Distantly Supervised Entity Relation Extraction with Adapted Manual Annotations

Changzhi Sun,[1] **Yuanbin Wu**[1,2]
[1]School of Computer Science and Software Engineering, East China Normal University
[2]Shanghai Key Laboratory of Multidimensional Information Processing
changzhisun@stu.ecnu.edu.cn, ybwu@cs.ecnu.edu.cn

Abstract

We investigate the task of distantly supervised joint entity relation extraction. It's known that training with distant supervision will suffer from noisy samples. To tackle the problem, we propose to adapt a small manually labelled dataset to the large automatically generated dataset. By developing a novel adaptation algorithm, we are able to transfer the high quality but heterogeneous entity relation annotations in a robust and consistent way. Experiments on the benchmark NYT dataset show that our approach significantly outperforms state-of-the-art methods.

Introduction

A fundamental problem in information extraction is to recognize entities and relations from plain texts. Given a sentence, the task aims to find out strings representing different objects (e.g., person ("Jobs", "Obama"), organization ("Labour Party")), and different semantic relations among entities (e.g., affiliation relation between a person and an organization). As it is always the first step to convert unstructured texts into structured knowledge, the entity and relation extraction task attracts long lasting interests in both researches and applications of natural language processing.

Given a manually annotated dataset, fully supervised models (especially, neural-network-based models) have achieved remarkable progress on the extraction task (Miwa and Bansal 2016; Katiyar and Cardie 2017; Zheng et al. 2017; Zhang, Zhang, and Fu 2017). However, the main factor limiting the application of these methods is the cost to obtain high quality annotations on entities and relations.

In order to gain more training data on broader domains, Mintz et al. (2009) start using distant supervision from knowledge bases. Specifically, instead of annotating entities and relations manually, we can generate training data automatically via aligning triples in knowledge bases and free texts. The main problem of distantly supervised datasets is the noisy samples: the aligned relations are not always true in contexts. For example, the triple ("Obama", "United States") holds a "born in" relation in a knowledge base, but it does not necessarily mean all appearances of the pair express the same relationship.

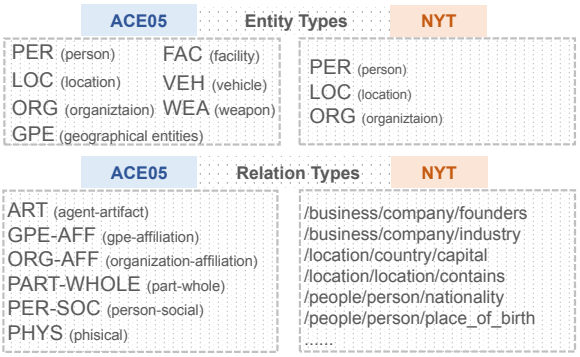

Figure 1: Datasets with different annotation schemas. ACE05 is a manually labelled dataset with 7 entity types and 6 relation types. NYT is a distantly supervised dataset (from Freebase) with 3 entity types and 24 relation types.

In this work, we focus on distantly supervised extraction models, and try to alleviate the effect of noisy training samples by adapting high quality annotations from manually labelled datasets. By introducing accurate context-dependent samples to automatically generated datasets, we aim to make distantly supervised models more accurate and robust.

The major challenge here is that the two datasets are usually heterogeneous: they could annotate different types of entities and relations following different guidelines. For example, Figure 1 shows annotation schemas of two benchmark datasets (ACE05 and NYT). A straightforward way to tackle this challenge is simply merging the two datasets. However, it ignores inherent differences between the two schemas and hardly brings performance gains to distantly supervised models. Another way is trying to establish a (approximate) mapping between the two schemas manually (Qiu, Zhao, and Huang 2013). However, it would be inflexible to extend to more schemas. Thus, how to balance scalability and specificity, and how to carefully control the adaptation process are key problems to the task.

We propose a novel fine-grained framework to incorporate manually labelled dataset to the distantly supervised learning process. Specifically, we introduce two shared tasks to bridge discrepancies between the two heterogeneous annotation schemas. For extracting entities, the shared task is

to locate boundaries of entities. For extracting relations, the shared task is to determine whether two entities form a valid relation. We assume that the two shared tasks are much less sensitive to different annotation schemas, and can be utilized to transfer context-dependent knowledge on entities and relations from the manually labelled dataset. In order to further control the adaptation process, we also investigate consistency constraints between shared extraction tasks and the original extraction tasks. A set of new loss functions is proposed to characterize the constraints.

We conduct extensive experiments on the benchmark NYT dataset, and observe that our model achieves 3.7% improvement of F1 score against the state-of-the-art system. To summarize, the major contributions of this paper are

- We first investigate the problem of using manually labelled datasets to help distantly supervised entity relation extraction.
- We design a novel adaptation framework which transfers heterogeneous knowledge through new shared tasks.
- Our proposed model significantly outperforms the state-of-the-art methods on benchmark NYT dataset. [1]

Related Work

There have been extensive studies on extracting entities and relations from plain texts. Currently, the state-of-the-art systems usually adopt the supervised joint learning algorithm. It can mitigate the error propagation and strengthen the interaction between the entity model and the relation model. Feature-based joint models (Miwa and Sasaki 2014; Li and Ji 2014) use manually extracted features to perform entity detection and relation detection simultaneously. Those methods rely on handcrafted features, which leads to additional complexity. To overcome this limitation, several neural-network-based joint models have been proposed, such as tree LSTM-based model (Miwa and Bansal 2016), attention-based model (Katiyar and Cardie 2017), sequential labelling model (Zheng et al. 2017) and global normalization model (Zhang, Zhang, and Fu 2017). Besides, Ren et al. (2017) study domain-independent framework by modeling entity-relation interactions jointly. Wang et al. (2018) develop a transition-based system for joint extraction task. Specially, our basic models are derived from Sun et al. (2018), which introduce joint minimum risk training to provide a new joint learning paradigm. Different from Ren et al.; Wang et al. (2017; 2018), our basic models do not perform joint decoding or model the dependencies between relations in a sentence.

Another thread of related work is multi-task learning. It has been proven effective in many NLP tasks (Collobert and Weston 2008; Jalali et al. 2010; Peng and Dredze 2016). The basic method is hard parameter sharing (Caruana 1993). (Søgaard and Goldberg 2016) only share parameters at lower layers for lower level tasks. Liu, Qiu, and Huang; Chen et al. (2017; 2017) induce the adversarial shared-private space. However, those approaches do not model the relationships between labels. Meanwhile, Augenstein, Ruder, and Søgaard (2018) present a multi-task framework over disparate label spaces to learn transfer functions between label embeddings. Chen, Zhang, and Liu; Peng, Thomson, and Smith (2016; 2017) train models on disparate annotations of the same task. In contrast, the difference of our method requires multiple tasks on multiple datasets. Besides, Jiang, Huang, and Liu; Qiu, Zhao, and Huang (2009; 2013) study the joint Chinese word segmentation and the part-of-speech tagging task with heterogeneous annotation datasets. Comparing with Qiu, Zhao, and Huang (2013), the shared representations of our method are more explanatory.

Task Definition

Given an input sentence $s = w_1, \ldots, w_{|s|}$ (w_i is a word), we study the task of extracting a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$ from s. An entity $e \in \mathcal{E}$ is a sequence of words labeling 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 this work, we focus on improving distantly supervised information extraction using high quality heterogeneous datasets. Specifically, given two training sets D^a, D^b containing sentences annotated with entities and relations, we assume that D^a is a large automatically labeled dataset which includes (many) noisy annotations, while D^b is a small manually labeled dataset which is more accurate. Furthermore, D^a and D^b could be annotated with different types of entities ($\mathcal{T}_e^a, \mathcal{T}_e^b$) and relations ($\mathcal{T}_r^a, \mathcal{T}_r^b$) following different guidelines (i.e., heterogeneous). Our goal is to strengthen the model performance when training on the noisy dataset D^a.

Basic Models

Before describing our proposed models, we first introduce two building blocks commonly applied in modern information extraction systems: a biLSTM-based sequence labeling model for detecting entities ($\mathcal{M}_{\text{seq}}$) and a CNN-based multi-class classifiers for detecting relations ($\mathcal{M}_{\text{rel}}$).

The Sequence Model $\mathcal{M}_{\text{seq}}$

To extract certain text spans from a sentence, we adopt the BILOU tagging scheme: B, I, L and O denote the begin, inside, last and outside of a target span, U denotes a single word span. For example, to extract entities annotated with entity types $\mathcal{T}_e$, we assign a tag t_i to each word w_i, where $t_i \in \{\text{B}, \text{I}, \text{L}, \text{O}, \text{U}\} \times \mathcal{T}_e$ [2] encodes both the span and the type information of an entity (e.g., (B, PER) means the begin of a PER entity).

Given an input sentence s, the sequence labeling model tries to predict the true tags $\mathbf{t} = t_1, t_2, \ldots, t_{|s|}$ using a biLSTM (bi-directional long short time memory network) chain with parameter $\boldsymbol{\theta}$:

$$\mathbf{h}_i = \text{biLSTM}(\mathbf{x}_i; \boldsymbol{\theta}), \quad (1)$$

[1] We will make our implementation publicly available.

[2] We merge all (O, *) as the single tag O, where $* \in \mathcal{T}_e$.

where $\mathbf{h}_i$ is the hidden state vector of the biLSTM (i.e., concatenation of a forward and a backward LSTM's hidden states at position i), and $\mathbf{x}_i$ is the word representation of w_i which contains pre-trained embeddings and character-based word representations by running a CNN on the character sequences of w_i. Then, the posterior of tag $\hat{t}_i$ is given by

$$P_{\text{seq}}(\hat{t}_i|s) = \text{Softmax}(\mathbf{W}_e\mathbf{h}_i),$$

where $\mathbf{W}_e$ is the parameter. The objective is to minimize

$$\mathcal{L}_{\text{seq}} = -\frac{1}{|s|}\sum_{i=1}^{|s|}\log P(\hat{t}_i = t_i|s). \quad (2)$$

The Relation Model $\mathcal{M}_{\text{rel}}$

Given a pair of entities (e_1, e_2) extracted by a sequence labeling model $\mathcal{M}_{\text{seq}}$, we use a multi-class classifier to determine whether they form a certain type of relation. For example, to identify relations in $\mathcal{T}_r$, the classifier assigns a relation label $l \in \{\texttt{NONE}\} \cup \mathcal{T}_r$ to the pair, where NONE means no relation exists.

The multi-class classifier first applies CNNs (convolution neural networks) to extract features for (e_1, e_2):

$$\mathbf{f}_{e_1,e_2} = \text{CNNs}(e_1, e_2, s; \boldsymbol{\omega}), \quad (3)$$

where different CNNs are applied on words inside entities e_1, e_2, and context words between them (see the experiment section for details). Here, each word is represented using hidden state vectors $\mathbf{h}_i$ of the biLSTM chain (i.e., sharing parameters with $\mathcal{M}_{\text{seq}}$) and one-hot entity tag representations. Then, the posterior of relation type $\hat{l}$ is obtained by a multi-layer perceptron with one hidden layer,

$$P_{\text{rel}}(\hat{l}|e_1, e_2, s) = \text{Softmax}(\mathbf{W}_{r_2}\text{ReLU}(\mathbf{W}_{r_1}\mathbf{f}_{e_1,e_2})), \quad (4)$$

and the training objective is to minimize

$$\mathcal{L}_{\text{rel}} = -\sum_{(e_1,e_2)}\frac{\log P(\hat{l} = l|e_1, e_2, s; \boldsymbol{\omega})}{\#\text{ candidate pairs } (e_1, e_2)}, \quad (5)$$

where the true label l can be read from annotations, and model parameters are $\mathbf{W}_{r_1}$, $\mathbf{W}_{r_2}$ and $\boldsymbol{\omega}$.

A Joint Extraction Model on D^a

With the help of the two building blocks, we are able to build a simple joint extraction system given the training set D^a [3]. First, we apply a sequence labeling model $\mathcal{M}^a_{\text{seq}}$ to extract entities with type $\mathcal{T}^a_e$. Then, for each pair of the extracted entities, we use a relation classifier $\mathcal{M}^a_{\text{rel}}$ to recognize relations in $\mathcal{T}^a_r$. Finally, to train the model jointly, we simply minimize the sum of their objectives $\mathcal{L}^a_{\text{seq}} + \mathcal{L}^a_{\text{rel}}$.

In the following section, we will describe our strategies to improve the performances of this simple model. In general, instead of directly training with noisy samples in D^a, we will try to utilize another heterogeneous but high quality dataset D^b.

[3] To make notations clear, we will always use superscript to indicate different models and parameters. For example $\mathcal{M}^a_{\text{seq}}, \mathcal{M}^b_{\text{seq}}$ are sequence models training on D^a and D^b respectively, $\mathbf{h}^a_i, \mathbf{h}^b_i$ are their hidden states, and P^a, P^b are their posterior distributions.

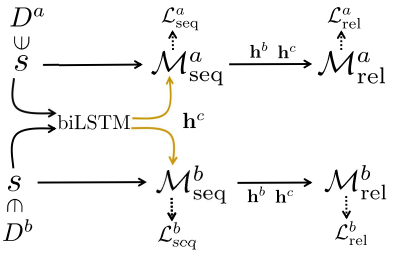

Figure 2: Adapting via shared representations $\mathbf{h}^c$.

Adaptation Models

Although distant supervision is an effective way to obtain large amounts of data, drawbacks of the automatically generated D^a are also obvious. For example, some annotated relations are incorrect (false positive) due to the heuristic grounding of knowledge base triples, and some true relations may not be annotated (false negative) due to the low coverage of seed relations of the distant supervision process.

On the other hand, there are many manually labeled datasets which are built for extracting entities and relations in various domains. They may not follow the same annotation schema with D^a, but the high quality annotations in them may provide some useful common knowledge which is invariant with respect to domains. Thus, it's interesting to study whether we can improve model performances on D^a by adapting those heterogeneous datasets.

Adapting via Merging Datasets

Our first attempt is simply merging datasets D^a and D^b, and feeding the merged dataset into a vanilla joint extraction model ($\mathcal{M}_{\text{seq}}, \mathcal{M}_{\text{rel}}$). The system is the same as trained on D^a, except that now it needs to work with larger sets of entity types $\mathcal{T}^a_e \cup \mathcal{T}^b_e$ and relation types $\mathcal{T}^a_r \cup \mathcal{T}^b_r$.

However, according to our empirical evaluation (on the testing set of D^a), this simple solution fails to improve performances. In fact, since the distant supervised set D^a is usually much larger than the manually labeled set D^b, directly mixing the two training sets is inefficient to explore the samples in D^b. Thus, we may need to develop fine-grained methods to control the adaptation process.

Adapting via Shared Representations

Instead of using a unified model, we could keep extraction models of D^a and D^b separated and capture interactions between the two models using an additional model. Specifically, we break the task of adapting D^b to D^a into two steps. First, we try to identify shared information between the two datasets which will serve as common knowledge for extracting entities and relations. Second, we combine the shared and the private (dataset-dependent) information in each model's prediction (Figure 2).

Formally, for each sentence $s \in D^a$, we feed it into a joint extraction model with a sequence model $\mathcal{M}^a_{\text{seq}}$ and a relation model $\mathcal{M}^a_{\text{rel}}$. Similarly, for sentences in D^b, we apply another pair of models $\mathcal{M}^b_{\text{seq}}$, $\mathcal{M}^b_{\text{rel}}$. The biLSTMs and CNNs

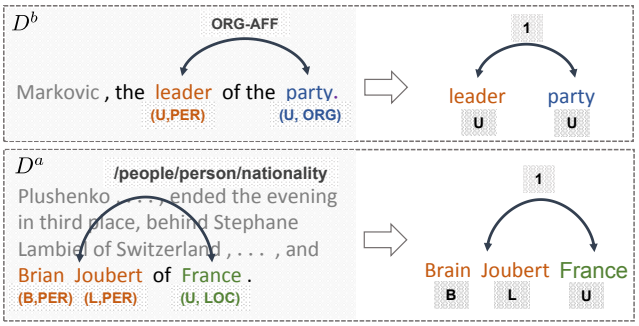

Figure 3: An example of different annotations and their transformation versions for the shared extraction tasks. The left part is original annotations in D^a and D^b. The right part is corresponding annotations for the entity span detection and the binary relation detection.

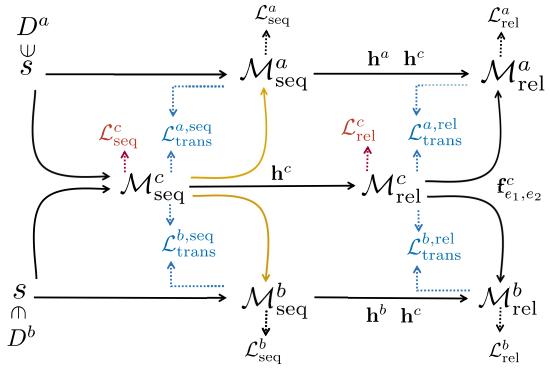

Figure 4: Adapting via shared tasks. As Figure 2, shared representations are included (the yellow flow). The main differences are the new loss functions in red and blue .

in these models will aim to represent dataset-dependent features.

In order to encode the shared information, we introduce a new biLSTM chain which accepts both sentences from D^a and D^b. The hidden states $\mathbf{h}^c_i$ of this shared biLSTM are concatenated to $\mathbf{h}^a_i$ of $\mathcal{M}^a_{\text{seq}}$ and $\mathbf{h}^b_i$ of $\mathcal{M}^b_{\text{seq}}$ before performing the private prediction on the entity tag t_i,

$$P^a_{\text{seq}}(\hat{t}_i|s) = \text{Softmax}(\mathbf{W}^a_e(\mathbf{h}^a_i \oplus \mathbf{h}^c_i)), \quad (6)$$

$$P^b_{\text{seq}}(\hat{t}_i|s) = \text{Softmax}(\mathbf{W}^b_e(\mathbf{h}^b_i \oplus \mathbf{h}^c_i)). \quad (7)$$

The relation models $\mathcal{M}^a_{\text{rel}}, \mathcal{M}^b_{\text{rel}}$ also build feature vectors $\mathbf{f}^a_{e_1,e_2}, \mathbf{f}^b_{e_1,e_2}$ using the shared representation $\mathbf{h}^c_i$.

Here, the main assumption is that if the representation $\mathbf{h}^c$ is useful on both D^a and D^b, it probably captures some shared information between them. Thus, from the perspective of D^a, instead of only using its private feature representation, we also adapt some knowledge from D^b through the shared representation. Our major concern about only using shared representations in adaptation is that the meaning of shared information is unclear. In fact, there is no criteria about which representations are better except the end-performances. Thus, it may be possible that $\mathbf{h}^c$ hold unnecessary information to overfit the training sets.

Adapting via Shared Tasks

Inspired by the separation of shared and private representations, we can add more control on the adaptation process. A key observation is that although D^a and D^b have different annotation schemas, we can decompose the extraction tasks into overlapped subtasks. For example, to extract entities, we can first find the start and end positions of them, then determine their entity types. It is possible that even though entity types $\mathcal{T}^a_e, \mathcal{T}^b_e$ are different, the distributions of the start and end positions could be much closer in the two datasets. For example, in Figure 3, "party" and "France" are annotated with different entity types in D^a and D^b, but the start positions of them are both right behind the preposition "of". Thus, it would be reasonable to assume that they share a task of predicting boundaries of entity spans.

Similarly, in order to extract relations, we can first predict whether an entity pair forms a valid relation, then determine their relation types. In Figure 3, entity pairs ("leader", "party") and ("Brian Joubert", "France") are annotated with different types in D^a and D^b, but both of them hold a syntax relation of nominal modification. It suggests that we can also add a share task to predict the existence of a relation.

Formally, we introduce two shared tasks to help the adaptation: entity span detection and binary relation detection (Figure 4). The **entity span detection** task focuses on locating boundaries of entities. As for the ordinary entity extraction task, we apply a sequence labeling model $\mathcal{M}^c_{\text{seq}}$. For each word w_i, $\mathcal{M}^c_{\text{seq}}$ predicts a tag in $\{\texttt{B}, \texttt{I}, \texttt{L}, \texttt{O}, \texttt{U}\}$ to mark the appearance of an entity (ignoring specific entity types). It also shares hidden states $\mathbf{h}^c$ with $\mathcal{M}^a_{\text{seq}}$ and $\mathcal{M}^b_{\text{seq}}$ as Equation 6 and 7. More importantly, $\mathcal{M}^c_{\text{seq}}$ now equips with a loss function $\mathcal{L}^c_{\text{seq}}$ which is based on the entity span annotations in both D^a and D^b. We compute $\mathcal{L}^c_{\text{seq}}$ using Equation 2 by transforming ground truth annotations $\mathbf{t}$ into the BILOU schema.

The **binary relation detection** task is to predict whether a certain relation exists between an entity pair (ignoring specific relation types). We use a relation model $\mathcal{M}^c_{\text{rel}}$ which is applied on the candidate entity pairs from $\mathcal{M}^c_{\text{seq}}$, and extract entity pair representations $\mathbf{f}^c_{e_1,e_2}$ based on the hidden states $\mathbf{h}^c$. It will assign each entity pair a label in $\{0, 1\}$ to indicate the existence of a relation. Like $\mathcal{M}^c_{\text{seq}}$, we add a loss function $\mathcal{L}^c_{\text{rel}}$ on outputs of $\mathcal{M}^c_{\text{rel}}$ to supervise the training process. The computation of $\mathcal{L}^c_{\text{rel}}$ follows Equation 5 with true binary annotations l transformed from the original typed relation labels. One additional observation is that we could also utilize representations of entity pairs $\mathbf{f}^c_{e_1,e_2}$ in the private relation models.

Comparing with sharing representations, the main difference here is the loss functions $\mathcal{L}^c_{\text{seq}}, \mathcal{L}^c_{\text{rel}}$ on the shared task. In fact, as the shared loss functions are computed on the merged dataset $D^a \cup D^b$, we are combining the paradigms of adapting via merged dataset and adapting via shared representations in a more careful way.

Transition Loss Functions

Based on the designed shared tasks, we can add a set of consistent constrains to further control the relation between the shared and private models.

For entity models, $\mathcal{M}^a_{\text{seq}}$ outputs a posterior $P^a_{\text{seq}}(\hat{t}_i|s)$ over the tag set $\{\texttt{B}, \texttt{I}, \texttt{L}, \texttt{O}, \texttt{U}\} \times \mathcal{T}^a_e$, and $\mathcal{M}^c_{\text{seq}}$ outputs a posterior $P^c_{\text{seq}}(\hat{t}_i|s)$ over $\{\texttt{B}, \texttt{I}, \texttt{L}, \texttt{O}, \texttt{U}\}$ which is a projected set. A natural consistency requirement between P^a_{seq} and P^c_{seq} is the margin constraint,

$$P^c_{\text{seq}}(\star|s) = \sum_{*\in\mathcal{T}^a_e} P^a_{\text{seq}}((\star,*)|s), \quad \star \in \{\texttt{B}, \texttt{I}, \texttt{L}, \texttt{O}, \texttt{U}\}. \quad (8)$$

In other words, given a sentence in D^a and a position i, the probability of being the start of an entity (B) should be the sum of probabilities of being the start of any specific entity (B, *). The same consistency problem also appears between P^b_{seq} and P^c_{seq}.

Instead of adding hard constraints, we characterize the consistency using soft constraints. Specifically, we will try to minimize two new **transition loss functions** $\mathcal{L}^{a,\text{seq}}_{\text{trans}}, \mathcal{L}^{b,\text{seq}}_{\text{trans}}$,

$$\mathcal{L}^{\triangledown,\text{seq}}_{\text{trans}} = \|P^c_{\text{seq}} - M^{\triangledown}_{\text{seq}} P^{\triangledown}_{\text{seq}}\|_2, \quad \triangledown \in \{a, b\}. \quad (9)$$

where $M^a_{\text{seq}}, M^b_{\text{seq}}$ are transition matrices which convert discrete distributions $P^a_{\text{seq}}, P^b_{\text{seq}}$ to P^c_{seq}. We can also add similar transition loss functions for the relation model,

$$\mathcal{L}^{\triangledown,\text{rel}}_{\text{trans}} = \|P^c_{\text{rel}} - M^{\triangledown}_{\text{rel}} P^{\triangledown}_{\text{rel}}\|_2, \quad \triangledown \in \{a, b\}. \quad (10)$$

The adaptation model jointly minimize the sum of loss functions

$$\sum_{\triangle\in\{a,b,c\}} \left(\mathcal{L}^{\triangle}_{\text{seq}} + \mathcal{L}^{\triangle}_{\text{rel}}\right) + \sum_{\triangledown\in\{a,b\}} \left(\mathcal{L}^{\triangledown,\text{seq}}_{\text{trans}} + \mathcal{L}^{\triangledown,\text{rel}}_{\text{trans}}\right) \quad (11)$$

To summarize, adapting via shared tasks is a finer model than only sharing representations. It has a clear interpretation of the shared model, and the connection between the shared and the private task can be explicitly characterized. These prior knowledge on model design could make the learned models more stable and robust.

Training the Models

To train the joint model, we optimize the Equation 11 on two datasets in an alternative way. Specifically, we alternately select a random batch from the two datasets D^a and D^b, then the Equation 11 on batch B is reduced to

$$\sum_{\triangle\in\{\triangledown,c\}} \left(\mathcal{L}^{\triangle}_{\text{seq}} + \mathcal{L}^{\triangle}_{\text{rel}}\right) + \left(\mathcal{L}^{\triangledown,\text{seq}}_{\text{trans}} + \mathcal{L}^{\triangledown,\text{rel}}_{\text{trans}}\right), \triangledown \in \{a, b\}. \quad (12)$$

The training procedure is described in Algorithm 1. We employ the scheduled sampling strategy (Bengio et al. 2015) in the entity model similar to (Miwa and Bansal 2016). We optimize our model using Adadelta (Zeiler 2012) with gradient clipping. The network is regularized with dropout. [4] Within a fixed number of batches, we select the model according to the best relation performance on development sets.

[4]Our word embeddings is initialized with 100-dimensional glove (Pennington, Socher, and Manning 2014) word embeddings. The dimensionality of the hidden units is 128. For all CNN in our network, the kernel sizes are 2 and 3, and the output channels are 25.

Algorithm 1 The training procedure

Input: D^a, D^b,
Output: the trained model

```
randomly initialize parameters
while the model has not converged do
    for ▽ ∈ {a, b} do
        randomly select batch B from corpus D^▽
        compute L^▽_seq, L^c_seq on batch B by Equation 2
        compute L^{▽,seq}_trans by Equation 9
        generate candidate relations
        compute L^▽_rel, L^c_rel by Equation 5
        compute L^{▽,rel}_trans by Equation 10
        compute final loss L by Equation 12
        update parameters by minimizing loss L
    end for
end while
```

Experiments

We evaluate the proposed framework on public **NYT** dataset. [5] The training set has 353k relation triples, which are generated by distant supervision. The test set is manually labeled and contains 3880 relation triples. Following (Ren et al. 2017; Zheng et al. 2017; Wang et al. 2018), we randomly select 10% of the test set as the development set and use the remaining data as evaluation. We use standard **ACE05** dataset as the manually labeled dataset (see Figure 1) [6] .

We compute the Equation 3 according to (Sun et al. 2018). Specifically, the $\mathbf{f}_{e_1,e_2}$ is concatenated by six feature vectors, namely, $\mathbf{f}_{e_1}, \mathbf{f}_{e_2}, \mathbf{f}_{\text{middle}}, \mathbf{f}_{\text{left}}, \mathbf{f}_{\text{right}}$ and $\mathbf{f}_{\text{dist}}$. $\mathbf{f}_{e_1}$ and $\mathbf{f}_{e_2}$ are feature vectors by running two CNNs on word representations of e_1 and e_2. Similarly, we build $\mathbf{f}_{\text{middle}}$ with another CNN, on words between e_1 and e_2. We use "LSTM-Minus" method (Wang and Chang 2016; Zhang, Zhang, and Fu 2017; Sun et al. 2018) to compute left context feature vectors $\mathbf{f}_{\text{left}}$ and right context feature vectors $\mathbf{f}_{\text{right}}$. For $\mathbf{f}_{\text{dist}}$, we use one-hot feature vectors to represent the distance between e_1 and e_2 in the sentence.

We evaluate the performances 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., "exactly match"). In previous work, the entity type are not considered when computing the relation F1 score (Ren et al. 2017; Zheng et al. 2017; Wang et al. 2018). We also report this results for comparison.

In this paper, the default setting is the adaptation model with entity transition loss (line 6 in Table 2), which achieves the best relation performance on NYT dataset.

Results on NYT

First, we compare our method with previous work (Table 1). The first part contains pipelined methods, and the second part contains joint extraction models. The last part includes joint extraction models with the "exactly match" evaluation.

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

[6]https://github.com/tticoin/LSTM-ER.

Model	Relation		
	P	R	F
Gormley (2015)	55.3	15.4	24.0
Mintz (2009)	25.8	39.3	31.1
Tang (2015)	33.5	32.9	33.2
Hoffman (2011)	33.8	32.7	33.3
L&J (2014)	57.4	25.6	35.4
Ren (2017)	42.3	**51.1**	46.3
Zheng (2017)	61.5	41.4	49.5
Wang (2018)	64.3	42.1	50.9
Sun (2018)	67.4	42.0	51.7
Our Model	**70.4**	45.6	**55.4**
Sun (2018)(exactly match)	65.2	40.6	50.0
Our Model(exactly match)	68.3	44.2	53.7

Table 1: Results on the NYT dataset.

In general, our proposed method achieves significant improvements over all the existing models in relation F1 score. In particular, it achieves 5.9 percent improvement over the joint sequence labeling method (Zheng et al. 2017) and outperforms 4.5 percent comparing with the joint transition-based system (Wang et al. 2018). Comparing with the state-of-the-art method (Sun et al. 2018), it achieves 3.7 point improvement. It shows that our method can boost the performance of distant supervision using the manually labeled high quality dataset.

Next, we analyse the contributions and effects of the various components of our method (Table 2). We have some observations regarding this results.

1. "only D^a" (line 1) is competitive with current best joint decoders (Wang et al. 2018; Sun et al. 2018). It suggests that basic models $\mathcal{M}_{\text{seq}}, \mathcal{M}_{\text{rel}}$ (see previous section) are effective.

2. "$D^a \cup D^b$" (line 2) has poor relation performance comparing"only D^a". We think that directly mixing the two datasets is inefficient since the D^a dataset is much larger than the D^b dataset,

3. After exploiting the manually labeled ACE05 dataset via shared representations (line 3), it achieves slight improvements on both the entity and the relation performances. These observations show that the shared representations can improve performances and it is a simple method for adaptation model.

4. After adding two shared tasks (line 4), both the entity and relation performances have large improvements (1.2 percent for entity and 2.7 percent for relation). It demonstrates that this model can combine the paradigms of adapting via merged dataset and adapting via shared representations in a more careful way.

5. After imposing the transition losses both on entity model and relation model (line 5), the relation performance has slight improvement (0.1 percent), but the entity performance decreases. Interestingly, when we keep only transition loss on entity model (line 6), it largely improves the relation performance. Meanwhile, when we keep only transition loss on relation model (line 7), it fails to improve relation performance but achieves the best entity performance. These observations show that the transition loss added to the entity model or relation model could bias the performance of entity or relation. One possible reason is that the entity model and the relation model are closely related to each other, and imposing restrictions on one side will affect the other. However, how they affect each other in this joint settings is still a open question.

Model	Entity			Relation		
	P	R	F	P	R	F
only D^a	82.6	91.2	86.7	61.8	43.3	50.9
$D^a \cup D^b$	84.5	91.9	88.1	60.7	42.3	49.8
only $\mathbf{h}^c$	83.5	93.1	88.0	65.6	42.0	51.2
$\mathbf{h}^c + \mathcal{L}^c$	86.2	92.5	89.2	66.5	45.4	53.9
$+ \mathcal{L}_{\text{trans}}$	82.8	89.6	86.1	64.8	46.2	54.0
$+ \mathcal{L}^{\text{seq}}_{\text{trans}}$	86.6	92.9	89.6	70.4	**45.6**	**55.4**
$+ \mathcal{L}^{\text{rel}}_{\text{trans}}$	**87.2**	**93.9**	**90.4**	**72.9**	40.9	52.4

Table 2: Results on the NYT dataset in different settings. "only D^a" uses basic models trained on D^a; "$D^a \cup D^b$" denotes the adaptation model via merging datasets; "only $\mathbf{h}^c$" denotes adaptation model via shared representations; "$\mathbf{h}^c + \mathcal{L}^c$" is the adaptation model via shared task. "$+ \mathcal{L}_{\text{trans}}$" is the "$\mathbf{h}^c + \mathcal{L}^c$" with entity and relation transition losses; Similarly, "$+ \mathcal{L}^{\text{seq}}_{\text{trans}}$" and "$+ \mathcal{L}^{\text{rel}}_{\text{trans}}$" keep only transition losses on entity and relation respectively.

Percentage of D^b	Entity			Relation		
	P	R	F	P	R	F
100%	**86.6**	**92.9**	**89.6**	**70.4**	45.6	**55.4**
75%	83.6	91.1	87.2	69.0	42.0	52.2
50%	85.9	92.7	89.2	62.8	**47.6**	54.2
25%	84.5	91.4	87.8	67.3	44.5	53.6

Table 3: Results on the NYT dataset varying on the percentage of the ACE05 dataset.

Thirdly, we present the influences of quantity of manually labeled dataset (Table 3). We randomly select 25%, 50%, 75% of the ACE05 dataset. We note that as the quantity increases, the performance does not increase steadily. We think this is caused by random sampling. In other word, the impact of each sample of ACE05 dataset on performance is different. For example, the result with 50% ACE05 dataset has high recall, but the result with 100% ACE05 dataset has high precision. How to select samples efficiently could be an interesting future work.

Forthly, we visualize the transition matrix in Figure 5 (take M^a_{seq} as a example). Darker color indicates larger weight. The diagonal entries have a darker color, which means the transition meets the Equation 8 approximately.

S1	after the authorities described suspects talking about blowing up the [sears tower]$^{\text{ORG:}\heartsuit\clubsuit\spadesuit}_{\texttt{contains-2:}\heartsuit\clubsuit}$ in [chicago]$^{\text{LOC:}\heartsuit\clubsuit\spadesuit}_{\texttt{contains-1:}\heartsuit\clubsuit}$ and the f.b.i. 's [miami]$^{\text{LOC:}\heartsuit\clubsuit\spadesuit}$ headquarters .
S2	said [dennis rice]$^{\text{PER:}\heartsuit\clubsuit\spadesuit}_{\texttt{company-1:}\heartsuit\spadesuit}$, senior vice president of marketing for [disney]$^{\text{ORG:}\heartsuit\clubsuit\spadesuit}_{\texttt{company-2:}\heartsuit\spadesuit}$'s buena vista [pictures]$^{\text{ORG:}\clubsuit}$ unit .
S3	in [california]$^{\text{LOC:}\heartsuit\clubsuit\spadesuit}_{\texttt{contains-1:}\heartsuit\clubsuit\spadesuit\texttt{|contains-3:}\clubsuit}$, where parents first started educational foundations in response to a statewide law capping property taxes , the combined district of [santa monica]$^{\text{LOC:}\heartsuit\clubsuit\spadesuit}_{\texttt{contains-4:}\clubsuit\texttt{|contains-2:}\spadesuit}$ and [malibu]$^{\text{LOC:}\heartsuit\clubsuit\spadesuit}_{\texttt{contains-2:}\heartsuit\clubsuit}$ requires . . .

Table 4: Examples from the NYT dataset with label annotations from "$+\ \mathcal{L}^{\text{seq}}_{\text{trans}}$" model and "only $\mathbf{h}^c$" model for comparison. The $\heartsuit$ is the gold standard, and the $\clubsuit$, $\spadesuit$ are the output of the "$+\ \mathcal{L}^{\text{seq}}_{\text{trans}}$" ,"only $\mathbf{h}^c$" model respectively.

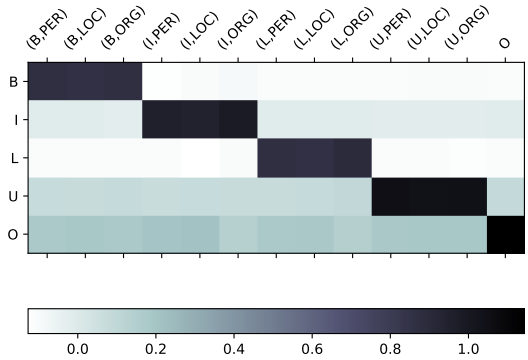

Figure 5: Visulization of the transition matrix M^a_{seq}.

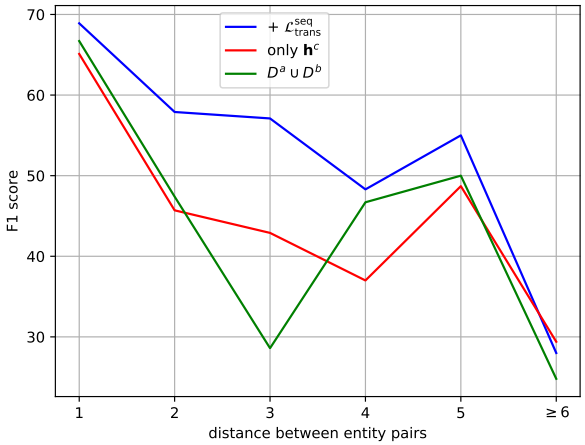

Figure 6: F1 scores on NYT dataset with respect to the distance between entity pairs.

For example, the probability of tag `B` mainly comes from the transition of tag (`B,PER`), (`B,LOC`) and (`B,ORG`). In addition, to compare with automatically learned transition matrix, we try to fix the matrix by the mapping between the tag of original task and the tag of shared task. Specifically, in Figure 5, the diagonal entries are set to 1.0, while others are set to 0.0. In this case, it achieves 88.9 entity F1 score and 52.8 relation F1 score, demonstrating the effectiveness of the soft constraints on modeling consistency.

Fifthly, we examine the performance with respect to different distances between entity pairs (Figure 6). In general, our model outperforms the two baselines significantly when the distance is lower than 6. Besides, we observe that the performances of all models are very low when the distance is greater than 6. Thus, joint decoding algorithms which can capture long distance dependencies might be a promising direction in this joint extraction task.

Finally, in this work, we focus on improving distantly supervised information extraction using high quality heterogeneous datasets. We report the performances on ACE05 dataset. We use the same data split as previous work (Li and Ji 2014; Miwa and Bansal 2016; Sun et al. 2018). Our basic models achieve 57.8 relation F1 score trained on ACE05. Comparing with the model only trained on ACE05, our systems (with NYT) have nearly the same performance. For example, our best model "$+\ \mathcal{L}^{\text{seq}}_{\text{trans}}$" achieves 56.6 relation F1 score on test set. We think the automatically generated data degrades the quality of whole dataset, due to it includes many noisy samples and has the low coverage.

Case Study

We compare the "$+\ \mathcal{L}^{\text{seq}}_{\text{trans}}$" with the "only $\mathbf{h}^c$" on some concrete examples, as shown in Table 4. For S1, our model identifies a `contains` relation between "[chicago]$^{\text{LOC}}$" and "[sears tower]$^{\text{LOC}}$", while the "only $\mathbf{h}^c$" do not find this relation even the entities are correct. For S2, the "$+\ \mathcal{L}^{\text{seq}}_{\text{trans}}$" does not detect `company` relation while the "only $\mathbf{h}^c$" correctly find it. These two observations show that our model is good at dealing with the situation when the distance between entities is low, as expected. For S3, the "$+\ \mathcal{L}^{\text{seq}}_{\text{trans}}$" wrongly identifies the relation between "[california]$^{\text{LOC}}$" and "[santa monica]$^{\text{LOC}}$" even the relation between "[california]$^{\text{LOC}}$" and "[malibu]$^{\text{LOC}}$" is detected. We think advanced improvement methods which model dependencies between relations might be helpful in this situation.

Conclusions

We propose a novel adaptation framework for distantly supervised joint entity relation extraction using the high quality heterogeneous dataset. By introducing shared extraction

tasks and imposing consistency constraints between shared extraction tasks and the original extraction tasks, our framework could control the adaptation process in a more careful and interpretable way. Experiments on benchmark NYT dataset show the effectiveness of the proposed methods.

Acknowledgements

The authors wish to thank all reviewers for their helpful comments and suggestions. The corresponding author is Yuanbin Wu. This research is supported by STCSM (18ZR1411500).

References

Augenstein, I.; Ruder, S.; and Søgaard, A. 2018. Multi-task learning of pairwise sequence classification tasks over disparate label spaces. In *Proc. of NAACL*, volume 1, 1896–1906.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 1171–1179.

Caruana, R. A. 1993. Multitask learning: A knowledge-based source of inductive bias. *Machine Learning Proceedings* 10(1):41–48.

Chen, X.; Shi, Z.; Qiu, X.; and Huang, X. 2017. Adversarial multi-criteria learning for chinese word segmentation. In *Proc. of ACL*, 1193–1203.

Chen, H.; Zhang, Y.; and Liu, Q. 2016. Neural network for heterogeneous annotations. In *Proc. of EMNLP*, 731–741.

Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*, 160–167.

Gormley, M. R.; Yu, M.; and Dredze, M. 2015. Improved relation extraction with feature-rich compositional embedding models. *arXiv preprint arXiv:1505.02419*.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 541–550.

Jalali, A.; Sanghavi, S.; Ruan, C.; and Ravikumar, P. K. 2010. A dirty model for multi-task learning. In *Advances in neural information processing systems*, 964–972.

Jiang, W.; Huang, L.; and Liu, Q. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging – a case study. In *Proc. of IJCNLP*, 522–530.

Katiyar, A., and Cardie, C. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proc. of ACL*, 917–928.

Li, Q., and Ji, H. 2014. Incremental joint extraction of entity mentions and relations. In *Proc. of ACL*, 402–412.

Liu, P.; Qiu, X.; and Huang, X. 2017. Adversarial multi-task learning for text classification. In *Proc. of ACL*, 1–10.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proc. of IJCNLP*, 1003–1011. Suntec, Singapore: Association for Computational Linguistics.

Miwa, M., and Bansal, M. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proc. of ACL*, 1105–1116.

Miwa, M., and Sasaki, Y. 2014. Modeling joint entity and relation extraction with table representation. In *Proc. of EMNLP*, 1858–1869.

Peng, N., and Dredze, M. 2016. Multi-task multi-domain representation learning for sequence tagging. *CoRR, abs/1608.02689*.

Peng, H.; Thomson, S.; and Smith, N. A. 2017. Deep multitask learning for semantic dependency parsing. In *Proc. of ACL*, 2037–2048.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*, 1532–1543.

Qiu, X.; Zhao, J.; and Huang, X. 2013. Joint Chinese word segmentation and POS tagging on heterogeneous annotated corpora with multiple task learning. In *Proc. of EMNLP*, 658–668.

Ren, X.; Wu, Z.; He, W.; Qu, M.; Voss, C. R.; Ji, H.; Abdelzaher, T. F.; and Han, J. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proc. of WWW*, 1015–1024.

Søgaard, A., and Goldberg, Y. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proc. of ACL*, 231–235.

Sun, C.; Wu, Y.; Lan, M.; Sun, S.; Wang, W.; Lee, K.-C.; and Wu, K. 2018. Extracting entities and relations with joint minimum risk training. In *Proc. of EMNLP*.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proc. of WWW*, 1067–1077.

Wang, W., and Chang, B. 2016. Graph-based dependency parsing with bidirectional lstm. In *Proc. of ACL*, 2306–2315.

Wang, S.; Zhang, Y.; Che, W.; and Liu, T. 2018. Joint extraction of entities and relations based on a novel graph scheme. In *IJCAI*, 4461–4467.

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

Zhang, M.; Zhang, Y.; and Fu, G. 2017. End-to-end neural relation extraction with global optimization. In *Proc. of EMNLP*, 1731–1741.

Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; and Xu, B. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proc. of ACL*, 1227–1236.