A Fine-grained and Noise-aware Method for Neural Relation Extraction

Jianfeng Qu
College of Computer Science and Technology, Jilin University, Changchun, China
Key laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Minstry of Education, Changchun, China
qujfjlu@163.com

Wen Hua
School of Information Technology and Electrical Engineering, The University of Queensland, Australia
w.hua@uq.edu.au

Dantong Ouyang*
College of Computer Science and Technology, Jilin University, Changchun, China
Key laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Minstry of Education, Changchun, China
ouyd@jlu.edu.cn

Xiaofang Zhou
School of Information Technology and Electrical Engineering, The University of Queensland, Australia
zxf@itee.uq.edu.au

Ximing Li
College of Computer Science and Technology, Jilin University, Changchun, China
ximingli@jlu.edu.cn

ABSTRACT

Distant supervision is an efficient way to generate large-scale training data for relation extraction without human efforts. However, a coin has two sides. The automatically annotated labels for training data are problematic, which can be summarized as multi-instance multi-label problem and coarse-grained (bag-level) supervised signal. To address these problems, we propose two reasonable assumptions and craft reinforcement learning to capture the expressive sentence for each relation mentioned in a bag. More specifically, we extend the original expressed-at-least-once assumption to multi-label level, and introduce a novel express-at-most-one assumption. Besides, we design a fine-grained reward function, and model the sentence selection process as an auction where different relations for a bag need to compete together to achieve the possession of a specific sentence based on its expressiveness. In this way, our model can be dynamically self-adapted, and eventually implements the accurate one-to-one mapping from a relation label to its chosen expressive sentence, which serves as training instances for the extractor. The experimental results on a public dataset demonstrate that our model constantly and substantially outperforms current state-of-the-art methods for relation extraction.

CCS CONCEPTS

• **Information systems → Information extraction**.

*Corresponding author

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357997

KEYWORDS

Distant supervision; relation extraction; multi-instance multi-label; coarse-grained supervised signal; reinforcement learning

ACM Reference Format:
Jianfeng Qu, Wen Hua, Dantong Ouyang, Xiaofang Zhou, and Ximing Li. 2019. A Fine-grained and Noise-aware Method for Neural Relation Extraction. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357997

1 INTRODUCTION

Knowledge graph (KG), which stores relational facts in a graph format, has exhibited its indispensable effect in question-answering (QA) [27, 28] and information retrieval (IR) [5]. In the era of knowledge explosion, a great deal of new information appears with unstructured format on newswire and web pages. To automatically structure the information and improve the completeness of KG, researchers pay more attention to the task of relation extraction.

Relation extraction intends to identify relationships between two entities of interest with reference to plain texts mentioning them. Take the following sentence with two target entities, "Donald Trump" and "United States", as an example:

Donald Trump is the 45th President of the *United States*.

A well-trained relation extractor should be able to identify the triple *President_of (Donald Trump, United States)* in which *President_of* is the relationship between the entity pair *(Donald Trump, United States)*. Supervised relation extraction heavily relies on human-annotated data in order to achieve outstanding performance [6]. Manually labelled datasets are limited in size and domain-specific, preventing large-scale supervised relation extraction.

One way to circumvent this issue is distant supervision proposed by [14], which automatically generates training data through heuristic alignment between a knowledge base (e.g., Freebase) and plain texts (e.g., New York Times). Specifically, the alignment is based

Table 1: The alignment process between KB and texts

Relations in KB	*Place_of_birth (Donald Trump, United States)* *President_of (Donald Trump, United States)*
Sentences in plain texts	S1: *Donald Trump* is the 45th President of the *United States. (President_of)* S2: *Donald Trump* was born in the *United States. (Place_of_birth)* S3: *Donald Trump* believes the *United States* has incredible potential.(-)

on a strong assumption: *if an entity pair participates in a relation in the knowledge base (KB), then all the sentences mentioning these two entities should express that relation between them.* Table 1 shows an example of the alignment process. According to the assumption, S1, S2 and S3 can be treated as training evidence for relations *Place_of_birth* and *President_of*, and hence all these sentences constitute **a training bag** for this entity pair with *Place_of_birth* and *President_of* as the relation labels.

Obviously, distant supervision strategy is dubious, leading to noise label problem and coarse-grained signal problem in training instances:

Challenge 1: Noise data can be summarized as **multi-instance multi-label problem (MIML)** [25]. Multi-instance denotes that some sentences do not express any relation between the target entities (e.g., S3) while multi-label indicates that an entity pair might be involved in multiple relations in the KB (e.g., *Place_of_birth* and *President_of* for *(Donald Trump, United States)*). Alleviating both multi-instance and multi-label problems together poses a great challenge in distant supervision. Existing neural relation extraction models [4, 13, 30, 32] *only treat noise as multi-instance problem*, ignoring that multi-label problem is also grievous in distant supervision. [25] reports that 7.5% of training bags contain more than one relation when aligning Freebase with New York Times.

Challenge 2: There is **no fine-grained sentence-level supervised signal** to train an accurate relation extractor in distant supervision. *The assigned relation labels are annotated at bag-level (a set of sentences) instead of sentence-level.* In other words, the mapping is from a set of sentences to a label. This type of mapping, however, is vague and cannot specifically point out which sentence in the set explicitly describes the relation label corresponding to the set. Consequently, excessive fuzzy information submerges clear information in the sentence set, misleading the extractor and making the extraction ineffective.

Contributions: To address these challenges, in this paper, we take advantage of reinforcement learning paradigm [23], and craft it based on our demands so as to train an effective relation extractor.

Before constructing the reinforcement learning model, we propose two reasonable assumptions for every training bag to obtain sentence-level evidence:

- **Expressed-at-least-once** for each relation of the bag, which means at least one sentence in the bag mentions a particular relation of the bag. This assumption is similar to [21] but we extend it to multiple labels.
- **Express-at-most-one** for each sentence in the bag, which means one sentence expresses at most one relation of the bag.

Given a training bag, our model intends to *discover the most informative sentence from candidates for every relation label (i.e., the first assumption). Additionally, different relations need to compete with each other (auction mechanism) if they choose the same sentence as training evidence (i.e., the second assumption).*

To this end, we develop a **value-based reinforcement learning framework** for distantly supervised relation extraction. Specifically, we devise the main components in reinforcement learning, namely state, action and reward, to address the aforementioned challenges. An agent needs to decide for each relation whether to select a sentence as its training evidence (multi-label problem) or discard it due to limited correlation (multi-instance problem). We combine entity information, sentence information and **confidence information**, and encode them into each state to provide diverse evidence for the agent to make decisions. The action in our model consists of two parts, taking into account accuracy and efficiency. Moreover, we propose a **fine-grained sentence-level reward function** to provide intensive feedbacks to the agent, so that the model can eventually achieve the maximum accumulative rewards. In this way, our model is able to *simultaneously solve multi-instance and multi-label problem without prior knowledge, and generates the one-to-one mapping from a relation to an expressive sentence for each training bag (i.e., fine-grained supervised signal).* It is worth noting that we **jointly train** the reinforcement learning and the relation extractor, making our model dynamically self-adapted and optimal.

In summary, we make the following major contributions in this work:

- We develop value-based reinforcement learning to simultaneously solve multi-instance and multi-label problem in neural relation extraction.
- We craft state, reward and action in the model to automatically capture fine-grained sentence-level signal which can be regarded as high-quality evidence to train an effective relation extractor.
- Our model constantly and substantially outperforms the current state-of-the-art methods for relation extraction on the widely-used dataset.

The remainder of this paper is organized as follows: We introduce preliminaries and the technical details of our proposed model in Section 2 and Section 3 respectively; Experimental results on the public dataset are presented in Section 4, followed by a discussion of related work in Section 5 and a brief conclusion in Section 6.

2 PRELIMINARIES

Definition 2.1 (Relation Extractor). Given a sentence S that mentions the target entities $\langle e_1, e_2 \rangle$, as well as a collection of predefined relations $(r_1, \cdots, r_l)$, a relation extractor is a trained model that calculates, for each relation r_i, a confidence score $p(r_i|S)$ to estimate whether S can reflect the relation r_i between e_1 and e_2.

There are two fundamental aspects we need to consider in order to build an effective relation extractor:

- How to encode the sentence in a format that can be understood by the extractor?
- How to construct an accurate and large-scale dataset to train the extractor?

We adopt one of the most effective neural models for sentence encoding, i.e., convolutional neural networks (CNN) [31]. We briefly summarize the workflow of CNN in this section. Actually, the most important contribution in this work is to automatically eliminate noise in the training data with reinforcement learning. We elaborate on the technical details in Section 3.

2.1 Sentence Encoding with CNN

Let $S = \{w_1, w_2, w_3, \cdots, w_n\}$ be a sentence of length n, where w_i is the i-th token in the sentence. We first transform each word into low-dimensional real-valued vectors. Follow [13, 30], the embedding representation for a word consists of two parts: word embeddings and position embeddings. Specifically, word embeddings are obtained by using word2vec tool[1]. As for position embeddings, it is defined by the relative distance between two target entities and randomly initialized for each specific number. Then S can be denoted as a matrix $\mathbb{R}^{n\times d}$ in which $d = dw + 2 * dp$, dw and dp are the dimension of word embeddings and position embeddings, respectively.

As the evidence used to predict relations might appear anywhere in the sentence, we employ convolutional filters to extract every local feature. For a matrix $\mathbb{R}^{n\times d}$, a filter $W \in \mathbb{R}^{l\times d}$ slides over the matrix to get every part features by dot product with $w_{i-l+1:i}$, where l is the size of convolutional window and $w_{i-l+1:i}$ is a concatenation from w_{i-l+1} to $w_i (1 \le i \le n+l-1)$. We will pad all-zero values for out-of-ranege w_i (i.e., $i < 1$ or $i > n$). Finally, similar to [9, 13], we utilize max pooling layer to take the maximum value of each filter and use the Symbol $\boldsymbol{S}$ to indicate the convolutional encoding for S.

Given $\boldsymbol{S}$, we can define a naive extractor using softmax function:

$$p(r_i|S;\theta) = \frac{exp(o^{r_i})}{\sum_{k=1}^{|r|} exp(o^{r_k})}$$

where $|r|$ is the total number of relations in the dataset and $o = \boldsymbol{W_r}(\boldsymbol{S} \odot \boldsymbol{D}) + \boldsymbol{b_r}$. $\boldsymbol{W_r}$ is a transform matrix from features to relations and each value in $\boldsymbol{W_r}$ represents the weight of the corresponding feature for predicting a specific relation. $\boldsymbol{D}$ is the dropout layer for generalization and $\boldsymbol{b_r}$ is the bias vector.

In the remainder of this article, we use "the extractor" to represent the combination of CNN encoding for sentences and the softmax layer.

3 MODEL

As discussed in Section 1, distant supervision is an efficient way to construct large-scale training data for relation extraction. However, it always encounters noise problem in the training instances and hence impairs the effectiveness of learned extractor. Let $B = \{\langle e_1, e_2\rangle, (r_1, \cdots, r_l), \{S_1, \cdots, S_n\}\}$ be a training bag, where $\langle e_1, e_2\rangle$ is an entity pair, $(r_1, \cdots, r_l)$ are the relations that link the entity pair in the KB, and $\{S_1, \cdots, S_n\}$ are the sentences from plain texts which mention this entity pair. Our objective in this work is to automatically eliminate noise in the training data by figuring out the most expressive sentence S_v for each relation r_i in B, which can serve as a positive training instance and promote the extractor's ability to recognize relation r_i.

[1] https://code.google.com/p/word2vec/

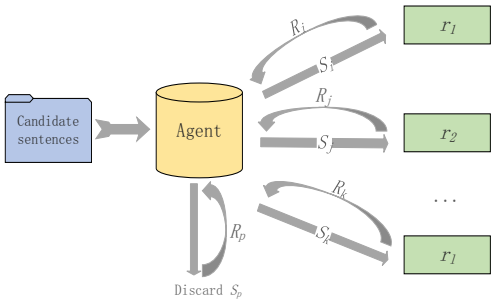

Figure 1: Framework of the RL model

In particularly, we build a value-based reinforcement learning algorithm [26] and craft it to meet our demand. Figure 1 shows the framework of the reinforcement learning model. Actually, the discovery process can be cast as an auction process. We regard each relation as a bidder who seeks for a specific sentence that can accurately express the desired relation, while an agent (i.e., auctioneer) will continuously allocate candidate sentences in the bag to the bidder who will give the higher reward (multi-label problem) or just discard it (multi-instance problem). In this way, we hope that the agent in reinforcement learning could make sequential actions with the advent of sentences in a training bag, and eventually picks out one representative sentence for each relation (expressed-at-least-once). Also, there are no overlapping sentences between different relations (express-at-most-one).

Next, we introduce the main components of our value-based reinforcement learning: state, action, and reward.

3.1 State

In reinforcement learning, the state is designed to provide abundant evidence for the agent to take optimal actions. To this end, here, our state is constituted by: target entity pair, encoding for the previously chosen sentence S_{pre}, confidence scores of S_{pre} w.r.t all the relations (not only restrict to the anchored relations of the bag), encoding for the currently processing sentence S_{cur}, confidence scores of S_{cur} w.r.t all the relations. We describe these elements below:

- *target entity pair*: embedding representation for entities implicitly contains clues about the relations between them. In the task of knowledge graph completion, a relation r between an entity pair of interest is formulated as a translation from the head entity e_1 to the tail entity e_2: $e_1 + r = e_2$ [2, 8, 12, 18]. These works have sufficiently proved this semantic information embedded in word embeddings. Inspired by their works, we concatenate e_1 and e_2: $[e_1; e_2]$, where the embedding representation for entities is acquired by a pre-trained word2vec tool, to serve as entity clues for the agent.

- *encoding for S_{pre}*: neural networks exhibit outstanding ability in automatic encoding for sentences [1, 24, 31]. Since it is not the focus of this paper, we directly adopt one of the state-of-the-art encodings: CNN (detailed in *Preliminaries*). Let $\boldsymbol{S_{pre}}$ be the convolutional encoding for S_{pre}. Certainly, the model is of enough capacity to use any existing sentence encodings, including piecewise convolutional neural networks(PCNN) [30], long short-term memory(LSTM) [15]. For conciseness, we don't integrate them all into our model.
- *confidence scores for S_{pre}*: for this part, we utilize the current extractor (softmax function)[2] to get the confidence scores on all possible relations. In the experiment, the total number of relations in the widely used dataset is 53, including 52 positive relations and NA (no relation between entities of interest). The confidence information is able to provide straightforward information from the current extractor and facilitate the process of action decision, especially when the processing sentence is too long. We will demonstrate this utility in the experiment part (*Case Study*).
- *encoding for S_{cur}*: similar to S_{pre}, we employ convolutional encoding for currently processing sentence, denoted as $\boldsymbol{S_{cur}}$.
- *confidence scores for S_{cur}*: the confidence scores are acquired by the current relation extractor when $\boldsymbol{S_{cur}}$ serves as the sentence feature.

On the basis of these analyses, we represent the state as $St = [e_1; e_2; \boldsymbol{S_{pre}}; p(r|S_{pre}, \theta); \boldsymbol{S_{cur}}; p(r|S_{cur}, \theta)]$, where the symbol $[x; y]$ means the concatenation operation and $p(r|S, \theta) \in \mathbb{R}^{53}$ indicates the confidence scores of all relations with reference to the sentence S using the current parameter θ.

3.2 Action

Given a training bag $B = \{\langle e_1, e_2\rangle, (r_1, \cdots, r_l), \{S_1, \cdots, S_n\}\}$, the sentences in the bag will be fetched out one by one as a candidate description of the relation r_i between the target entities. In order to address multi-label problem, our model creates l episodes, and each of them represents one specific relation. In this way, the agent only needs to make decision that whether to adopt the current sentence to replace the previously chosen one for its anchored relation. We denote the first part of the action controlled by our agent as U, which takes value in $\{0, 1\}$ and "1" indicates update action of the chosen sentence using the current one while "0" means insistence on the previously chosen sentence. Generally, one single sentence expresses at most one existing relation among $\{r_1, r_2, \cdots, r_l\}$[3]. Hence, the agents in different episodes for one bag have to compete together to earn the possession of their confident sentences (detailed analysis of this circumstance is conducted later).

Another observation is that the quantities of sentences in some training bags are greatly redundant. Some bags have aligned more than 500 sentences that mention their entity pairs (e.g., (*China, Beijing*), (*Fort Green, Brooklyn*)). To improve the efficiency of the entire model, we build the second part of the action, which is denoted as $P \in \{0, 1\}$, determining whether to stop the search action. In one episode, there are two cases that result in the termination of the episode: (1) all sentences in the bag have been processed by the agent; (2) the agent decides to take the stop action (i.e., $P = 1$), which is expected to occur when the agent has adequate confidence that it has picked out the expressive sentence for its target relation.

In summary, the action of our agent consists of two parts: $A = \{U, P\}$. In our model, since we utilize value-based reinforcement learning, the agent determines the optimal action according to $Q(St, A)$ [26], including $Q(St, U)$ and $Q(St, P)$, and takes the action A^* that equals $argmax_A Q(St, A)$ (Q-function serves as a navigator, guiding the agent to achieve higher accumulate reward). And we define Q-function using a multilayer perceptron (MLP) as:

$$Q(St, A; \eta) = \begin{cases} Q(St, U) = f(\boldsymbol{W_u}(f(\boldsymbol{W} \cdot St)) + \boldsymbol{b_u}) \\ \\ Q(St, P) = f(\boldsymbol{W_p}(f(\boldsymbol{W} \cdot St)) + \boldsymbol{b_p}) \end{cases}$$

where $f(\,)$ is a non-linear function, such as *Tanh* and *Relu*, $\boldsymbol{W}$ is an intermediate matrix that further encodes state features, $\boldsymbol{W_u}$ and $\boldsymbol{W_p}$ serve as mapping matrix from state to action, and $\boldsymbol{b_u}$ and $\boldsymbol{b_p}$ are the bias vectors.

3.3 Reward

The objective of the entire model is to seek the relatively effective sentence for each relation in a bag so that the chosen sentence's features can boost the extractor's ability to recognize the specific relation. Therefore, we are more concerned about sentence-level rewards instead of bag-level rewards [4, 32], and carefully design the reward function to subtly reflect the changes that each action brings with respect to the target relation.

Particularly, we resort to the current extractor to compute the fine-grained rewards. When sentences in the bag come one after another, the agent should make consequential actions. The reward for each action is mainly based on the degree of deviation for the updated sentence to the requirement of the target relation with reference to the already chosen sentence. Eventually, the goal of the agent in one episode is to maximize the total reward which is composed of all the received rewards during the active period.

Let T be the terminal of an episode that can be triggered by two situations (described in *Action* part) and r_t is the target relation anchored in this episode. According to the above analysis, we define the reward function $Re(St, A)$ as above.

In this equation, we carefully produce the impact of changes brought by the specific action in each case with reference to the target relation. For example, in the second case (i.e., the episode continues to run, the agent reserves the previously chosen sentence and the most confident relation of S_{cur} is the target one), the reward equals the difference in confidence between the previously selected sentence and the currently processing sentence for the target relation. In the fifth case (i.e., the episode stops, the agent chooses the current sentence and the most confident relation of S_{cur} is not the target one), the reward equals that the confidence score of S_{cur} for the target relation subtracts the score for the most confident relation. Obviously, the calculated reward is negative, which will make the agent aware of its unwise decisions. For conciseness, we

[2]The word "current extractor" denotes the current values of the parameters used for the extractor. During training iterations, the parameter will be continuously updated.
[3]KB often suffers from incompleteness, that is, some entity pairs may have other relations but not included in the KB. We regard all these relations that are not considered in this paper as NA.

$$Re(St,A)=\begin{cases} p(r_t|S_{cur};\theta)-p(r_t|S_{pre};\theta) & (\neg T)\wedge(U=1)\\ p(r_t|S_{pre};\theta)-p(r_t|S_{cur};\theta) & (\neg T)\wedge(U=0)\wedge(\arg\max_r p(r|S_{cur};\theta)=r_t)\\ 0 & (\neg T)\wedge(U=0)\wedge(\arg\max_r p(r|S_{cur};\theta)\neq rt)\\ p(r_t|S_{cur};\theta) & (T)\wedge(U=1)\wedge(\arg\max_r p(r|S_{cur};\theta)=rt)\\ p(r_t|S_{cur};\theta)-\max(p(r|S_{cur};\theta)) & (T)\wedge(U=1)\wedge(\arg\max_r p(r|S_{cur};\theta)\neq r_t)\\ p(r_t|S_{pre};\theta) & (T)\wedge(U=0)\wedge(\arg\max_r p(r|S_{pre};\theta)=rt)\\ p(r_t|S_{pre};\theta)-\max(p(r|S_{pre};\theta)) & (T)\wedge(U=0)\wedge(\arg\max_r p(r|S_{pre};\theta)\neq r_t)\end{cases}$$

don't explain every case here. Actually, they are all designed to encourage the beneficial action while discouraging inadvisable one.

Additionally, considering the excessive number of sentences in some bags, we adopt the penalty factor γ to penalize the search action (i.e., $P=0$) so that the model can get rid of exorbitant search and improve efficiency.

3.4 Reinforcement Learning for MIML problem

For a bag $B=\{\langle e_1,e_2\rangle,(r_1,\cdots,r_l),\{S_1,\cdots,S_n\}\}$, we start l episodes, denoted as $Ep_1,Ep_2,\cdots,Ep_l$, with the initial state St_i^0, where $i\in\{1,2,\cdots,l\}$, and then $St_i^0=[e_1;e_2;\ \mathbf{0};\ \mathbf{0};\ \boldsymbol{S_1};\ p(r|S_1,\theta)]$ in which the first **0** means no previous sentence features while the second **0** indicates no confidence scores.

As aforementioned (i.e., in Section 3.2), the agent determines the optimal action according to $Q(St,A)$, including $Q(St,U)$ and $Q(St,P)$, and takes the action A^* that equals $argmax_A Q(St,A)$.

Here, our model meets two **challenges**:

- According to express-at-most-one, one sentence can at most express one single relational fact between entities of interest. When two or more episodes simultaneously intend to update the previous sentence with the current one (i.e., $Q(St,U=1)>Q(St,U=0)$), how to decide that which episode should be permitted the update action?
- The initial states St_i^0 for all episodes of a bag are exactly the same (including entity information, sentence information and confidence scores). Obviously, given the same states (i.e., inputs), the agent in all episodes will generate the same output values of Q-function. How to initially separate these episodes and equip them with the ability to be aware of the relation represented by them, respectively (like a cold-start problem)?

To solve the first challenge, we treat the assignment of each sentence from a bag as an auction. The auctioneer (i.e., the agent in reinforcement learning) permits the update action of whose bidder produces the highest $Q(St,U=1)$ among these episodes, while the update action of the rest episodes is forbidden. More formally, $U_k^j\leftarrow 1$, where the subscript $k\leftarrow argmax_i Q(St_i^j,U=1)$ is the k-th episode and the superscript j means the j-th step of the episode, and $U_t^j\leftarrow 0$ in which $t\in[1,k-1]\cup[k+1,l]$.

To resolve the second challenge, we design a **heuristic initialization**. Specifically, we fetch out l sentences from the bag. For the first sentence, we calculate the confidence scores $p(r|S_1;\theta)$ for each relation in the candidate relations of the bag (i.e., $\{r_1,\cdots,r_l\}$). Then choose the relation with the maximum value $r^*\leftarrow argmax_r p(r|S_1;\theta)$ and replace its anchored episode's initial state with $St_i^0=[e_1;e_2;\ \boldsymbol{S_1};\ p(r|S_1,\theta);\ \boldsymbol{S_{l+1}};\ p(r|S_{l+1},\theta)]$. After that, process the second sentence according to the rest relation of the bag (i.e., $\{r_1,\cdots,r_l\}-\{r^*\}$). Repeat the above procedures until all the sentences (i.e., $\{S_1,\cdots,S_l\}$) are processed. The initial states for different episodes are then clearly distinct.

According to the above analysis and preprocessing, we define the procedure of reinforcement learning for MIML in *Algorithm* 1.

In the algorithm, line 6-12 is intended to discover the episode that gives the maximum value of $Q(St,U=1)$ and updates the episode's chosen sentence. Line 13-16 is to check whether the bag has unprocessed sentences. Line 17-29 describes the the transition from the current states (e.g., St_i^{j-l-1}) to the new states (e.g., St_i^{j-l}) of all the running episodes in Ep according to the actions generated by the agent (involving some changes to the actions selected by the agent, for only one update action can be permitted). Additionally, put the transition into the replay memory Rm. Line 30-31 checks whether there still exist running episodes. The output of the algorithm consists of two components: $TrainR$ and Rm, in which $TrainR$ is the collection of chosen sentences and used to feed the training of the extractor while Rm is the replay memory (store transitions) for the training of value-based reinforcement learning.

It worth noting that the algorithm also utilizes ε-greedy exploration to get rid of the local optimal solution. In particularly, we generate a random value $\rho\in[0,1]$. If $\rho<\varepsilon$, the action $A=\{U,P\}$ is randomly determined. Otherwise, we select the action A by $argmax_A Q(St_i^{j-l-1},A)$ (i.e., in line 18 of *Algorithm* 1). As ε decreases, the probability of the exploration will be reduced and the model will gradually converge to the optimal solution.

3.5 Joint training process of the extractor and reinforcement learning

The joint training phases of the extractor and reinforcement learning are shown in *Algorithm* 2. We first select expressive sentences for each relation of a bag, and then feed these selected sentences in TS to train the extractor while using the transitions in D to train the agent. Next, we elaborate on the optimization of these two models (i.e., line 10 and line 13).

3.5.1 Optimization of the extractor. We define the objective function for the extractor using cross-entropy at sentence-level as follows:

$$J(\theta)=\sum_{i=1}^{|S|}\log p(y_i|x_i;\theta)$$

where x_i is a sentence in TS with relation y_i and $|S|$ is the total number of all the chosen training sentences from the original training data. To implement optimization, we employ stochastic gradient descent (SGD) over the batch size bs_1 with the learning rate λ_1.

Algorithm 1 Reinforcement learning for MIML

Input:
(1) a training bag $B = \{\langle e_1, e_2\rangle, (r_1, \cdots, r_l), \{S_1, \cdots, S_n\}\}$;
(2) the current agent Ge;
(3) the current extractor Ex

initialize episodes $Ep = \{Ep_1, \cdots, Ep_l\}$ and $TrainR_i$, where Ep_i for r_i with the initial state St_i^0 ($i \in [1, l]$), and $TrainR_i$ records the sentence that will be finally chosen to serve as the training instance for the relation r_i
replay memory $Rm \leftarrow \phi$
for $j = l+1, \cdots, n$ **do**
  $Q \leftarrow \phi$
  $i^* \leftarrow 0$
  for each $Ep_i \in Ep$ **do**
    compute $Q(St_i^{j-l-1}, U)$ by Ge
    if $argmax_U Q(St_i^{j-l-1}, U) = 1$ **then**
      put $Q(St_i^{j-l-1}, U = 1)$ into Q
  if $Q \neq \phi$ **then**
    get $i^* \leftarrow argmax_i Q(St_i^{j-l-1}, U = 1)$ from Q
    $TrainR_{i^*} \leftarrow S_j$
  if $j + 1 <= n$ **then**
    get S_{j+1} and $p(r|S_{j+1}; \theta)$ by Ex
  else
    break
  for each $Ep_i \in Ep$ **do**
    $A \leftarrow argmax_A Q(St_i^{j-l-1}, A)$
    if $argmax_P Q(St_i^{j-l-1}, P) = 0$ **then**
      if $i = i^*$ **then**
        $St_i^{j-l}(\boldsymbol{S_{pre}^{j-l}} \leftarrow \boldsymbol{S_{cur}^{j-l-1}}; p(r|S_{pre}^{j-l}; \theta) \leftarrow p(r|S_{cur}^{j-l-1}; \theta))$
      else
        $St_i^{j-l}(\boldsymbol{S_{pre}^{j-l}} \leftarrow \boldsymbol{S_{pre}^{j-l-1}}; p(r|S_{pre}^{j-l}; \theta) \leftarrow p(r|S_{pre}^{j-l-1}; \theta))$
        $U \leftarrow 0$
      $Reward = Re(St_i^{j-l-1}, A)$
      St_i^{j-l} $(\boldsymbol{S_{cur}^{j-l}} \leftarrow \boldsymbol{S_{j-l}}; p(r|S_{cur}^{j-l}; \theta) \leftarrow p(r|S_{j-l}; \theta))$
      put $(St_i^{j-l-1},\ A,\ Reward,\ St_i^{j-l})$ into Rm
    else
      delete Ep_i from Ep
  if $Ep = \phi$ **then**
    break
return $TrainR$, Rm

3.5.2 Optimization of reinforcement learning. Follow [16], we adopt Deep Q-learning to optimize $Q(St, A)$. Given a transition $(St^i, A, Reward, St^{i+1})$, we can obtain the value z_i using Bellman equation:

$$z_i = \begin{cases} \text{Re}ward & St^{i+1} \in terminal \\ \text{Re}ward + \alpha \max_{A'} Q(St^{i+1}, A') & St^{i+1} \notin terminal \end{cases}$$

where z_i is regarded as the accurate value of $Q(St^i, A)$ and α is the decay factor. Then the loss function of reinforcement learning is denoted by the sum-squared error:

$$Loss = \sum_{i=1}^{|Z|} (z_i - Q(St_i, A_i; \eta))^2$$

Algorithm 2 Joint training for the extractor and reinforcement learning

Input:
(1) A set of training bags $B = \{B_1, B_2, \cdots, B_b\}$
(2) Untrained extractor and untrained Q-function

initialize the parameter θ for the extractor and the parameter η for Q-function
for $t = 1, \cdots, T$ **do**
  replay memory $D \leftarrow \phi$
  while $B \neq \phi$ **do**
    sample random batch Bs from B with batch size bs_1
    chosen training sentences $TS \leftarrow \phi$
    for each $B_i \in Bs$ **do**
      $TrainR$, $Rm \leftarrow$ execute *Algorithm 1*
      put $TrainR$ into TS, and Rm into D
    update θ using TS
    if size(D)>Ω **then**
      sample random batch Br from D with batch size bs_2
      update η using Br
return θ, η

where $|Z|$ is the total number of all the transitions used for training, and we utilize SGD over the batch size bs_2 with the learning rate λ_2.

4 EXPERIMENTS

In this section, we empirically conduct comparative experiments to demonstrate the effectiveness of the proposed method. To this end, we first introduce the dataset and the evaluation metrics used in the experiments. Then we describe some details about the model implementation. Finally, we present the experimental results along with some discussions.

4.1 Dataset and evaluation metrics

We evaluate our model on a widely used dataset [4], which is developed by [21] and has also been used by [4, 13, 21, 30, 32]. The data is generated by aligning Freebase with New York Times corpus (NYT). To find entities mentioned in texts, they use Stanford named entity recognizer [5] and treat consecutive mentions which share the same category as a single entity mention. The association between Freebase and NYT is built by performing a string match between entity mention phrases and canonical names of entities in Freebase. The relations in Freebase are divided into two parts, one for training and the other for testing. Then the former is aligned to NYT in the year 2005-2006 and the later to NYT in the year 2007.

Following [4, 13, 30, 32], we adopt held-out evaluation. In testing phase, the precision and recall are calculated by comparing the predictions with the relational facts in Freebase. To sufficiently demonstrate the performance of each model, we evaluate them using various aspects of metrics, including precision/recall curves, the highest F1 value and P@N metrics.

[4] Available at http://iesl.cs.umass.edu/riedel/ecml/
[5] Available at http://nlp.stanford.edu/software/CRF-NER.shtml

4.2 Implementation details

In this paper, we train word embeddings on NYT corpus by using word2vec[6] tool in advance, and we concatenate consecutive words to represent an entity when the entity has multiple words.

For our model, we tune all the parameters using three-fold validation. In detail, we utilize a grid search to determine the optimal parameters. The parameter settings are listed as follows:

parameters in the extractor:

- dimension of word embeddings: $dw = 50$
- dimension of position embeddings: $dp = 5$
- size of convolutional window: $l = 3$
- number of convolutional filter: $nc = 230$
- size of mini batch: $bs_1 = 2000$
- probability of dropout: $p = 0.5$
- learning rate: $\lambda_1 = 0.01$

parameters in reinforcement learning:

- size of mini batch: $bs_2 = 1000$
- size of replay memory: $\Omega = 2,5000$
- decay factor: $\alpha = 1.0$
- learning rate: $\lambda_2 = 0.01$
- penalty factor: $\gamma = -0.02$
- ε-greedy exploration is from 1.0 to 0.1, including 10,000 transitions
- the hidden units in $\boldsymbol{W}$ are 50

Additionally, the training iterations of the entire model: T=10

4.3 Baselines

To evaluate the effectiveness of our model, we use the following state-of-the-art baselines:

feature-based methods

- **Mintz** is a traditional feature-based method and is the first to develop distant supervision for relation extraction[14].
- **Hoffmann** is a probabilistic graphic model that intends to resolve multi-instance problem with overlapping relations in distant supervision[7].
- **Surdeanu** utilizes latent variables to alleviate MIML problem and employs Expectation-Maximization(EM) algorithm to optimize the model[25].

neural network methods

- **CNN** applies CNN to automatically encode sentences for relation extraction, without considering noise problem[31].
- **CNN+ONE** follows expressed-at-least-once assumption and simply selects the sentence with the maximum confidence in each bag to serve as a training instance[30].
- **CNN+ATT** resolves multi-instance problem using attention mechanism and achieves the current state-of-the-art performance[13].
- **Feng** attempts to use policy-based reinforcement learning to alleviate multi-instance problem in noisy data[4].

To implement above baselines, we use their public released codes and follow the parameter settings reported in their papers[7].

[6]Available https://code.google.com/p/word2vec/

[7]The codes for *Mintz*, *Hoffmann* and *Surdeanu* are available at http://nlp.stanford.edu/software/mimlre.shtml; The codes for *CNN*, *CNN+ONE* and *CNN+ATT* are available at https://github.com/thunlp/NRE; The code for *Feng* is available at https://github.com/JuneFeng/RelationClassification-RL

Table 2: P@N for relation extraction

P@N(%)	100	200	300	500	mean
CNN	59.00	53.00	47.67	46.20	51.47
CNN+ONE	69.00	61.00	61.33	56.80	62.03
CNN+ATT	76.00	70.50	66.67	59.40	68.14
Feng	57.00	59.50	60.67	56.00	58.29
CNN+RL	**82.00**	**76.50**	**72.00**	**64.40**	**73.73**

Our model is named as: **CNN+RL** and develops value-based **r**einforcement **l**earning to deal with MIML problem.

4.4 Result and discussion

4.4.1 Held-out evaluation. **PR curves with neural networks methods**

Figure 2(a) shows the PR curves of our model together with other neural models. All of these models adopt CNN to encode sentences, the difference between them lies in the solutions to noisy data. From Fig.2(a), we have the following observations: (1) *CNN+RL* consistently and substantially outperforms all the baselines, including *CNN+ATT* (the current state-of-the art method). The result indicates that our model can effectively solve MIML problem and promote the performance of the extractor. (2) *CNN* often gets the lowest precision with the same point of recall among these methods, which demonstrates that noise problem is indeed grievous in distant supervision. (3) *CNN+RL* achieves higher precision and recall than *Feng*. We believe that it verifies the effectiveness of fine-grained reward function and considering multi-label problem in noisy data. Specifically, *Feng* designs reward function at bag-level and the agent cannot obtain the reward values until it completes selection of sentences for each bag. In contrast, our model develops sentence-level reward function, resulting in a well-trained Q-function that acts as a guider to supervise every action $A(U, P)$. In this way, our model is able to better capture the most expressive sentence for each relation of a bag, and get rid of the influence of noisy data. (4) *CNN+ONE* and *CNN+ATT* only alleviate multi-instance problem in distant supervision regardless of multi-label problem, degrading the performance of their models.

PR curves with feature-based methods

We depict the comparison of *CNN+RL* with feature-based methods in Figure 2(b). In this figure, as recall increases, *CNN+RL* performs much better than all these methods, especially, when the recall is larger than 0.15, the improvement of precision is over 20%. We believe that feature-based methods (*Mintz, Hoffmann, Surdeanu*) utilize features derived from NLP tools, which inevitably suffer from error propagation and accumulation. Unlike them, *CNN+RL* applies CNN to implement automatic feature engineering and automatically learn a better semantic representation of sentences.

P@N scores Now we rank the predictions in decreasing order of confidence scores. Then fetch out the top N predictions and check their accuracy. The results are shown in Table 2.

In Table 2, we can see that: (1) In P@100, P@200, P@300 and P@500, as expected, *CNN+RL* always achieves higher precision than

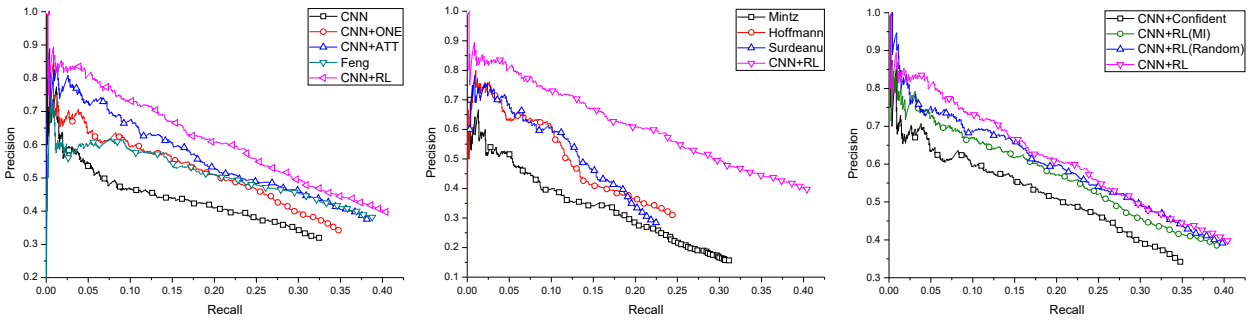

Figure 2: Effectiveness of the proposed model

all the previous models. Eventually, *CNN+RL* gets the highest average precision score among them (5.59% higher than *CNN+ATT* and 15.44% higher than *Feng*). We conclude that CNN+RL can choose informative sentences among mixed data (well-labeled and wrongly-labeled data are mixed together) to serve as training instances that are beneficial to generating precise predictions in testing phase. (2) *Feng* performs merely better than *CNN* and worse than the rest of methods, especially *CNN+RL*. The result illustrates that coarse reward function is unable to guide the agent to make ideal actions and the ignorance of multi-label problem will do harm to the extractor.

4.4.2 Reasonability of the model design. To comprehensively evaluate the reasonability of our model, we develop several variants of *CNN+RL*:

- **CNN+RL(MI)** only treats distant supervision as multi-instance learning. For a training bag with multiple labels (i.e., $B = \{\langle e_1, e_2\rangle, (r_1, \cdots, r_l), \{S_1, \cdots, S_n\}\}$), we simply generate l bags, where l is the number of relations linking the entity pair of interest. Then all the original candidate sentences will be put into these new bags (i.e., $B_1 = \{\langle e_1, e_2\rangle, (r_1), \{S_1, S_2, \cdots, S_n\}\}, \cdots, B_l = \{\langle e_1, e_2\rangle, (r_l), \{S_1, \cdots, S_n\}\}$). Given one of these bags B_i, the model assigns an episode for it and conducts reinforcement learning to choose the most suitable sentence describing r_i.
- **CNN+RL(Random)** solves the second challenge (mentioned in Section 3.4) through randomly initializing the initial states of l episodes with the first l sentences, instead of the heuristic initialization.
- **CNN+Confident** is similar to *CNN+ONE*, straightforwardly select the sentence with the maximum confidence value given by the current extractor for each bag. Moreover, the model neglects multi-label problem.

Figure 2(c) shows the overall performance of these methods using PR curves. According to Fig.2(c), we conclude that: (1) *CNN+RL* outperforms *CNN+RL (random)* in most range of the curves. When the recall rate is low (from 0.025 to 0.15), the gap between them is more pronounced. The main reason for this phenomenon is that with random initialization, the episodes cannot grasp the relations that they represent at the beginning. Then they gradually understand their relations through the continuous competition with other episodes and the feedback from reward function. If the initial states are given incorrect sentences, the process will be tougher. Consequently, random initialization confuses *CNN+RL(Random)*, and the model cannot give particularly high confidence even when faced with explicitly-expressed sentences. In contrast, *CNN+RL* starts each episode with heuristic information that facilitates the discrimination of multiple relations for one single bag. (2) Compared with *CNN+RL(MI)*, *CNN+RL* always achieves higher precision with the same recall. We believe that it is necessary to pay attention to multi-label problem in distant supervision. Because distant supervision utilizes a large-scale KB to generate training data and the case that two entities participate more than one relation is very common in reality. Under this circumstance, *CNN+RL* exploits auction mechanism to reach self-adaption, and significantly solves noise problem. (3) The performance of *CNN+Confident* is far less than *CNN+RL*. In fact, these two methods both adopt expressed-at-least-once assumption, so the number of chosen sentences used for training extractors is the same. *CNN+Confident* selects sentences with the highest confidence while our model develops reinforcement learning to explore accurate sentences. We argue that *CNN+Confident* is a naive strategy and is unable to reach the complexity of the problem. On the contrary, to discover informative sentences, our model builds reinforcement learning and takes advantage of various information, including entity information, sentence information and confidence scores. In this way, *CNN+RL* excellently accomplishes the task of screening sentences.

4.4.3 Case study. To further check the effectiveness of the proposed model, we give a deeper inspection on training instances. Table 3 shows two examples of the entity pairs with multiple relations. In the first example, *Feng* selects both of the sentences (*S1*, *S2*) as the training sentences for relation "/location/contain" and "/country/capital". However, the actual relation for *S1* is "/country/capital" while *S2* expresses "/location/contain". Unfortunately, none relation information embedded in the state and ignorance of multi-label problem results in wrongly-labeled problem in their model. In contrast, with the help of heuristic initialization and auction mechanism, our model identifies the actual relation for each of them and makes the ideal selection. In the second example, *Feng* doesn't find that *S4* explicitly describes the relational fact: *Jane Jacobs* died in *Toronto*. The reason for this, we believe, is that their model doesn't utilize confidence scores as evidence for selection

Table 3: Choose sentences from the dataset for training the extractor

Method	Sentences	/location/contains	/country/capital
Feng	*S1:* according to local legend , recounted by the Africa scholar Stephen Ellis in his book " the mask of anarchy , " a baby born in **Monrovia** , **Liberia** 's capital , miraculously spoke english straight from the womb	choose	choose
CNN+RL		not choose	choose
Feng	*S2:* my sister was living in the **Monrovia** suburb of Paynesville , **Liberia** , with her family and a handful of orphans and other refugees from the liberian civil war.	choose	choose
CNN+RL		choose	not choose
		/person/place_lived	/deceased_person /place_of_death
Feng	*S3:* **Jane Jacobs** , the activist who took him on , now lives in **Toronto** .	choose	choose
CNN+RL		choose	not choose
Feng	*S4:* **Jane Jacobs** , the writer and thinker who brought penetrating eyes and ingenious insight to the sidewalk ballet of her own greenwich village street and came up with a book that challenged and changed the way people view cities , died yesterday in **Toronto** , where she moved in 1968 .	not choose	not choose
CNN+RL		not choose	choose

and sentence information may be noneffective when sentences are relatively long (the sentence may contain various information, and we randomly examine 600 sentences in original training data and over 70% has more than 30 words.). Additionally, the coarse reward function is not of enough capacity to lead the agent to determine precise sentence-level action. Compared with them, our model embeds confidence scores in state and develops a fine-grained reward function, making up for these deficiencies. Therefore, as expected, *CNN+RL* chooses *S3* to serve as the training instance for "/person/place_lived" and *S4* for "/deceased_person/place_of_death".

5 RELATED WORKS

Supervised relation extraction

Relation extraction is one of the most important research tasks in NLP. Many efforts based on supervised learning have been invested to boost the performance of relation extractors. [33] employs kernel methods for relation extraction. Other classifiers, such as maximum entropy model [10] and conditional random fields [3], have also demonstrated the ability to achieve outstanding performance on domain-specific data. Recently, neural networks have been successfully applied to many NLP tasks [19]. To avoid hand-designed features, researchers have investigated the possibility of using neural networks to automatically learn features for relation extraction: recursive neural networks (RNNs) [24], convolutional neural network (CNN) [31] and long short-term memory (LSTM) [15]. However, supervised methods entirely rely on manually annotated data, and cannot meet the demand of big data era.

Distantly supervised relation extraction

Distant supervision for relation extraction, firstly introduced by [14], automatically generates training data through heuristic alignment between a knowledge base and plain texts. Although distant supervision is an efficient way to scale relation extraction to a large number of relations, the basic assumption used in the alignment is so strong that will inevitably bring wrong labelling problem. To alleviate noise, [7, 21] build multi-instance learning paradigms. Specifically, [21] applies constraint-driven semi-supervision to train their model. [7] builds a probabilistic graphic model and intends to resolve multi-instance with overlapping relations in distant supervision. [25] further extends multi-instance to MIML and trains a Bayesian framework by expectation maximization (EM) algorithm.

Later, considering automatic feature engineering, [13, 30] integrate multi-instance learning model with PCNNs to extract relations on distantly supervised data. Between them, [30] simply selects the sentence with the highest confidence scores in each bag, while [13] establishes sentence-level attention to select valid sentences, achieving the current state-of-the-art performance.

Reinforcement learning in NLP

Recently, reinforcement learning has exhibited prominent achievements in many NLP tasks, including semantic parsing[11], abstractive summarization[22], sentiment analysis[20] and question answering[29]. [17] employs reinforcement learning to retrieve incidents with reference to external evidence. In relation extraction, [4] tackles noise problem with the help of policy-based reinforcement learning . Although these two works are more related to our model, the substantial difference between our model and theirs can be listed as follows: (1) they design reward function at bag-level, and the agent in their model can only get reward after finishing the selection of all the sentence in a bag (zero-value reward within an episode). In contrast, our model offers a fine-grained (sentence-level) reward function, refining every action's reward and resulting in a well-trained agent. (2) they arbitrarily regard noise problem as multi-instance learning and ignore that multi-label problem also exists in distant supervision. However, as [7, 25] mentioned, multi-label problem is very common and severe in distant supervision due to the usage of a large-scale knowledge base. Hence, we simultaneously consider multi-instance with multi-label and develop an auction mechanism. Different relations for a bag have to compete with each other to acquire the most expressive sentence for each of them. (3) the state in their models only contains sentence information and entity information while our model employs entity

information, sentence information and confidence scores to constitute the state so that the agent makes decision with reference to abundant information. We argue that with various information, the model is able to be more stable and reliable, avoiding unilateral information that leads to miscarriage of judgement. In particular, we show that the confidence information takes crucial effect when sentences are relatively long (i.e., express various information). (4) the search action P can early terminate the episode when the agent has adequate confidence with the chosen sentence, improving the efficiency of the proposed model. However, their models have to process all the sentences in each training bag.

6 CONCLUSION

In this paper, we extend expressed-at-least-once to multi-label level and develop express-at-most-one assumption. On the basis of these assumptions, we craft reinforcement learning to solve MIML problem, and generate the sentence-level annotated signal in distantly supervised relation extraction. Then these chosen expressive sentences serve as training instances to feed the extractor. We conduct extensive experiments and the experimental results demonstrate that our model can effectively alleviate MIML problem and achieve the new state-of-the-art performance.

7 ACKNOWLEDGEMENTS

This research is partially supported by Natural Science Foundation of China (Grant No. 61772356, 61602204) and the Australian Research Council (Grants No. DP170101172)

REFERENCES

[1] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Aligh and Translate. In *International Conference on Learning Representations*. 1–15.
[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Proceedings of Neural Information Processing Systems*. 2787–2795.
[3] Aron Culotta, Andrew McCallum, and Jonathan Betz. 2006. Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text. In *Proceedings of Association for Computational Linguistics*. 296–303.
[4] Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement Learning for Relation Classification from Noisy Data. In *Proceedings of AAAI Conference on Artificial Intelligence*.
[5] Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. Neural Knowledge Acquisition via Mutual Attention between Knowledge Graph and Text. In *Proceedings of AAAI Conference on Artificial Intelligence*.
[6] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O, Seaghdha, Sebastian Pado, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. 33–38.
[7] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of Association for Computational Linguistics*. 541–550.
[8] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2016. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In *Proceedings of AAAI Conference on Artificial Intelligence*. 985–991.
[9] Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions. In *Proceedings of AAAI Conference on Artificial Intelligence*. 3060–3066.
[10] Nanda Kambhatla. 2004. Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations. In *Proceedings of Association for Computational Linguistics*. 1003–1011.
[11] Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision. In *Proceedings of 55th Annual Meeting of the Association for Computational Linguistics*. 23–33.
[12] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of AAAI Conference on Artificial Intelligence*. 2181–2187.
[13] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huangbo Luan, and Maosong Sun. 2016. Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of Association for Computational Linguistics*. 2124–2133.
[14] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*. 1003–1011.
[15] Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of Association for Computational Linguistics*. 1105–1116.
[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Natural* 518, 7450 (2015), 529–533.
[17] Karthik Narasimhan, Adam Yala, and Regina Barzilay. 2016. Improving Information Extraction by Acquiring External Evidence with Reinforcement Learning. In *Proceedings of Empirical Methods in Natural Language Processing*. 2355–2365.
[18] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic Embeddings of Knowledge Graphs. In *Proceedings of AAAI Conference on Artificial Intelligence*. 1955–1961.
[19] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Mazumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-Dependent Sentiment Analysis in User-Generated Videos. In *Proceedings of Association for Computational Linguistics*. 873–883.
[20] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to Generate Reviews and Discovering Sentiment. In *arXiv:1704.01444v2*.
[21] Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling Relations and Their Mentions without Labeled Text. In *Proceedings of North American Chapter of the Association for Computational Linguistics*. 148–163.
[22] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. ReasoNet: Learning to Stop Reading in Machine Comprehension. In *Proceedings of Knowledge Discovery and Data Mining*. 1047–1055.
[23] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 1995. Mastering the game of go with deep neural networks and tree search. *Natural* 31, 15 (1995), 1244–1245.
[24] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces. In *Proceedings of Empirical Methods in Natural Language Processing*. 1201–1211.
[25] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance Multi-label Learning for Relation Extraction. In *Proceedings of Empirical Methods in Natural Language Processing*. 1003–1011.
[26] Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. The MIT Press, Reading, MA.
[27] Wen tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In *Proceedings of Association for Computational Linguistics*. 1321–1331.
[28] Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge Base Completion via Search-Based Question Answering. In *Proceedings of World Wide Web*. 515–526.
[29] Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *International Conference on Learning Representations*. 1047–1055.
[30] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks. In *Proceedings of Empirical Methods in Natural Language Processing*. 1753–1762.
[31] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING*. 2335–2344.
[32] Xiangrong Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Large Scaled Relation Extraction with Reinforcement Learning. In *Proceedings of AAAI Conference on Artificial Intelligence*.
[33] Shubin Zhao and Ralph Grishman. 2005. Extracting Relations with Integrated Information Using Kernel Methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. 419–426.