\hat{\mathcal{E}}$ (obtaining from the entity tag sequence $\hat{\mathbf{t}}$), we consider all entity pairs in $\hat{\mathcal{E}}$ as candidate relations. The task of relation detection is to predict a relation type $l \in \mathcal{T}_r$ for each pair, [3] and output a relation set

[2] We have also tried biLSTM-CRF (Huang et al., 2015) as an advanced entity model, but performances are nearly the same in our experiments.

[3] We include a NONE relation type in $\mathcal{T}_r$ which means that there exists no relation between e_1 and e_2.

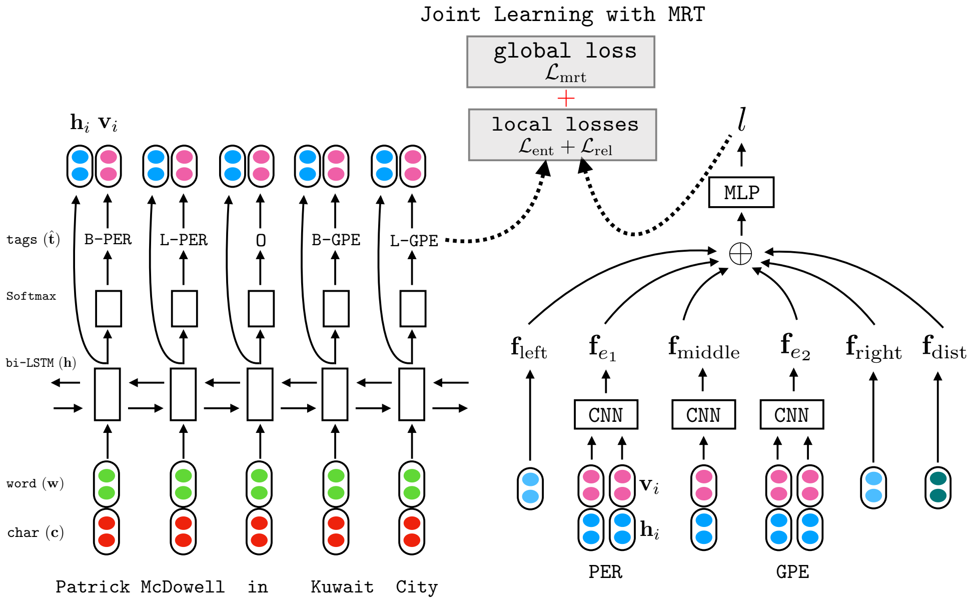
Joint Learning with MRT
global loss
$\mathcal{L}_{\text{mrt}}$
+
local losses
$\mathcal{L}_{\text{ent}} + \mathcal{L}_{\text{rel}}$
l
MLP
$\oplus$
$\mathbf{h}_i$ $\mathbf{v}_i$
tags ($\hat{\mathbf{t}}$)
B-PER
L-PER
O
B-GPE
L-GPE
Softmax
bi-LSTM ($\mathbf{h}$)
word ($\mathbf{w}$)
char ($\mathbf{c}$)
Patrick McDowell in Kuwait City
$\mathbf{f}_{\text{left}}$
$\mathbf{f}_{e_1}$
$\mathbf{f}_{\text{middle}}$
$\mathbf{f}_{e_2}$
$\mathbf{f}_{\text{right}}$
$\mathbf{f}_{\text{dist}}$
CNN
CNN
CNN
$\mathbf{v}_i$
$\mathbf{h}_i$
PER
GPE

Figure 2: Our network structure for the joint entity and relation extraction.

$\hat{\mathcal{R}} = \{(e_1, e_2, l) | e_1, e_2 \in \hat{\mathcal{E}}, e_1 \neq e_2, l \in \mathcal{T}_r\}$. To build the relation model, we extract two types of features, namely, features regarding words in e_1, e_2 and features regarding contexts of the entity pair (e_1, e_2).

To extract features on words in e_1, e_2, we use two convolutional neural networks. Taking e_1 as an example, for each word w_i in e_1, we first collect w_i's bi-LSTM hidden vector $\mathbf{h}_i$ from the entity model. Then, we concatenate $\mathbf{h}_i$ with a one-hot entity tag representation $\mathbf{v}_i$ of $\hat{t}_i$. We build a feature vector $\mathbf{f}_{e_1}$ for e_1 by running a CNN (a single convolution layer with a max-pooling layer) on vectors $\{\mathbf{h}_i \oplus \mathbf{v}_i | w_i \in e_1\}$. Similarly, we build $\mathbf{f}_{e_2}$ for e_2 with another CNN.

For context features of the entity pair (e_1, e_2), we build three feature vectors by looking at words between e_1 and e_2 ($\mathbf{f}_{\text{middle}}$), words on the left of the pair ($\mathbf{f}_{\text{left}}$) and words on the right of the pair ($\mathbf{f}_{\text{right}}$). For $\mathbf{f}_{\text{middle}}$, we run a CNN on words between e_1 and e_2 like the case of $\mathbf{f}_{e_1}, \mathbf{f}_{e_2}$. For $\mathbf{f}_{\text{left}}$ and $\mathbf{f}_{\text{right}}$, we use the "LSTM-Minus" method as (Wang and Chang, 2016; Zhang et al., 2017). Assume that the left context of (e_1, e_2) is from sentence position 0 to i, then $\mathbf{f}_{\text{left}} = \overrightarrow{\mathbf{h}}_i \oplus (\overleftarrow{\mathbf{h}}_0 - \overleftarrow{\mathbf{h}}_{i+1})$. Similarly, if the right context of (e_1, e_2) is from j to $|s| - 1$, then $\mathbf{f}_{\text{right}} = (\overrightarrow{\mathbf{h}}_{|s|-1} - \overrightarrow{\mathbf{h}}_{j-1}) \oplus \overleftarrow{\mathbf{h}}_j$. We also use a one-hot feature $\mathbf{f}_{\text{dist}}$ to describe the distance between e_1 and e_2 in the sentence.

Finally, $\mathbf{f}_{e_1}$, $\mathbf{f}_{e_2}$, $\mathbf{f}_{\text{middle}}$, $\mathbf{f}_{\text{left}}$, $\mathbf{f}_{\text{right}}$ and $\mathbf{f}_{\text{dist}}$ are concatenated to a single vector $\mathbf{f}_{e_1,e_2}$. To get the posterior of the relation type $\hat{l}$, we apply a multilayer perceptron with one hidden layer on $\mathbf{f}_{e_1,e_2}$,

$$\begin{aligned} & \mathrm{P}_{\text{rel}}(\hat{l} | s, e_1, e_2; \boldsymbol{\theta}_R) \\ & = \mathrm{Softmax}(\mathbf{W}_2 \cdot \mathrm{ReLU}(\mathbf{W}_1 \cdot \mathbf{f}_{e_1,e_2})), \quad (2) \end{aligned}$$

where $\boldsymbol{\theta}_R = \{\boldsymbol{\theta}_{e_1}, \boldsymbol{\theta}_{e_2}, \boldsymbol{\theta}_{\text{middle}}, \mathbf{W}_1, \mathbf{W}_2\}$ contains parameters of the relation model (shared parameters with the entity model are omitted).

Given an input sentence s, the training objective is to minimize

$$\mathcal{L}_{\text{rel}}(\boldsymbol{\theta}_R) = - \sum_{\substack{e_1, e_2 \in \hat{\mathcal{E}} \\ e_1 \neq e_2}} \frac{\log P_{\text{rel}}(\hat{l} = l | s, e_1, e_2; \boldsymbol{\theta}_R)}{|\hat{\mathcal{E}}|(|\hat{\mathcal{E}}| - 1)},$$

where the true label l of a candidate entity pair (e_1, e_2) can be read from true annotations.

3.3 Joint Minimum Risk Training

To jointly learn the entity model and the relation model, one common strategy is to optimize the combined objective function $\widetilde{\mathcal{L}} = \mathcal{L}_{\text{ent}} + \mathcal{L}_{\text{rel}}$, where the joint learning is accomplished by the shared parameters. However, we would think that $\widetilde{L}$ optimizes a "local" loss by observing that a) in both $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{rel}}$, the loss functions are calculated by only looking at local parts. For example, the loss in $\mathcal{L}_{\text{ent}}$ is based on the correctness of local entity tags t_i rather than a global measurement (e.g., F1 score of extracted entities), b) both the entity model and the relation model are unaware of the loss from the other side. For example, the entity model needs to wait for the relation model to update the shared parameters rather than get direct supervision from the loss of the relation model.

Here we introduce the minimum risk training framework to the joint model. Comparing with optimizing the local loss in $\widetilde{\mathcal{L}}$, the joint MRT will optimize a global loss and provide a tighter connection between the entity decoder and the relation decoder. To illustrate the algorithm, we first aggregate some notations.

Let $\mathbf{y} \triangleq (\mathcal{E}, \mathcal{R})$ contain the ground truth entity tag sequence and relations, $\hat{\mathbf{y}} \triangleq (\hat{\mathcal{E}}, \hat{\mathcal{R}})$ contain outputs of the joint extraction model and $\mathcal{Y}(s)$ be the set of all possible outputs of the input sentence s ($\mathbf{y}, \hat{\mathbf{y}} \in \mathcal{Y}(s)$). We define the joint probability,

$$\begin{aligned} &P(\hat{\mathbf{y}}|s;\boldsymbol{\theta}) = P(\hat{\mathcal{E}}|s;\boldsymbol{\theta}_E)P(\hat{\mathcal{R}}|s,\hat{\mathcal{E}};\boldsymbol{\theta}_R) \\ &= \prod_i P_{\text{ent}}(\hat{t}_i|s;\boldsymbol{\theta}_E) \prod_{\substack{e_1,e_2\in\hat{\mathcal{E}} \\ e_1\neq e_2}} P_{\text{rel}}(\hat{l}|s,e_1,e_2;\boldsymbol{\theta}_R), \end{aligned}$$

where $\boldsymbol{\theta} = \boldsymbol{\theta}_E \bigcup \boldsymbol{\theta}_R$ is the joint model parameter, and $P_{\text{ent}}, P_{\text{rel}}$ are in Equation 1 and 2.

The objective of MRT is to minimize the following expected loss (i.e., *risk*),

$$\mathbf{E}_{\hat{\mathbf{y}}\sim P(\hat{\mathbf{y}}|s;\boldsymbol{\theta})}\Delta(\hat{\mathbf{y}},\mathbf{y}) = \sum_{\hat{\mathbf{y}}\in\mathcal{Y}(s)} P(\hat{\mathbf{y}}|s;\boldsymbol{\theta})\Delta(\hat{\mathbf{y}},\mathbf{y}), \tag{3}$$

where $\Delta(\hat{\mathbf{y}},\mathbf{y})$ is a (arbitrary) loss function describing the difference between $\hat{\mathbf{y}}$ and $\mathbf{y}$.

In our model, the loss function $\Delta(\hat{\mathbf{y}},\mathbf{y})$ is the key factor to enhance the joint extraction performances. First, in $\Delta(\hat{\mathbf{y}},\mathbf{y})$, we consider sentence-level F1 scores of entity and relation extraction results (denoted by $\mathrm{F}_{\text{ent}}(\hat{\mathcal{E}},\mathcal{E})$, $\mathrm{F}_{\text{rel}}(\hat{\mathcal{R}},\mathcal{R})$). Specifically, we use $1-\mathrm{F}_{\text{ent}}(\hat{\mathcal{E}},\mathcal{E})$ and $1-\mathrm{F}_{\text{rel}}(\hat{\mathcal{R}},\mathcal{R})$ as the metric of the entity loss and the relation loss respectively. On the one hand, F1 scores characterize the overall performance of the outputs and make the training objective be consistent with the testing time evaluation metric. On the other hand, F1 scores cannot be decomposed onto local predictions of $\hat{\mathcal{E}}$ and $\hat{\mathcal{R}}$ like the log losses in $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{rel}}$, thus we need a different training algorithm.

Second, different from previous applications of MRT on single tasks (Xu et al., 2016; Shen et al., 2016), we have two sources of losses in the joint extraction. By integrating losses of individual tasks in the learning algorithm, the entity model could forecast how plausible a candidate entity is according to the relation model, and the relation model could also know the confidence of the entity extraction results. Here, we define a global loss by adding losses of the two models,

$$\Delta_{E+R}(\hat{\mathbf{y}},\mathbf{y}) = 1 - \frac{1}{2}[\mathrm{F}_{\text{ent}}(\hat{\mathcal{E}},\mathcal{E}) + \mathrm{F}_{\text{rel}}(\hat{\mathcal{R}},\mathcal{R})].$$

Algorithm 1 The Sampling Algorithm

```
Input: Entity model $\boldsymbol{\theta}_E$, relation model $\boldsymbol{\theta}_R$, sentence $s$, the
    sample size $K$
Output: A subset $\mathcal{Y}'(s)$ of $\mathcal{Y}(s)$
$\mathcal{Y}'(s) \leftarrow \{(\mathcal{E},\mathcal{R})\}$ // add the ground truth
while $|\mathcal{Y}'(s)| \leq K$ do
    $i \leftarrow 1$
    while $i \leq |s|$ do
        with prob. 0.9, sample $t'_i \sim P_{\text{ent}}(\cdot|s;\boldsymbol{\theta}_E)$
        with prob. 0.1, sample $t'_i$ uniformly
        $i \leftarrow i+1$
    end while
    $\mathcal{E}' \leftarrow \mathbf{t}' = t'_1, t'_2, \cdots, t'_{|s|}$
    $\mathcal{R}' \leftarrow \emptyset$
    for $e_1, e_2 \in \hat{\mathcal{E}}, e_1 \neq e_2$ do
        sample $l' \sim P_{\text{rel}}(\cdot|s,e_1,e_2;\boldsymbol{\theta}_R)$
        $\mathcal{R}' \leftarrow \mathcal{R}' \cup \{(e_1,e_2,l')\}$
    end for
    $\mathcal{Y}'(s) \leftarrow \mathcal{Y}'(s) \cup \{(\mathcal{E}',\mathcal{R}')\}$
end while
```

To compare with Δ_{E+R}, we also try two alternatives of $\Delta(\hat{\mathbf{y}},\mathbf{y})$ in experiments, namely, $\Delta_E(\hat{\mathbf{y}},\mathbf{y}) = 1-\mathrm{F}_{\text{ent}}(\hat{\mathcal{E}},\mathcal{E})$ and $\Delta_R(\hat{\mathbf{y}},\mathbf{y}) = 1-\mathrm{F}_{\text{rel}}(\hat{\mathcal{R}},\mathcal{R})$. They only look one model's loss.

Third, in addition to handcrafted loss functions, we further ask whether the joint MRT model could benefit from automatic "loss engineering". Specifically, let $\Gamma(\hat{\mathbf{y}})$ be the loss learned from the training set, we augment $\Delta(\hat{\mathbf{y}},\mathbf{y})$ of the MRT objective with $\Gamma(\hat{\mathbf{y}})$, and require the learning process to assign a smaller Γ value (with a margin) to the ground truth output $\mathbf{y}$ than other $\hat{\mathbf{y}} \in Y\backslash\{\mathbf{y}\}$,

$$\begin{aligned} &\min. \sum_{\hat{\mathbf{y}}\in\mathcal{Y}(s)} P(\hat{\mathbf{y}}|s;\boldsymbol{\theta})\left(\Delta(\hat{\mathbf{y}},\mathbf{y}) + \Gamma(\hat{\mathbf{y}})\right) + \xi \\ &\text{s.t.} \quad \Gamma(\mathbf{y}^*) - \Gamma(\mathbf{y}) \geq 1-\xi,\ \xi \geq 0, \end{aligned} \tag{4}$$

where $\mathbf{y}^* = \arg\min_{\hat{y}\in Y(s)} \Gamma(\hat{\mathbf{y}})$. Here, we simply set $\Gamma(\hat{\mathbf{y}}) = 1 - P(\hat{\mathbf{y}}|s;\boldsymbol{\theta})$ [4] and reformulate above objective as

$$\begin{aligned} &\sum_{\hat{\mathbf{y}}\in\mathcal{Y}(s)} P(\hat{\mathbf{y}}|s;\boldsymbol{\theta})\left(\Delta(\hat{\mathbf{y}},\mathbf{y}) - P(\hat{\mathbf{y}}|s;\boldsymbol{\theta})\right) \\ &\quad + [1 - P(\mathbf{y}|s;\boldsymbol{\theta}) + P(\mathbf{y}^*|s;\boldsymbol{\theta})]_+ . \end{aligned} \tag{5}$$

where $[u]_+ = \max(u,0)$ is the hinge loss.

Optimizing the expected loss is hard since the size of $\mathcal{Y}(s)$ is exponential. In practice, we could

[4] Further study on different $\Gamma(\hat{\mathbf{y}})$ is left for future work.

approximate the expectation in Equation 3 by sampling a tractable subset $\mathcal{Y}'(s)$ of $\mathcal{Y}(s)$. Specifically, we first obtain an entity set $\mathcal{E}'$ by sampling (without replacement) an entity tag sequence $\mathbf{t}'$ from P_{ent}. [5] Then based on the sampled entities, we get a relation set $\mathcal{R}'$ by sampling l' from P_{rel} for each entity pairs. Algorithm 1 lists the pseudo code. [6] In experiments, we also try a variant of Algorithm 1 which only samples from the entity model, and selects relation labels with the maximum posterior (i.e., doesn't sample relations).

With the sampled subset $\mathcal{Y}'(s)$, we consider a revised version of the original MRT objective,

$$\mathcal{L}_{\text{mrt}}(\boldsymbol{\theta}) = \sum_{\hat{\mathbf{y}} \in \mathcal{Y}'(s)} Q(\hat{\mathbf{y}}|s;\boldsymbol{\theta},\mu,\alpha)\Delta(\hat{\mathbf{y}},\mathbf{y}), \quad (6)$$

where $Q(\hat{\mathbf{y}}|s;\boldsymbol{\theta},\mu,\alpha)$ is a re-normalization of $P(\hat{\mathbf{y}}|s;\boldsymbol{\theta})$ on the subset $\mathcal{Y}'(s)$, [7]

$$Q(\hat{\mathbf{y}}|s;\boldsymbol{\theta},\mu,\alpha) = \tfrac{1}{Z}[P(\hat{\mathcal{E}}|s,\boldsymbol{\theta}_E)^{\mu}P(\hat{\mathcal{R}}|s,\hat{\mathcal{E}},\boldsymbol{\theta}_R)^{1-\mu}]^{\alpha}$$

$$Z = \sum_{(\mathcal{E}',\mathcal{R}') \in \mathcal{Y}'(s)} [P(\mathcal{E}'|s,\boldsymbol{\theta}_E)^{\mu}P(\mathcal{R}'|s,\mathcal{E}',\boldsymbol{\theta}_R)^{1-\mu}]^{\alpha}$$

The hyper-parameter α controls the sharpness of the Q distribution (Och, 2003), and μ weights the importance of the entity model and the relation model in Q. Similary, we can rewrite the objective in Equation 5 with $\mathcal{Y}'(s)$ and Q.

Finally, we remark that if we view MRT as a fine tuning step, it can be applied in any joint learning model based on building the joint distribution $P(\hat{\mathbf{y}}|s,\boldsymbol{\theta})$ (e.g., the globally normalized P in (Zhang et al., 2017)). Thus, we would think that MRT is a flexible and lightweight framework for the joint learning.

3.4 Training

To train the joint extraction model, we first pretrain the model with objective $\widetilde{\mathcal{L}}$ (i.e., minimize the local loss), then optimize the local loss and the global loss simultaneously with objective $\widetilde{\mathcal{L}} + \mathcal{L}_{mrt}$. The setting is slightly different from previous work which only optimize $\mathcal{L}_{\text{mrt}}$ in the second step. We find that adding $\widetilde{\mathcal{L}}$ in the experiments could make the training more stable.

When training with $\widetilde{\mathcal{L}}$ in the pre-training step, we apply the scheduled sampling strategy (Bengio et al., 2015) in the entity model as (Miwa and Bansal, 2016). Models are regularized with dropout and trained using Adadelta (Zeiler, 2012). We give the full derivation of Equation 6's gradient in the supplementary. [8]

We select models using development sets: within a fix number of epochs, the model with the best relation extraction performance on the development set is picked out for testing. [9]

4 Experiments

We evaluate the proposed model on two datasets. **ACE05** is a standard corpus for the entity relation extraction task. It is labelled with 7 entity types and 6 relation types. We use the same split of ACE05 documents as previous work (351 training, 80 development, and 80 testing). [10] **NYT** (Riedel et al., 2010) is a larger corpus which is labelled with 3 entity types and 24 relation types. [11] The training set has 353k relation triples which are generated by distant supervision. It also provides another 3880 manually labelled relation triples. Following (Ren et al., 2017; Zheng et al., 2017), we exclude the `None` relation label and randomly select 10% of the labelled data as the development set. We will mainly discuss the results on ACE05 where many previous joint learning models are available for comparison.

We list detailed hyper-parameter settings in the supplementary. Note that, except μ, α, K which are introduced in the joint MRT and selected on the development set, [12] we don't tune hyper-parameters extensively. For example, we use the same setting in both ACE 05 and NYT rather than tune parameters on each of them.

As previous work, we evaluate performances

[5] To accelerate sampling, we borrow the idea of ε-greedy in reinforcement learning: with probability 0.9, we sample t_i' from P_{ent}, and with probability 0.1, we sample it uniformly.

[6] The time complexity is $O(K|s|)$ which is the same to the beam search algorithm with beam size K (Zhang et al., 2017).

[7] Here we follow the literature of MRT to apply the renormalization on $\mathcal{Y}'(s)$. Another formulation is the policy gradient framework which sticks to the original probability.

[8] We remark that the MRT objective (Equation 6) is differentiable with respect to model parameters (Shen et al., 2016). The non-decomposability of the F1 score does not make the model non-differentiable. In our implementation, the gradient is automatically calculated using autograd tools. Please see the supplementary for more details.

[9] We focus on the performance of the ent-to-end relation extraction, so we select models by the relation extraction results. It is also possible to consider both the performances of the entity model and the relation model. We leave the study of advanced model selection algorithms for future work.

[10] We use the dataset in https://github.com/tticoin/LSTM-ER, which is from (Miwa and Bansal, 2016).

[11] https://github.com/shanzhenren/CoType.

[12] The default setting is $\alpha = 10^{-4}, \mu = 1.0, K = 3$ in systems without self-learned Γ loss and $\alpha = 1, \mu = 1.0, K = 2$ in systems with Γ loss.

Model	Entity			Relation		
	P	R	F	P	R	F
L&J (2014)	**85.2**	76.9	80.8	**65.4**	39.8	49.5
M&B (2016)	82.9	**83.9**	83.4	57.2	54.0	55.6
Zhang (2017)	-	-	83.5	-	-	57.5
K&C (2017)	84.0	81.3	82.6	55.5	51.8	53.6
NN	84.0	82.9	83.4	59.5	**56.3**	57.8
MRT	83.9	83.2	**83.6**	64.9	55.1	**59.6**

Table 1: Results on the ACE05 test data. (Miwa and Bansal, 2016) and (Katiyar and Cardie, 2017) are joint training systems without joint decoding. (Li and Ji, 2014) and (Zhang et al., 2017) are joint decoding algorithms. NN is our neural network model without minimum risk training. MRT is minimum risk training with loss Γ (Equation 5). We omit pipeline methods which underperform joint models (see (Li and Ji, 2014) for details).

Settings		F1 of Entity	F1 of Relation
Default sampling	Δ_E	**83.8** $_{+0.4}$	57.9 $_{+0.1}$
	Δ_R	83.5 $_{+0.1}$	58.9 $_{+1.1}$
	Δ_{E+R}	83.6 $_{+0.2}$	59.0 $_{+1.2}$
	Γ	83.6 $_{+0.2}$	58.3 $_{+0.5}$
	$\Gamma + \Delta_{E+R}$	83.6 $_{+0.2}$	**59.6** $_{+1.8}$
Only sampling entity	Δ_E	**83.7** $_{+0.3}$	57.4 $_{-0.4}$
	Δ_R	83.5 $_{+0.1}$	59.1 $_{+1.3}$
	Δ_{E+R}	83.6 $_{+0.2}$	57.9 $_{+0.1}$
	Γ	83.6 $_{+0.2}$	58.7 $_{+0.9}$
	$\Gamma + \Delta_{E+R}$	83.3 $_{-0.1}$	**59.2** $_{+1.4}$

Table 2: MRT with different loss functions and sampling methods. The numbers in subscripts indicate improvements over the NN setting in Table 1.

using precision (P), recall (R) and F1 scores. Specifically, an output entity e is correct if its type and the region of its head are correct, and an output relation r is correct if its e_1, e_2, l are correct (i.e., "exact match").

4.1 Results on ACE05

We first compare proposed models with previous work (Table 1). In general, our plain neural network model (NN) is competitive, and after compiling with MRT, it achieves non-negligible improvement over existing state-of-the-art systems. (both on the entity and the relation extraction). [13] We have following two detailed comparisons.

Among systems which only rely on shared parameters ((Miwa and Bansal, 2016; Katiyar and Cardie, 2017) and NN), NN gives the best result (we give detailed results on different relation types in the supplement). One possible reason is that the "RNN+CNN" network structure is not fully explored in previous joint learning models. More importantly, it suggests that how to build powerful sub-models and utilize shared parameters are still among the key problems of the task.

Comparing with the best joint decoding system which adopts global normalization in training (Zhang et al., 2017), MRT mainly improves the relation extraction results. We think that the improvement may come from the sentence-level loss applied in MRT: both systems consider interactions between decoders, and both objectives are approximated by sampling, but MRT optimizes F1 score while Zhang et al. (2017) optimize label accuracy. For the joint decoding system in (Li and Ji, 2014), although it cannot beat recent neural network-based models, it is interesting to compare MRT with a feature-enriched version of (Li and Ji, 2014)'s model in the future work.

Next, we evaluate the joint MRT with different loss functions and sampling methods.

As mentioned in Section 3.3, we have three options (Δ_{E+R}, Δ_E, Δ_R) for $\Delta(\hat{\mathbf{y}}, \mathbf{y})$ and a self-learned loss function Γ. The first five rows of Table 2 show their performances on the test data. We have three observations regarding the results.

1. Δ_R, Δ_{E+R} have higher relation F1 scores than Δ_E and NN. Thus, adding relation loss in $\Delta(\hat{\mathbf{y}}, \mathbf{y})$ is helpful for relation extraction. We think that knowing the relation loss could bias the entity model to highlight the entities appearing in relations, which provides a better candidate relation set for the relation extraction model.

2. Δ_E has the best entity extraction results, which implies that the sentence-level entity loss alone could benefit entity extraction. While after adding relation loss (Δ_{E+R}), the entity performance slightly decreases. One reason might be that our model selection strategy only focuses on the relation part (footnote 9), thus the model with improved entity performances may not be selected.

3. The learned loss Γ can help to improve performances, but only using Γ is not as effective as the handcrafted Δ functions (which are tailored to the evaluation metrics). By combining both the prior knowledge and information from the dataset, $\Gamma + \Delta_{E+R}$ achieves the best results.

Regarding the sampling method, we test a variant of Algorithm 1 which samples entities but not

[13] It is worth noting that our models don't access additional linguistic resources such as POS tags and dependency trees. We have tried to add syntactic features in (Zhang et al., 2017), but didn't observe improvements.

relations (the last five rows of Table 2). Comparing with the default sampling algorithm, it has similar entity extraction performances, but its behaviour on the relation extraction is different. Specifically, adding entity loss in $\Delta(\hat{\mathbf{y}}, \mathbf{y})$ (i.e., Δ_E, Δ_{E+R}) now affects relation results negatively. It may suggest that when only exploring the output of entity extraction, the entity loss may dominate the relation loss, and trap the joint model to exploit the entity model only. On the other hand, the performances of self-learned loss Γ are less sensitive to the sampling method. We haven't had a clear understanding of the relationship between sampling algorithms and loss functions, but the above results show that adding data-related loss function could improve the robustness of MRT in practice.

Thirdly, we present influences of hyper-parameters for MRT with Δ_{E+R} on the development set in Figure 3 and 4 (other settings have similar results). We find that, for the parameters examined here, it is hard for the entity model and the relation model to agree with each other: parameters achieving high relation performances usually get low entity performances, and vise versa. Thus, if we perform the model selection by only looking at relation extraction results, the joint model may sacrifice entity extraction performances. For α and μ (Figure 3), we observe that on the ACE05 dataset, the model prefers a small α (which means a sharper Q) and μ at boundary (i.e., Q is either close to the entity model or the relation model). Regarding the sample size K (Figure 4), we don't observe a convergence of performances in a small range of K. Since the computation cost increases rapidly as we increase the sample size ($K = 5$ is about 2x slower than $K = 3$ in our implementation), we stick to a small K.

Finally, due to lack of space, we provide more discussions on model configurations (including results regarding different entity pair distances, additional experiments on tuning hyper parameters etc.), and detailed error analyses on concrete samples in the supplementary.

4.2 Results on NYT

We briefly list results on the NYT dataset in Table 3. The baseline methods are (Ren et al., 2017) which is based on a joint embedding of entities and relations, and (Zheng et al., 2017) which conducts joint decoding with an augmented sequence labelling tag set. Both NN and MRT outperform baseline results. In particular, comparing with the joint tagging scheme in (Zheng et al., 2017), MRT adds no constraint on the relation extraction model and can explore the large NYT training set more effectively. At the same time, since the training set is automatically generated, the global losses observed in MRT are also noisy. Like recent work on bandit structured prediction (Kreutzer et al., 2017; Nguyen et al., 2017), the results here suggest that MRT could be a reasonable choice when the supervision of the joint learning is partial and noisy.

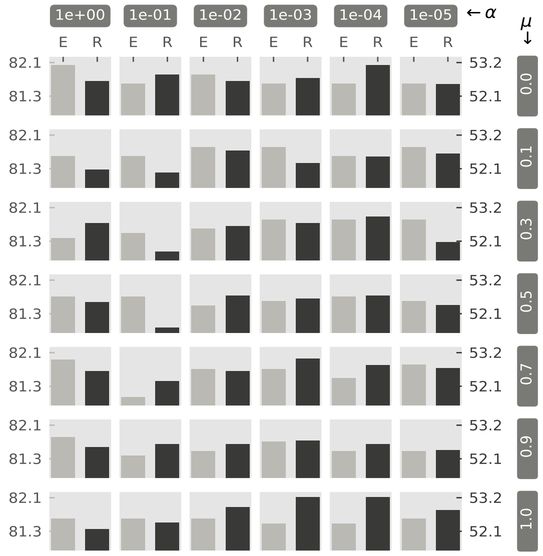

Figure 3: MRT with different Q distributions on the development set. Rows are settings of μ, and columns are settings of α. In each cell, we draw F1 scores of the entity extraction (the left gray bar) and the relation extraction (the right dark bar) under the combination of corresponding α and μ.

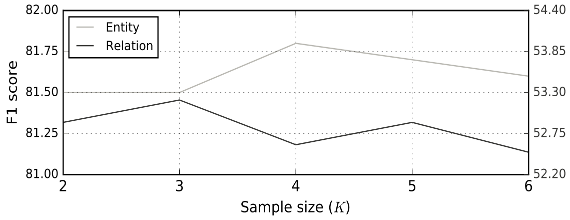

Figure 4: MRT with different sample size K on the development set.

5 Conclusion

We introduced minimum risk training to the task of joint entity and relation extraction. We showed that, with a global loss function, MRT could enhance the connection between the sub-models. Extensive experiments on benchmark datasets witness the effectiveness of the joint MRT.

Model	Relation		
	P	R	F
(Zheng et al., 2017)	61.5	41.4	49.5
NN	61.8	**43.3**	50.9
MRT	**67.4**	42.0	**51.7**
(Ren et al., 2017)	42.3	**51.1**	46.3
NN (exact match)	59.4	41.7	49.0
MRT (exact match)	**65.2**	40.6	**50.0**

Table 3: Results on the NYT dataset. To compare with (Ren et al., 2017), we give results under the "exact match" criterion as ACE05. To compare with (Zheng et al., 2017), we give results which ignore the entity type in the justification of relations. We use $\alpha = 1, \mu = 1, K = 2$ and $\Delta_{E+R} + \Gamma$.

Acknowledgement

The authors wish to thank the reviewers for their helpful comments and suggestions. This research is (partially) supported by STCSM (18ZR1411500), NSFC(61673179), Shanghai Knowledge Service Platform Project (ZF1213), and Shanghai Key Laboratory of Trustworthy Computing (07dz22304201604). The corresponding authors are Yuanbin Wu, Man Lan and Shiliang Sun.

References

Pieter Abbeel and Andrew Y. Ng. 2004. Apprenticeship learning via inverse reinforcement learning. In *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*.

Shiqi Shen Ayana, Zhiyuan Liu, and Maosong Sun. 2016. Neural headline generation with minimum risk training. *arXiv preprint arXiv:1604.01904*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin crfs: Training log-linear models with cost functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 733–736, Los Angeles, California. Association for Computational Linguistics.

Xiaodong He and Li Deng. 2012. Maximum expected bleu training of phrase and lexicon translation models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 292–301, Jeju Island, Korea. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Arzoo Katiyar and Claire Cardie. 2016. Investigating lstms for joint extraction of opinion entities and relations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 919–929, Berlin, Germany. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

Julia Kreutzer, Artem Sokolov, and Stefan Riezler. 2017. Bandit structured prediction for neural sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1503–1513, Vancouver, Canada. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 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*

(Volume 1: Long Papers), pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2009. A rich feature vector for protein-protein interaction extraction from multiple corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 121–130, Singapore. Association for Computational Linguistics.

Khanh Nguyen, Hal Daumé III, and Jordan Boyd-Graber. 2017. Reinforcement learning for bandit neural machine translation with simulated human feedback. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1464–1474, Copenhagen, Denmark. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Nathan D. Ratliff, J. Andrew Bagnell, and Martin Zinkevich. 2006. Maximum margin planning. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 729–736.

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

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*, pages 148–163.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 787–794, Sydney, Australia. Association for Computational Linguistics.

Richard S. Sutton and Andrew G. Barto. 1998. *Introduction to Reinforcement Learning*, 1st edition. MIT Press, Cambridge, MA, USA.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315, Berlin, Germany. Association for Computational Linguistics.

Wenduan Xu, Michael Auli, and Stephen Clark. 2016. Expected f-measure training for shift-reduce parsing with recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 210–220, San Diego, California. Association for Computational Linguistics.

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

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1731–1741, Copenhagen, Denmark. Association for Computational Linguistics.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. 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 (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.