# Exploring Encoder-Decoder Model for Distant Supervised Relation Extraction

**Sen Su, Ningning Jia, Xiang Cheng*, Shuguang Zhu, Ruiping Li**
State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China
{susen, jianingning, chengxiang, zsg1990ok, liruiping}@bupt.edu.cn

## Abstract

In this paper, we present an encoder-decoder model for distant supervised relation extraction. Given an entity pair and its sentence bag as input, in the encoder component, we employ the convolutional neural network to extract the features of the sentences in the sentence bag and merge them into a bag representation. In the decoder component, we utilize the long short-term memory network to model relation dependencies and predict the target relations in a sequential manner. In particular, to enable the sequential prediction of relations, we introduce a measure to quantify the amounts of information the relations take in their sentence bag, and use such information to determine the order of the relations of a sentence bag during model training. Moreover, we incorporate the attention mechanism into our model to dynamically adjust the bag representation to reduce the impact of sentences whose corresponding relations have been predicted. Extensive experiments on a popular dataset show that our model achieves significant improvement over state-of-the-art methods.

## 1 Introduction

Knowledge bases (KBs) such as Freebase [Bollacker *et al.*, 2008], DBpedia [Auer *et al.*, 2007], and NELL [Carlson *et al.*, 2010] are extremely useful resources for many NLP tasks including information retrieval and question answering. These KBs compose of relational facts with triple format, e.g., */location/country/capital* (New Zealand, Wellington). Although existing KBs contain more than millions of facts, they are still far from complete compared to the infinite real-world facts. Relation extraction, which aims to extract relations between two entities, is a crucial task to enrich KBs.

Distant supervision is a promising approach for relation extraction, which can automatically generate training instances by aligning facts in knowledge bases with sentences in free texts [Mintz *et al.*, 2009]. As shown in Figure 1, */location/location/contains* (Washington, Cashmere) is a fact in Freebase, sentences that contain entity pair (Washington, Cashmere) will be regarded as training instances for relation */location/location/contains*. Distant supervision scales relation extraction to very large corpora which contains thousands of relations, and has been widely used for finding relational facts in free texts. Since an entity pair in KBs may have one or multiple relations, distant supervised relation extraction can be formalized as a multi-instance multi-label learning problem [Hoffmann *et al.*, 2011; Surdeanu *et al.*, 2012].

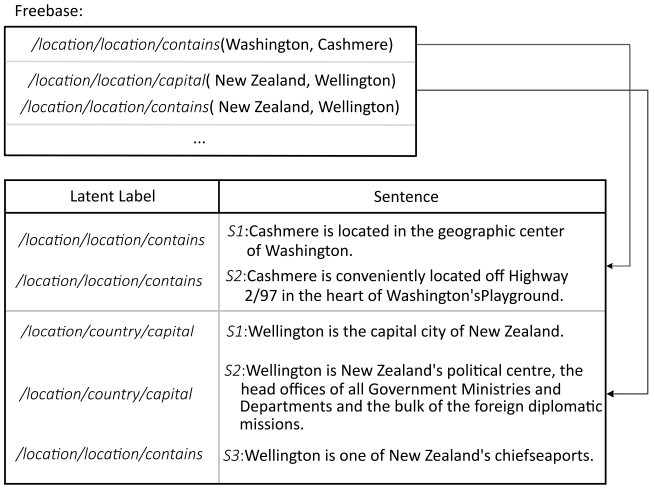


Figure 1: Training instances generated through distant supervision.

The dependencies among relations of an entity pair are common (e.g., belong to the same domain). For example, if an entity pair has the relation */people/person/place_of_birth*, it may also have the relation */people/person/nationlity* in high probability, while low probability in having the relation */location/country/captial*. Therefore, it is beneficial to consider relation dependencies while predicting the target relations of an entity pair.

Based on the above observation, in this paper, we present an encoder-decoder model for distant supervised relation extraction. Specifically, given an entity pair and its sentence bag as input, in the encoder component of our model, we employ the convolutional neural network to extract the features of the sentences in the sentence bag and merge them into a bag representation. In the decoder component of our model, we u-

*Corresponding author

tilize the long short-term memory network to model relation dependencies and predict the target relations in a sequential manner. In this sequential procedure, the relations should be predicted in a way that, the relations having more information in the bag representation are predicted earlier and used as prior knowledge for further predictions. To this end, we introduce a measure to quantify the amount of information the relations take in their sentence bag, and use such information to determine the order of these relations during model training. In doing so, the learned model will have the ability to predict the target relations in the order of their amounts of information contained in the bag representation. Additionally, we incorporate the attention mechanism into our model which dynamically adjusts the bag representation to reduce the impact of sentences whose corresponding relations have been predicted. We conduct extensive experiments on a widely used dataset released by [Riedel *et al.*, 2010]. Experimental results show that our model significantly and consistently outperforms state-of-the-art methods.

# 2 Preliminaries

## 2.1 Task Definition

Given the training data $D = (B_i, L_i)_{i=1}^{N}$, which consists of $N$ bags of sentences, where each bag $B_i$ can be represented as $z_i$ sentences such as $\{x_{i,1}, x_{i,2}, ..., x_{i,z_i}\}$. The output relations $L_i$ is a subset of all relations $\{l_1, l_2, ..., l_{n_l}\}$, where $n_l$ is the number of all relations. By training $D$, the goal of distant supervised relation extraction is to derive a proper learning model, so that the model can predict the target relations $\hat{L}$ corresponding to a given bag $\hat{B}$.

## 2.2 RNN Encoder-Decoder

In this section, we briefly describe the RNN (Recurrent Neural Network) Encoder-Decoder, proposed by [Sutskever *et al.*, 2014; Cho *et al.*, 2014], which is successfully applied to many seq2seq tasks such as machine translation [Jinchao Zhang, 2017] and syntactic parsing [Vinyals *et al.*, 2015].

In the RNN Encoder-Decoder, an encoding RNN transforms a source sequence $X = [x_1, ..., x_{T_X}]$ into a fixed length vector $\mathbf{c}$, i.e.

$$\mathbf{h}_t = f(x_t, \mathbf{h}_{t-1}); \quad \mathbf{c} = \varphi(\{\mathbf{h}_1, ..., \mathbf{h}_{T_X}\}) \tag{1}$$

where $\{\mathbf{h}_t\}$ are the RNN hidden states, $\mathbf{c}$ is the context vector which is assumed as an abstract representation of $X$ though function $\varphi$ (e.g. choosing the last state $\mathbf{h}_{T_X}$), and $f$ is a nonlinear function.

Once the source sequence is encoded, another decoding RNN generates a target sequence $Y = [y_1, ..., y_{T_Y}]$ through the following prediction model:

$$\begin{aligned} &\mathbf{s}_t = f(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{c}); \\ &p(y_t|\{y_1, ..., y_{t-1}\}, X) = g(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{c}) \end{aligned} \tag{2}$$

where $\mathbf{s}_t$ is the RNN hidden state at time $t$, $y_t$ is the predicted target symbol at time $t$ with context vector $\mathbf{c}$ and all the previously predicted target symbols $\{y_1, ..., y_{t-1}\}$. The prediction model is typically a softmax classifier over a settled vocabulary through function $g$.

### Attention Mechanism

Attention mechanism was first introduced to RNN Encoder-Decoder [Bahdanau *et al.*, 2014] to release the burden of summarizing the entire source into a fixed-length vector as context. The attention mechanism can dynamically choose context $\mathbf{c}_t$ at each time step. For example, representing $\mathbf{c}_t$ as the weighted sum of the source states $\{\mathbf{h}_t\}$,

$$\mathbf{c}_t = \sum_{\tau=1}^{T_X} \alpha_{t\tau} \mathbf{h}_\tau; \quad \alpha_{t\tau} = \frac{\exp(\eta(s_{t-1}, \mathbf{h}_\tau))}{\sum_{k=1}^{T_X} \exp(\eta(s_{t-1}, \mathbf{h}_k))}, \tag{3}$$

where $\eta$ is a function to compute the attentive strength with each source hidden state, which usually adopts a multi-layer neural network.

## 2.3 Convolutional Neural Network and Long Short-Term Memory Network

In the RNN Encoder-Decoder, RNN can be replaced with other neural networks based on the requirements of different problems.

We explore encoder-decoder model for distant supervised relation extraction, the encoder component is used to extract the features of a sentence bag, and the decoder component is used to predict the relations of the sentence bag. Since the sentences in the bag are unordered, instead of RNN, we employ CNN to obtain the sentence bag representation, which is comparatively simple, efficient and has achieved great success in sentence embedding. In the decoder component, we utilize LSTM, a variant of RNN, to predict relations in a sequential manner, which can deal with the gradient vanishing issue during RNN training and has been demonstrated to be a powerful model of learning dependencies in a sequential data.

### Convolutional Neural Network

Denote a sentence $x = \{w_1, w_2, ...w_i, ...\}$, each word $w_i$ is mapped into a real-valued vector $\mathbf{e}_{w_i} \in \mathbb{R}^{d^w}$, which is also known as word embeddings [Mikolov *et al.*, 2013].

After encoding the sentence to real-valued vectors, a convolutional layer extracts local features by sliding a window of length $l$ over the sentence and perform the convolution operation within each sliding window. The output of convolutional layer for the $i$-th sliding window is computed as:

$$\mathbf{p}_i = \mathbf{K}\mathbf{w}_{i-l+1:i} + \mathbf{b}, \tag{4}$$

where $\mathbf{w}_{i-l+1:i}$ indicates the concatenation of $l$ word embeddings within $i$-th window, $\mathbf{K} \in \mathbb{R}^{d^s \times (l \times d^w)}$ is the convolution matrix and $\mathbf{b} \in \mathbb{R}^{d^s}$ is the bias vector ($d^s$ is the dimension of output embeddings of the convolution layer).

Afterwards, all local features via a max-pooling operation and a hyperbolic tangent function to obtain a fixed-sized sentence vector. The $i$-th element of the sentence vector $\mathbf{x} \in \mathbb{R}^{d^s}$ is calculated as:

$$[\mathbf{x}]_j = \tanh(\max_i(\mathbf{p}_{ij})). \tag{5}$$

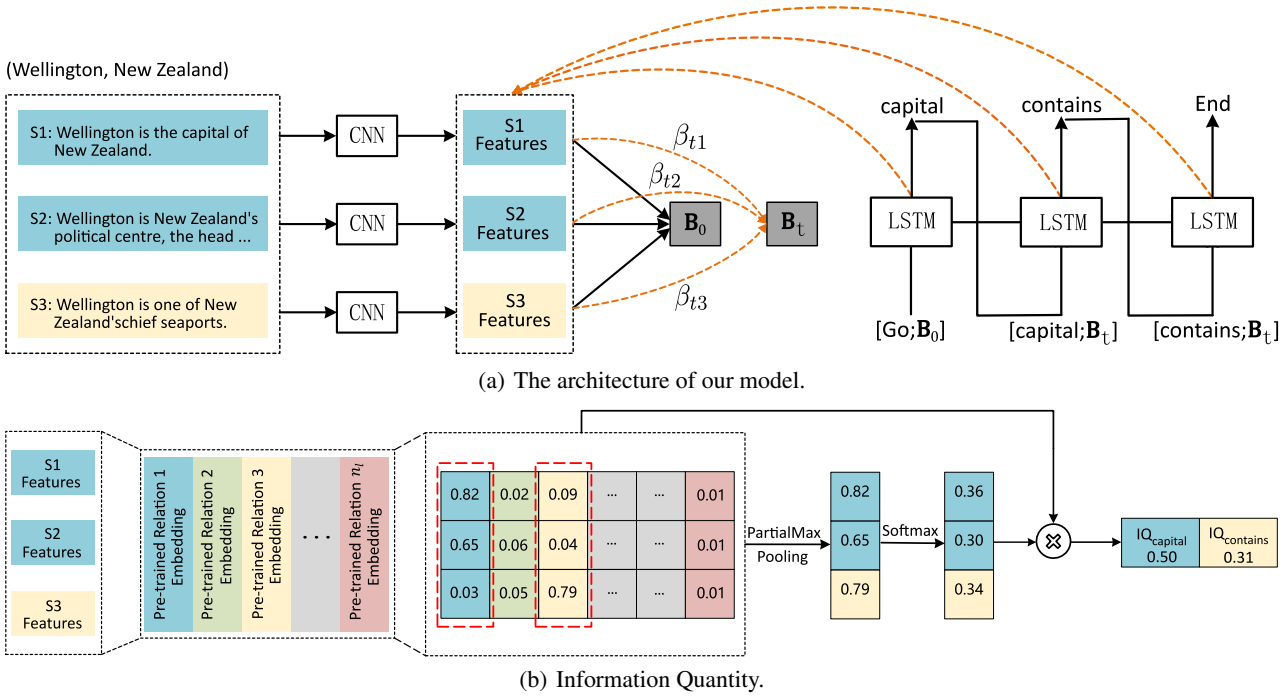


Figure 2: (a) The architecture of our model. In the encoder component, we employ CNN to extract the features of each sentence, and merges them into a bag representation. In the decoder component, we utilize LSTM to predict relations in a sequential manner, which directly models the dependencies among relations. The attention mechanism is incorporated into our model to dynamically adjust the bag representation. Where capital and contains are the abbreviation of */location/country/capital* and */location/location/contains* respectively. (b) Information Quantity. It is introduced to quantify the information contained in a sentence bag for each of its relations in model training.

**Long Short-Term Memory Network**

LSTM is capable of modeling long term dependencies in a sequential data by adding three gates to an RNN neuron: a forget gate $f$ to control whether to forget the current state; an input gate $i$ to indicate if it should read the input; an output gate $o$ to control whether to output the state.

$$\begin{aligned} i_t &= sigmoid(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\ f_t &= sigmoid(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\ o_t &= sigmoid(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\ \widetilde{c}_t &= tanh(W_{ic}x_t + b_{ic} + W_{hc}h_{t-1} + b_{hc}) \\ c_t &= f_t * c_{t-1} + i_t * \widetilde{c}_t \\ h_t &= o_t * tanh(c_t) \end{aligned} \tag{6}$$

where $c_t$, $\widetilde{c}_t$ are the cell state and candidate cell state, $h_t$ is the hidden state. The various $W$, $b$ are the weight matrices and bias vectors.

# 3 The Proposed Model

## 3.1 Model Overview

Our model consists of an encoder component and a decoder component, which takes a bag of sentences as input and gives a sequence of relations as output. The encoder component is a CNN for capturing salient meanings of each sentence and summarizing them into a vector. Vectors of all sentences are combined into a single context vector of the whole bag, which is the input of the decoder component. The decoder component is an LSTM which directly models dependencies of relations by predicting them in a sequential manner. This enables the model to use previously observed relations as prior knowledge for further predictions. The attention mechanism is additionally incorporated into our model to adjust the context vector during decoding in order to highlight sentences whose corresponding relations have not been predicted. During training, a certain order of relations is determined for each sentence bag using our proposed measure of Information Quantity for relations in a sentence bag. During testing, we only accept a bag of sentences as input, let the model automatically learn Information Quantity of each target relation and predict them in descending orders of their amounts of information. The architecture of our model is demonstrated in Figure 2.

## 3.2 Encoder

Given an entity pair and its sentence bag as input, the encoder component extracts the features of sentences by CNN firstly, and then merges them into a bag representation.

$$\mathbf{x}_i = \text{CNN}(x_i); \quad \mathbf{B} = \phi(\{\mathbf{x}_i\}) \quad i = 1, 2, 3, ..., n \tag{7}$$

where $x_i$ is the $i$-th sentence, $\mathbf{x}_i$ is the $i$-th sentence embedding obtained by CNN (Eq. (4), Eq. (5)), and $\phi$ is the function (e.g., average method) which merges the embeddings of sentences into a bag representation.

### 3.3 Decoder

By using the encoder component, we can get the bag representation. In this section, we introduce how to model the relation dependencies and predict the target relations of the given entity pair by LSTM in a sequential manner.

**Modeling Dependencies among Relations by LSTM**

Given a bag representation $\mathbf{B}$, LSTM predicts a sequence of relations $\{y_1, y_2, ...y_t...\}$. The predicted relation $y_t$ at time $t$ is computed by:

$$\begin{aligned} &\mathbf{s}_t = LSTM(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{B}) \\ &p(y_t = l_j | \{y_1, ..., y_{t-1}\}, \mathbf{B}) = \frac{\exp(l_j \mathbf{T}\mathbf{s}_t)}{\sum_{l_i \in L} \exp(l_i \mathbf{T}\mathbf{s}_t)} \end{aligned} \tag{8}$$

where $y_t$ is the predicted relation at time $t$, $l_i$ is the $i$-th relation, $L$ is the relation set, $\mathbf{T}$ is a transformation matrix and $\mathbf{s}_t$ is the hidden state computed by Eq. (6).

Since LSTM predicts the relations of an entity pair as a series of single relations (in a sequential manner) by the probability of a relation conditioned on previously observed relations, which can model the conditional dependencies among these relations. Moreover, the predicted relation at each time step is also used as the input of the next time step, which can provide prior knowledge for the prediction of the next relation.

### 3.4 Attention Mechanism

After a relation is predicted at each time step, our model should pay more attention to those sentences whose corresponding relations have not been predicted. Therefore, we incorporate the attention mechanism into our model to dynamically adjust the bag representation, which reduces the impact of sentences whose corresponding relations have been predicted, and highlights the sentences which have not been covered. The bag representation at time $t$ is computed by:

$$\mathbf{B}_t = \sum_{i=1}^{n} \beta_{ti} \mathbf{x}_i; \quad \beta_{ti} = \frac{\exp(\eta(s_{t-1}, \mathbf{x}_i))}{\sum_{k=1}^{n} \exp(\eta(s_{t-1}, \mathbf{x}_k))} \tag{9}$$

where $\beta_{ti}$ is the weight of each sentence, $\eta$ is a neural network. The score is based on the LSTM hidden state $\mathbf{s}_{t-1}$ and the $i$-th sentence embedding $\mathbf{x}_i$.

### 3.5 Training and Testing

This subsection first explains our method for determining the order of relations during training, and then gives the loss function during training. For both training and testing, the bag representation should be initialized before the first relation can be predicted, which is explained.

In the training phase, we first determine the order of relations for a sentence bag, thus formalize each training example as a sentence bag and a relation sequence. Then, we introduce the loss function which is constructed using the determined order.

**Determining the Order of Relations**

Given a sentence bag and its relations, the information contained in the sentence bag for each of its relations is typically not equal. The more information contained in the sentence bag for a relation, the more earlier the relation should be predicted. Therefore, we propose a measure, called Information Quantity, to quantify the information contained in a sentence bag for each of its relations, and determine the order of relations in descending order according to the amounts of information, as illustrated in Figure 2(b). Information Quantity calculates the amount of information of a relation by the following two stages:

(1) in the first stage, we compute the matching scores between each input sentence and each relation as:

$$\mathbf{W} = \mathbf{X}\mathbf{M}, \tag{10}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d^s}$ is the matrix of sentences embeddings ($n$ is the number of sentences in the given sentence bag, $d^s$ is the sentence embedding dimension), and $\mathbf{M} \in \mathbb{R}^{d^s \times n_l}$ is the representation matrix of pre-trained relations ($d^s$ is the relation representation dimension, $n_l$ is the total number of relations). To measure the matching score under a uniform standard, we normalize each row through a softmax layer.

(2) in the second stage, inspired by [Feng and Zhou, 2017], we define Partial Max-pooling method to compute the weight of each input sentence based on the matching matrix $\mathbf{W}$. Since we are more concerned about the Information Quantity of positive relations the sentence bag contains. The Partial Max-pooling method is computed as:

$$e_i = \max(\mathbf{W}_{ij}) \quad i = 1, 2, ..., n, j \in L^+, \tag{11}$$

where $W_{ij}$ represents the matching scores between the $i$-th sentence and the $j$-th relation. $j \in L^+$ is the positive relation of the given sentence bag. Afterwards, we normalize $e_i$ through a softmax layer to obtain the weight of each input sentence:

$$\alpha_i = \frac{\exp(e_i)}{\sum_{k=1}^{n} \exp(e_k)} \quad i = 1, 2, 3, ..., n. \tag{12}$$

Thereby, given a sentence bag and its relations, the information contained in the sentence bag for each of its relations $j$ can be computed as:

$$IQ_j = \sum_{i=1}^{n} \alpha_i \mathbf{W}_{ij} \quad i = 1, 2, ..., n, j \in L^+. \tag{13}$$

**Initializing Bag Representation**

Two methods are introduced to initialize the bag representation. One is average method, which treats all sentences equally. The other is Partial Max-pooling method, which could alleviate the wrong label problem of distant supervision.

1) Average Method. The average method considers each sentence equally in the contribution of the bag representation. It initializes the bag representation as:

$$\mathbf{B} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i. \tag{14}$$

2) Partial Max-pooling Method. The partial max-pooling method is originally defined for calculating Information Quantity. Here, we use it to initialize the bag representation. Since partial max-pooling method computes the weight

of each sentence in a sentence bag, which can alleviate the wrong label problem of distant supervision. The bag representation is initialized as a weighted sum of sentences:

$$\mathbf{B} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i, \tag{15}$$

where weight $\alpha_i$ is calculated by Eq.(12).

**Loss Function**

We define the loss function using mean squared error at the bag level as follows:

$$Loss_{squared} = \sum_{t=1}^{n_B} \sum_{i=1}^{n_l} (q_{ti} - p_{ti})^2, \tag{16}$$

where $n_B$ is the number of relations corresponding to a sentence bag, $n_l$ is the number of all relations, $q_{ti} \in \{0, 1\}$ is the true value on relation $i$ at time $t$, $p_{ti}$ is the predicted probability of the $i$-th relation at time $t$. The loss function is optimized with mini-batch stochastic gradient descent (SGD). In order to be able to automatically end the relations prediction in testing phase, we add a special symbol <End> at the end of the relations of each sentence bag in the training phase.

In the testing phase, given an entity pair and its sentence bag as input, in the encoder component, we use CNN to extract the features of sentences and then use average method to initialize the bag representation. In the decoder component, we utilize LSTM incorporated with attention mechanism to predict relations of the entity pair until encounter <End> or the 5 step.

# 4 Experiments

## 4.1 Dataset and Evaluation Metrics

We evaluate our model on a widely used dataset[1] released by (Riedel, Yao and McCallum 2010). This dataset was generated by aligning Freebase relations with the New York Times corpus (NYT), where sentences of year 2005 and 2006 are used for training and sentences of year 2007 are used for testing. Following previous work [Feng and Zhou, 2017], we evaluate our model on the held-out evaluation. The held-out evaluation compares the extracted facts with those in Freebase, which provides an approximate measure of precision without requiring expensive human evaluation. We evaluate the performance of each model with Precision-Recall curves and Precision@N (P@N) metric.

## 4.2 Experimental Settings

**Pre-trained Relation Embedding**

Inspired by [Wang *et al.*, 2016] who adopts the descending order of label frequency as their order in the training phase for multi-label image classification. We adopt the descending order of relation frequency to pre-train relation embedding in relation extraction, and use the pre-trained relation embedding to calculate the Information Quantity. In this way, the Information Quantity will more accurately reflect the information contained in a sentence bag for its relations.

[1] http://iesl.cs.umass.edu/riedel/ecml/

In order to analyze the effects of partial max-pooling initialize method, Information Quantity and attention mechanism, we train our model in four settings.

**Ave+Freq** uses average method to obtain bag representation and the descending order of relation frequency to determine the order of relations for each sentence bag in the training data.

**PartialMax+Freq** uses **Ave+Freq** to pre-train relation embedding, then uses partial max-pooling method to obtain bag representation, and uses the descending order of relation frequency to determine the order of relations for each sentence bag in the training data.

**PartialMax+IQ** has the same settings with **PartialMax+Freq**, except that it replaces relation frequency by Information Quantity.

**PartialMax+IQ+ATT** has the same settings with **PartialMax+IQ**, except that it incorporates the attention mechanism during decoding.

**Parameter Settings**

We use three-fold validation to tune our model on the training data. We select learning rate $\lambda$ among $\{0.01, 0.02, 0.03\}$, sliding window size $w$ among $\{3, 5, 7\}$, sentence embedding size $d^s$ among $\{150, 160, ..., 300\}$, and batch size $B$ among $\{40, 100, 160\}$. Following [Zeng *et al.*, 2015], we set the dropout rate to 0.5. Table 1 shows all parameters values in the experiments.

| Parameter | Value |
|---|---|
| Window size $l$ | 3 |
| Word embedding dimension $d^w$ | 50 |
| Sentence embedding size $d^s$ | 230 |
| Batch size $B$ | 100 |
| Learning rate $\lambda$ | 0.01 |
| Dropout probability $p$ | 0.5 |

Table 1: Parameter settings

**Compared Methods**

We compare our model with the following neural-based methods.

**PCNN** [Zeng *et al.*, 2015] is the original distant supervision model with neural networks for relation extraction.

**MIMLCNN** [Jiang *et al.*, 2016] is a neural network method with multi-instance multi-label learning for distant supervised relation extraction.

**CNN+ATT** [Lin *et al.*, 2016] is a sentence-level attention model, which can alleviate the wrong label problem in distant supervised relation extraction.

Ye et al. [Ye *et al.*, 2017] propose a model to jointly extract relations, and introduce three loss functions **Rank+AVE**, **Rank+ATT** and **Rank+ExATT**, which achieves state-of-the-art performance.

## 4.3 Experimental Results

We evaluate the performance of our model in the four settings and the compared models.

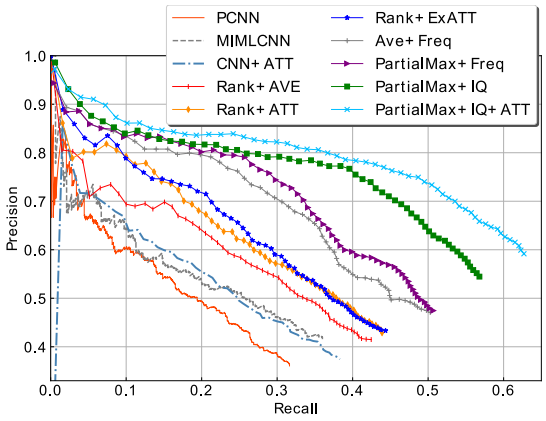


Figure 3: Precision-recall curves of our model in four settings and compared methods.

| P@N(%) | 100 | 200 | 300 | 400 | 500 | Ave. |
|---|---|---|---|---|---|---|
| PCNN | 0.76 | 0.71 | 0.67 | 0.65 | 0.63 | 0.68 |
| MIMLCNN | 0.74 | 0.70 | 0.66 | 0.63 | 0.61 | 0.67 |
| CNN+ATT | 0.79 | 0.74 | 0.72 | 0.70 | 0.68 | 0.73 |
| Rank+AVE | 0.80 | 0.76 | 0.74 | 0.73 | 0.72 | 0.75 |
| Rank+ATT | 0.89 | 0.85 | 0.83 | 0.81 | 0.79 | 0.83 |
| Rank+ExATT | 0.81 | 0.81 | 0.80 | 0.79 | 0.77 | 0.80 |
| Ave+Freq | 0.90 | 0.88 | 0.87 | 0.85 | 0.84 | 0.86 |
| PartialMax+Freq | 0.89 | 0.88 | 0.86 | 0.86 | 0.85 | 0.87 |
| PartialMax+IQ | **0.94** | **0.91** | 0.89 | **0.88** | 0.86 | **0.90** |
| PartialMax+IQ+ATT | 0.91 | **0.91** | **0.90** | **0.88** | **0.87** | **0.90** |

Table 2: Precisions for top 100, 200, 300, 400, 500 and average of them for our model in four settings and compared methods.

Experimental results are shown in Figure 3 and Table 2. From the results, we observe that: (1) Our model outperforms state-of-the-art methods in all four settings, showing that encoder-decoder model is promising for distant supervised relation extraction; (2) PartialMax+Freq achieves a higher precision than Ave+Freq, which indicates that Partial max-pooling method can alleviate the wrong label problem in distant supervised relation extraction; (3) PartialMax+IQ outperforms Ave+Freq and PartialMax+Freq significantly, from which we can conclude that measuring the information contained in a sentence bag for its relations by Information Quantity can greatly improve the prediction accuracy; (4) PartialMax+IQ+ATT achieves the best performance comparing with the other three settings and all the comparable methods, which shows that dynamically adjusting the bag representation by attention mechanism during decoding is helpful for the extraction of relations. (5) Comparing Ave+Freq with Rank+AVE, we can see that directly modeling dependency is quite effective. Ave+Freq use the same method as Rank+AVE to obtain bag representations by different methods to capture dependencies of relations. By comparing their results, we can see that the direct modeling of dependencies by LSTM is more effective that the indirect modeling through pairwise learning-to-rank. (6) In precision-recall curves, our model in each setting keeps a relatively high precision value (larger than 0.7) when recall is less than 0.3, and with recall increased, the precision value decreased slowly. The reason is that given an entity pair and its sentence bag as input, our model iteratively predicts relations in the following way: predicting the relation with the most information first, and leveraging relation dependencies to enhance predictions of those with less information.

## 5 Related Work

Distant supervision is an efficient approach that automatically generates training instances for relation extraction [Mintz *et al.*, 2009; Riedel *et al.*, 2010; Hoffmann *et al.*, 2011; Surdeanu *et al.*, 2012]. Since an entity pair may have one or multiple relations, and there are also multiple sentences corresponding to an entity pair, distant supervised relation extraction can be formalized as multi-instance multi-label learning problem.

Recently, learning the dependencies among relations for distant supervised relation extraction has gained much interest. Jiang et al. [Jiang *et al.*, 2016] handle the dependency simply by a shared bag representation, and apply sigmoid function to calculate the probability of each relation corresponding to the bag independently. Ye et al. [Ye *et al.*, 2017] jointly extract multiple relations of one entity pair and adopt pairwise learning to rank to capture the co-occurrence dependency between relations. However, in the testing phase they predict relations independently, which may lead to poor performance. Feng et al. [Feng *et al.*, 2017] develop a neural model with two memory networks, one for learning the weight of each context word, the other is used to capture the dependencies between relations. Also, the model predicts relations using multiple binary classifier individually, which cannot explicitly take advantage of the dependencies among relations. Different from the above methods, we propose an encoder-decoder model for distant supervised relation extraction, where an LSTM is used as the decoder for modeling dependencies among relations of an entity pair. Our model successively predict relations of the given entity pair in a sequential manner, which can explicitly model dependencies among these relations both in training and testing. To the best of our knowledge, this is the first effort to explore encoder-decoder model for distant supervised relation extraction.

## 6 Conclusion

In this paper, we present a simple yet effective encoder-decoder model for distant supervised relation extraction. Given the sentence bag of an entity pair as input, the CNN encoder extracts sentence features and merge them into a bag representation. While the LSTM decoder leverages the dependencies among the relations by predicting them in a sequential manner. To enable the sequential prediction of relations, we introduce a measure to quantify the amounts of information contained in a sentence bag for its relations, which are used to determine relation orders during training to let the model predict relations in a descending order of their amounts of information. Additionally, the attention mechanism is incorporated into our model to dynamically adjust the bag representation. Experimental results show that our model significantly outperforms state-of-the-art methods.

## Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant 61502047. We acknowledge anonymous reviewers for their valuable comments.

## References

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735, 2007.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*, 2014.

[Bollacker *et al.*, 2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[Carlson *et al.*, 2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3, 2010.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[Feng and Zhou, 2017] Ji Feng and Zhi-Hua Zhou. Deep miml network. In *AAAI*, pages 1884–1890, 2017.

[Feng *et al.*, 2017] Xiaocheng Feng, Jiang Guo, Bing Qin, Ting Liu, and Yongjie Liu. Effective deep memory networks for distant supervised relation extraction. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, pages 19–25, 2017.

[Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics, 2011.

[Jiang *et al.*, 2016] Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In *COLING*, pages 1471–1480, 2016.

[Jinchao Zhang, 2017] Jie Zhou Jinchao Zhang, Qun Liu. Me-md: An effective framework for neural machine translation with multiple encoders and decoders. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3392–3398, 2017.

[Lin *et al.*, 2016] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *ACL (1)*, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

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

[Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases*, pages 148–163, 2010.

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

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[Vinyals *et al.*, 2015] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781, 2015.

[Wang *et al.*, 2016] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2285–2294, 2016.

[Ye *et al.*, 2017] Hai Ye, Wenhan Chao, Zhunchen Luo, and Zhoujun Li. Jointly extracting relations with class ties via effective deep ranking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1810–1820, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[Zeng *et al.*, 2015] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Emnlp*, pages 1753–1762, 2015.