SEE: Syntax-aware Entity Embedding for Neural Relation Extraction

Zhengqiu He[1], Wenliang Chen[1,4]*, Zhenghua Li[1]
Meishan Zhang[3], Wei Zhang[2], Min Zhang[1]
[1]School of Computer Science and Technology, Soochow University, China
[2]Alibaba Group, China
[3]School of Computer Science and Technology, Heilongjiang University, China
[4]Collaborative Innovation Center of Novel Software Technology and Industrialization, China
zqhe@stu.suda.edu.cn, {wlchen, zhli13, minzhang}@suda.edu.cn
mason.zms@gmail.com, lantu.zw@alibaba-inc.com

Abstract

Distant supervised relation extraction is an efficient approach to scale relation extraction to very large corpora, and has been widely used to find novel relational facts from plain text. Recent studies on neural relation extraction have shown great progress on this task via modeling the sentences in low-dimensional spaces, but seldom considered syntax information to model the entities. In this paper, we propose to learn syntax-aware entity embedding for neural relation extraction. First, we encode the context of entities on a dependency tree as sentence-level entity embedding based on tree-GRU. Then, we utilize both intra-sentence and inter-sentence attentions to obtain sentence set-level entity embedding over all sentences containing the focus entity pair. Finally, we combine both sentence embedding and entity embedding for relation classification. We conduct experiments on a widely used real-world dataset and the experimental results show that our model can make full use of all informative instances and achieve state-of-the-art performance of relation extraction.

Introduction

Relation extraction (RE), defined as the task of extracting semantic relations between entity pairs from plain text, has received increasing interests in the community of natural language processing (Riedel et al. 2013; Miwa and Bansal 2016). The task is a typical classification problem after the entity pairs are specified (Zeng et al. 2014). Traditional supervised methods require large-scale manually-constructed corpus, which is expensive and confined to certain domains. Recently, distant supervision has gained a lot of attentions which is capable of exploiting automatically-produced training corpus (Mintz et al. 2009). The framework has achieved great success and has brought state-of-the-art performances in RE.

Given an entity pair (e', e'') from one knowledge base (KB) such as Freebase, assuming that the predefined semantic relation on the KB is r, we simply label all sentences containing the two entities by label r. This is the key principle for distant supervision to produce training corpus. While this may be problematic in some conditions, thus can result in noises. For example, the sentence "*Investors include Vinod Khosla of Khosla Ventures, who, with the private equity group of texas pacific group ventures, invested $20 million.*" is not for relation */business/company/founders* of *Khosla Ventures* and *Vinod Khosla* in Freebase, but it is still be regarded as a positive instance under the assumption of distant supervision. Based on the observation, recent work present multi-instance learning (MIL) to address the problem, by treating each produced sentence differently during the training (Riedel, Yao, and McCallum 2010; Zeng et al. 2015; Lin et al. 2016). Our work also falls into this category.

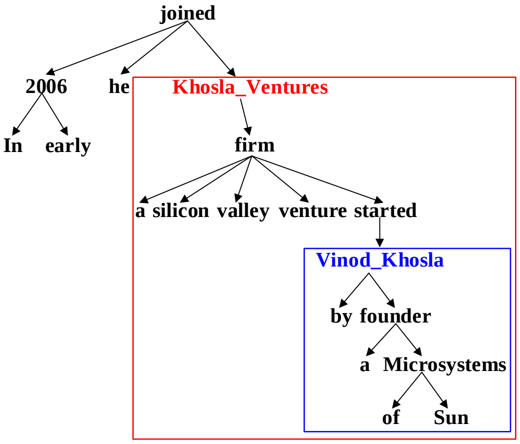

Figure 1: An example dependency tree containing two entities in sentence "In early 2006, he joined Khosla Ventures, a silicon valley venture firm started by Vinod Khosla, a founder of Sun Microsystems.".

Under the statistical models with handcrafted features, a number of studies have proposed syntactic features, and achieved better results by using them (Hoffmann et al. 2011; Surdeanu et al. 2012). Recently, the neural network models have dominated the work of RE because of higher performances (Lin et al. 2016; Ji et al. 2017). Similarly, the syntax information has also been investigated in neural RE. One representative method is to use the shortest dependency path (SDP) between a given entity pair (Miwa and Bansal 2016),

[*] The corresponding author is Wenliang Chen.

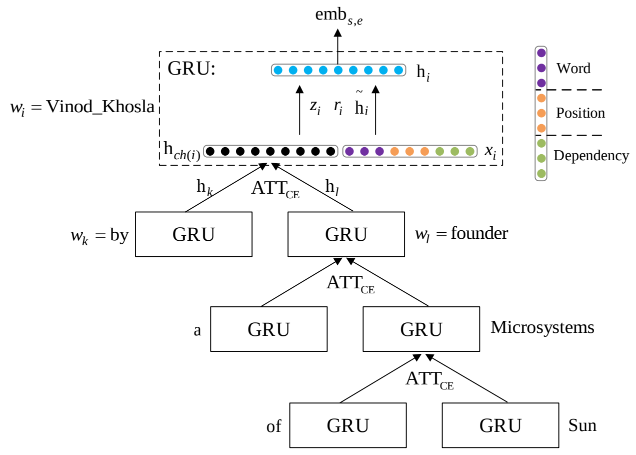

Figure 2: Workflow of entity embedding via tree-GRU.

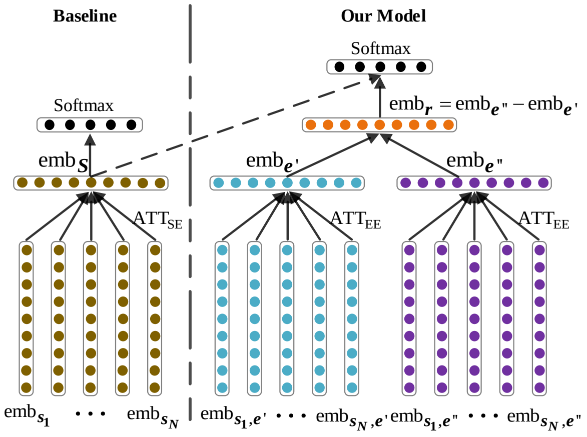
Baseline
Our Model
Softmax
Softmax
$\mathrm{emb}_{\boldsymbol{r}} = \mathrm{emb}_{\boldsymbol{e}''} - \mathrm{emb}_{\boldsymbol{e}'}$
$\mathrm{emb}_{\boldsymbol{S}}$
$\mathrm{emb}_{\boldsymbol{e}'}$
$\mathrm{emb}_{\boldsymbol{e}''}$
$\mathrm{ATT_{SE}}$
$\mathrm{ATT_{EE}}$
$\mathrm{ATT_{EE}}$
$\mathrm{emb}_{\boldsymbol{s_1}}$ ··· $\mathrm{emb}_{\boldsymbol{s_N}}$
$\mathrm{emb}_{\boldsymbol{s_1},\boldsymbol{e}'}$ ··· $\mathrm{emb}_{\boldsymbol{s_N},\boldsymbol{e}'}$
$\mathrm{emb}_{\boldsymbol{s_1},\boldsymbol{e}''}$ ··· $\mathrm{emb}_{\boldsymbol{s_N},\boldsymbol{e}''}$

Figure 3: Workflow of the baseline and our approach.

based on which long short term memory (LSTM) can be applied naturally to model it. This method has brought remarkable results, since the path words are indeed good indictors for semantic relation and meanwhile SDPs can remove abundant words between entity pairs.

The above work of using syntax concerns mainly on the connections between entity pairs, paying much attention on the words that link the two entities semantically, while neglects the representation of entities themselves. Previous entity embeddings purely based on their sequential words can be insufficient to generalize to unknown entities. But it can be different when we try to capture the meaning of entities by its syntactic contexts. For example, as shown in Figure 1, when use the subtrees rooted at *Khosla Ventures* and *Vinod Khosla* to represent the two entities, we could capture longer distance information than only use the entities themselves. It indicates that the syntax roles the entities played in the sentences are informative for RE.

In this paper, we propose syntax-aware entity embedding (SEE) for enhancing neural relation extraction. As illustrated in Figure 2, to enrich the representation of each entity, we build tree-structured recursive neural networks with gated recursive units (tree-GRU) to embed the semantics of entity contexts on dependency trees. Moreover, we employ both intra-sentence and inter-sentence attentions to make full use of syntactic contexts in all sentences: (1) attention over child embeddings in a parse tree to distinguish informative children; (2) attention over sentence-level entity embeddings to alleviate the wrong label problem. Finally, we combine all sentence embeddings and entity embeddings for relation classification. We evaluate our model on the widely used benchmark dataset and show that our proposed model achieves consistently better performance than the state-of-the-art methods.

The Baseline

Our baseline model directly adopts the state-of-the-art neural relation extraction model proposed by Lin et al. (2016), which also employs multi-instance learning for alleviating the wrong label problem faced by the distant supervision paradigm.

The framework of the baseline approach is illustrated in the left part of Figure 3. Suppose there are N sentences $S = \{s_1, ..., s_N\}$ that contain the focus entity pair e' and e''. The input is the embeddings of all the sentences. The i-th sentence embedding, i.e., $\mathbf{emb}_{s_i}$, is built from the word sequence, and encodes the semantic representation of the corresponding sentence. Then, an attention layer is performed to obtain the representation vector of the sentence set. Finally, a softmax layer produces the probabilities of all relation types.

Sentence Embedding

Figure 4 describes the component for building a sentence embedding from the word sequence. Given a sentence $s = \{w_1, ..., w_n\}$, where w_i is the i-th word in the sentence, the input is a matrix composed of n vectors $\mathbf{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$, where $\boldsymbol{x}_i$ corresponds to w_i and consists of the word embedding and its position embedding. Following Zeng et al. (2015) and Lin et al. (2016), we employ the skip-gram method of Mikolov et al. (2013b) to pretrain the word embeddings, which will be fine-tuned afterwards. Position embeddings are first successfully applied to relation extraction by Zeng et al. (2014). Given a word (e.g., “firm” in Figure 1), its position embedding corresponds to the relative distance (“6&-3”) from the word to the entity pairs (“Khosla Ventures” and “Vinod Khosla”) through lookup.

A convolution layer is then applied to reconstruct the original input $\mathbf{X}$ by learning sentence features from a small window of words at a time while preserving word order information. They use K convolution filters (a.k.a. feature maps) with the same window size l. The j-th filter uses a weight matrix $\mathbf{W}_j^f$ to map $\boldsymbol{X}$ into a j-th-view vector $\mathbf{Conv}_j(\boldsymbol{X})$, which contains $n - l + 1$ scalar elements. The i-th element is computed as follows:

$$\mathbf{Conv}_j(\boldsymbol{X})[i] = \mathbf{W}_j^f\ \boldsymbol{X}_{i:i+l-1}$$

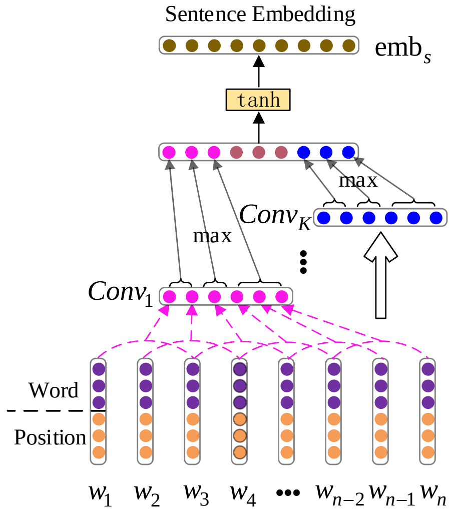

Figure 4: Workflow of sentence embedding.

Three-segment max-pooling is then applied to map K convolution output vectors of varying length into a vector of a fixed length $3K$. Suppose the positions of the two entities are p_1 and p_2 respectively.[1] Then, each convolution output vector $\mathbf{Conv}_j(\boldsymbol{X})$ is divided into three segments:

$$[0 : p_1 - 1]/[p_1 : p_2]/[p_2 + 1 : n - l]$$

The max scalars in each segment is preserved to form a 3-element vector, and all vectors produced by the K filters are concatenated into a $3K$-element vector, which is the output of the pooling layer.[2]

Finally, the sentence embedding $\mathbf{emb}_s$ is obtained after a non-linear transformation (e.g., tanh) on the $3K$-element vector.

Relation Classification

An attention layer over sentence embeddings (ATT$_{SE}$) is performed over the input sentence embeddings ($\mathbf{emb}_{s_i}, 1 \le i \le N$) to produce a vector that encodes the sentence set, as shown in Figure 3. We adopt the recently proposed self-attention method (Lin et al. 2017). First, each sentence s_i gains an attention score as follows:

$$\alpha_i = \mathbf{v}^{sa} \tanh(\mathbf{W}^{sa} \mathbf{emb}_{s_i})$$

where the matrix $\mathbf{W}^{sa}$ and the vector $\mathbf{v}^{sa}$ are the sentence attention parameters.

Then, the attention scores are normalized into a probability for summing all sentence embeddings into the representation vector of the sentence set S. As discussed in Lin et al. (2016), the attention layer aims to automatically detect noisy training sentences with wrong labels by allocating lower weights to them in this step.[3]

$$\mathbf{emb}_S = \sum_{1 \le i \le N} \left\{ \frac{\exp(\alpha_i)}{\sum\limits_{1 \le k \le N} \exp(\alpha_k)} \mathbf{emb}_{s_i} \right\} \quad (1)$$

A softmax layer is used to produce the probabilities of all relation types. First, we compute a output score vector as follows:

$$\mathbf{o}^s = \mathbf{W}^s \mathbf{emb}_S + \mathbf{b}^s \quad (2)$$

where the matrix $\mathbf{W}^s$ and the bias vector $\mathbf{b}^s$ are model parameters, and $|\mathbf{o}^s| = N_r$ is the number of relation types.

Then, the conditional probability of the relation r for given S is:

$$p(r|S) = \frac{\exp(\mathbf{o}^s[r])}{\sum_{1 \le k \le N_r} \exp(\mathbf{o}^s[k])} \quad (3)$$

Training Objective

Given the training data $\mathcal{D} = \{(S_1, r_1), ..., (S_M, r_M)\}$ consisting of M sentence sets and their relation types resulting from distant supervision, Lin et al. (2016) use the standard cross-entropy loss function as the training objective.

$$Loss(\mathcal{D}) = -\sum_{i=1}^{M} \log p(r_i|S_i) \quad (4)$$

Following Lin et al. (2016), we adopt stochastic gradient descent (SGD) with mini-batch as the learning algorithm and apply dropout (Srivastava et al. 2014) in Equation (2) to prevent over-fitting.

Our SEE Approach

The baseline approach solely relies on the word sequence of a given sentence. However, recent studies show that syntactic structures can help relation extraction by exploiting the dependence relationship between words. Unlike previous works which mainly consider the shortest dependency paths, our proposed approach tries to effectively encode the syntax-aware contexts of entities as extra features for relation classification.

Entity Embedding

Given a sentence and its parse tree, as depicted in Figure 1, we try to encode the focus entity pair as two dense vectors.

Previous work shows that recursive neural networks (RNN) are effective in encoding tree structures (Li et al.

[1] Lin et al. (2016) treat all entity names as single words.

[2] The combination of CNN and three-segment Max-pooling is first proposed by Zeng et al. (2015) and named as piecewise convolutional neural network (PCNN).

[3] Please note that Lin et al. (2016) actually use a more complicated attention schema. However, our preliminary experiments show that the simple self-attention method presented here can achieve nearly the same accuracy. Moreover, the same self-attention mechanism is employed as both local and global attention in our proposed approach.

2015). Inspired by Tai, Socher, and Manning (2015), we propose a simple attention-based tree-GRU to derive the context embedding of an entity over its dependency subtree in a bottom-up order.[4]

Figure 2 illustrates the attention-based tree-GRU. Each word corresponds to a GRU node. Suppose "Vinod_Khosla" is the i-th word w_i in the sentence, and take its corresponding GRU node as an example. The GRU node has two input vectors. The first input vector, denoted as $\boldsymbol{x}_i$, consists of the word embedding, the position embedding, and the dependency embedding of "started → Vinod_Khosla". It is similar to the input in Figure 4 except for the extra dependency embedding.

A dependency embedding is a dense vector that encodes a head-modifier word pair in contexts of all dependency trees, which can express richer semantic relationships beyond word embedding, especially for long-distance collocations. Inspired by Bansal (2015), we adopt the skip-gram neural language model of Mikolov et al. (2013a; 2013b) to learn the dependency embedding. First, we employ the off-shelf Stanford Parser[5] to parse the New York Times (NYT) corpus (Klein and Manning 2003). Then, given a father-child dependency $p \rightarrow c$, the skip-gram model is optimized to predict all its context dependencies. We use the following basic dependencies in a parse tree as contexts:

$$gp \rightarrow p \quad c \rightarrow gc_1 \quad \ldots \quad c \rightarrow gc_{\#gc}$$

where gp means grandparent; gc means grandchild; $\#gc$ is the total number of grandchildren.

The second input vector of the GRU node of "Vinod_Khosla" is the representation vector of all its children $\mathbf{ch}(i)$, and is denoted as $\boldsymbol{h}_{\mathbf{ch}(i)}$.

Attention over child embeddings (ATT$_{CE}$). Here, we adopt the self-attention for summing the hidden vector of the GRU nodes of its children. Suppose $j \in \mathbf{ch}(i)$, meaning w_j is a child of w_i. We use $\boldsymbol{h}_j$ to represent the hidden vector of the GRU node of w_j. Then, the attention score of $\boldsymbol{h}_j$ is:

$$\alpha_j^i = \mathbf{v}^{ch} \tanh(\mathbf{W}^{ch} \boldsymbol{h}_j)$$

where $\mathbf{v}^{ch}$ and $\mathbf{W}^{ch}$ are shared attention parameters.

Then, the children representation vector is computed as:

$$\boldsymbol{h}_{\mathbf{ch}(i)} = \sum_{j \in \mathbf{ch}(i)} \left\{ \frac{\exp(\alpha_j^i)}{\sum\limits_{k \in \mathbf{ch}(i)} \exp(\alpha_k^i)} \mathbf{h}_j \right\} \tag{5}$$

We expect that the ATT$_{CE}$ mechanism can be helpful for producing better representation of the father by 1) automatically detecting informative children via higher attention weights; 2) whereas lowering the weights of incorrect dependencies due to parsing errors.

Given the two input vectors $\boldsymbol{x}_i$ and $\boldsymbol{h}_{\mathbf{ch}(i)}$, the GRU node (Cho et al. 2014) computes the hidden vector of w_i as follows:

$$\begin{aligned}
\boldsymbol{z}_i &= \sigma(\mathbf{W}^z \boldsymbol{x}_i + \mathbf{U}^z \boldsymbol{h}_{\mathbf{ch}(i)} + \mathbf{b}^z) \\
\boldsymbol{r}_i &= \sigma(\mathbf{W}^r \boldsymbol{x}_i + \mathbf{U}^r \boldsymbol{h}_{\mathbf{ch}(i)} + \mathbf{b}^r) \\
\widetilde{\boldsymbol{h}}_i &= \tanh(\mathbf{W}^{\widetilde{h}} \boldsymbol{x}_i + \mathbf{U}^{\widetilde{h}} (\boldsymbol{r}_i \circ \boldsymbol{h}_{\mathbf{ch}(i)}) + \mathbf{b}^{\widetilde{h}}) \\
\boldsymbol{h}_i &= \boldsymbol{z}_i \circ \boldsymbol{h}_{\mathbf{ch}(i)} + (1 - \boldsymbol{z}_i) \circ \widetilde{\boldsymbol{h}}_i
\end{aligned} \tag{6}$$

where σ is the sigmoid function, and the $\circ$ is the element-wise multiplication, $\mathbf{W}^*$ and $\mathbf{U}^*$ are parameter matrices of the model, $\mathbf{b}^*$ is the bias vectors, $\boldsymbol{z}_i$ is the update gate vector and $\boldsymbol{r}_i$ is the reset gate vector.

Finally, we use $\boldsymbol{h}_i$ as the representation vector of the entity context of "Vinod_Khosla". In the same manner, we can compute the entity context embedding of "Khosla_Ventures".

Augmented Relation Classification

Again, we suppose there are N sentences $S = \{s_1, ..., s_N\}$ that contain the focus entity pair e' and e''. The corresponding word indices that e' occurs in S are respectively $\{j'_1, ..., j'_N\}$, whereas the positions of e'' are $\{j''_1, ..., j''_N\}$.

As discussed above, the entity context embedding of e' in the i-th sentence s_i is the hidden vector of the GRU node of $w_{j'_i}$ (which is e').

$$\mathbf{emb}_{s_i, e'} = \boldsymbol{h}_{j'_i}^{s_i}$$

Similarly, the entity context embedding of e'' in s_i is:

$$\mathbf{emb}_{s_i, e''} = \boldsymbol{h}_{j''_i}^{s_i}$$

Figure 3 shows the overall framework of our proposed approach. The input consists of three parts, i.e., the sentence embeddings , the context embeddings of e', and the context embeddings of e'':

$$\begin{array}{c}
\{\mathbf{emb}_{s_1} \ \ \ldots \ \ \mathbf{emb}_{s_N}\} \\
\{\mathbf{emb}_{s_1, e'} \ \ \ldots \ \ \mathbf{emb}_{s_N, e'}\} \\
\{\mathbf{emb}_{s_1, e''} \ \ \ldots \ \ \mathbf{emb}_{s_N, e''}\}
\end{array} \tag{7}$$

Similar to sentence attention in the baseline system, and for maximizing utilization the valid information in sentence and entity context, we enhance the model by separately applying attention to both the sentence and entity context embeddings simultaneously.

Attention over entity embeddings (ATT$_{EE}$). Similar to the attention over sentence embeddings in Equation (1), we separately apply attention to the three parts in Equation (7) and generate the final representation vectors of S, e', and e'' on the sentence set, i.e., $\mathbf{emb}_S$, $\mathbf{emb}_{e'}$, $\mathbf{emb}_{e''}$, respectively. We omit the formulas for brevity.

Then, the next step is to predict the relation type based on the three sentence set-level embeddings. Here, we propose two strategies.

The concatenation strategy (CAT). The most straightforward way is to directly concatenate the three embeddings

[4] In fact, Tai, Socher, and Manning (2015) propose two extensions to the basic LSTM architecture, i.e., the *N-ary tree-LSTM* and the *child-sum tree-LSTM*. However, the *N-ary tree-LSTM* assumes that the maximum number of children is N, which may be unsuitable for our task since $N = 19$ would be too large for our dataset. The *child-sum tree-LSTM* can handle arbitrary number of children, but achieves consistently lower accuracy than the simple attention-based tree-GRU according to our preliminary experiments.

[5] https://nlp.stanford.edu/software/lex-parser.shtml, and the version is 3.7.0

and obtain the score vector of all relation types via a linear transformation.

$$\mathbf{o}^{cat} = \mathbf{W}^{cat}[\mathbf{emb}_S; \mathbf{emb}_{e'}; \mathbf{emb}_{e''}] + \mathbf{b}^{cat} \tag{8}$$

where the matrix $\mathbf{W}^{cat}$ and the bias vector $\mathbf{b}^{cat}$ are model parameters.

The translation strategy (TRANS). According to Equation (8), the CAT strategy cannot capture the interactions among the three embeddings, which is counter-intuitive considering that the relation type must be closely related with both entities simultaneously. Inspired by the widely used TransE model (Bordes et al. 2013), which regards the embedding of a relation type r as the difference between two entity embeddings ($\mathbf{emb}_r = \mathbf{emb}_{e''} - \mathbf{emb}_{e'}$), we use the vector difference to produce a relation score vector via a linear transformation.

$$\mathbf{o}^{see} = \mathbf{W}^{see}(\mathbf{emb}_{e''} - \mathbf{emb}_{e'}) + \mathbf{b}^{see} \tag{9}$$

where $\mathbf{o}^{see}$ represents the score vector according to the entity context embeddings, and the matrix $\mathbf{W}^{see}$ and the bias vector $\mathbf{b}^{see}$ are model parameters.

To further utilize the sentence embeddings, we compute another relation score vector $\mathbf{o}^s$ according to Equation (2), which is the same with the baseline. Then we combine the two score vectors.

$$\mathbf{o}^{trans} = \boldsymbol{\alpha} \circ \mathbf{o}^s + (1 - \boldsymbol{\alpha}) \circ \mathbf{o}^{see} \tag{10}$$

where $\circ$ denotes element-wise product (a.k.a. Hadamard product), and $\boldsymbol{\alpha}$ is the interpolation vector for balancing the two parts. Actually, we have also tried a few different ways for combining the two score vectors, but found that the formula presented here consistently performs best.

Finally, we apply softmax to transform the score vectors ($\mathbf{o}^{cat}$ or $\mathbf{o}^{trans}$) into conditional probabilities, as shown in Equation (3), and adopt the same training objective and optimization algorithm with the baseline.

Experiments

In this section, we present the experimental results and detailed analysis.

Datasets. We adopt the benchmark dataset developed by Riedel, Yao, and McCallum (2010), which has been widely used in many recent works (Hoffmann et al. 2011; Surdeanu et al. 2012; Lin et al. 2016; Ji et al. 2017). Riedel, Yao, and McCallum (2010) use Freebase as the distant supervision source and the three-year NYT corpus from 2005 to 2007 as the text corpus. First, they detect the entity names in the sentences using the Stanford named entity tagger (Finkel, Grenager, and Manning 2005) for matching the Freebase entities. Then, they project the entity-relation tuples in Freebase into the all sentences that contain the focus entity pair. The dataset contains 53 relation types, including a special relation “NA” standing for no relation between the entity pair. We adopt the standard data split (sentences in 2005-2006 NYT data for training, and sentences in 2007 for evaluation). The training data contains $522,611$ sentences, $281,270$ entity pairs and $18,252$ relational facts. The testing set contains $172,448$ sentences, $96,678$ entity pairs and $1,950$ relational facts.

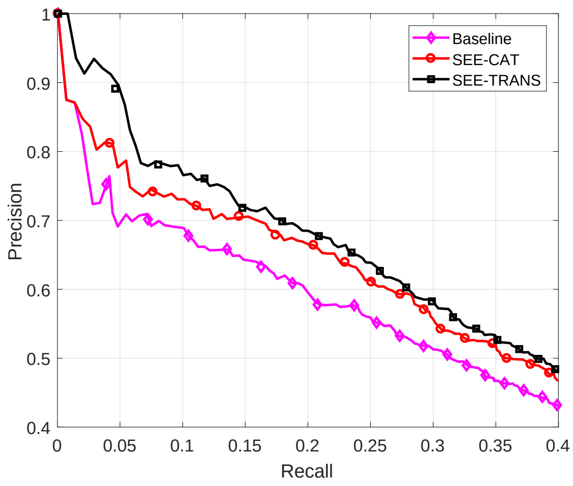

Figure 5: Comparison of the baseline and our approach under two different strategies.

Evaluation metrics. Following the practice of previous works (Riedel, Yao, and McCallum 2010; Zeng et al. 2015; Ji et al. 2017), we employ two evaluation methods, i.e., the *held-out evaluation* and the *manual evaluation*. The held-out evaluation only compares the entity-relation tuples produced by the system on the test data against the existing Freebase entity-relation tuples, and report the precision-recall curves.

Manual evaluation is performed to avoid the influence of the wrong labels resulting from distant supervision and the incompleteness of Freebase data, and report the Top-N precision $P@N$, meaning the the precision of the top N discovered relational facts with the highest probabilities.

Hyperparameter tuning. We tune the hyper-parameters of all the baseline and our proposed models on the training dataset using three-fold validation. We adopt the brute-force grid search to decide the optimal hyperparameters for each model. We try $\{0.1, 0.15, 0.2, 0.25\}$ for the initial learning rate of SGD, $\{50, 100, 150, 200\}$ for the mini-batch size of SGD, $\{50, 80, 100\}$ for both the word and the dependency embedding dimensions, $\{5, 10, 20\}$ for the position embedding dimension, $\{3, 5, 7\}$ for the convolution window size l, and $\{60, 120, 180, 240, 300\}$ for the filter number K. We find the configuration $0.2/150/50/50/5/3/240$ works well for all the models, and further tuning leads to slight improvement.

Held-out Evaluation

Comparison results with the baseline is presented in Figure 5. “SEE-CAT” and “SEE-TRANS” are our proposed approach with the CAT and TRANS strategies respectively. We can see that both our approaches consistently outperform the baseline method. It is also clear that “SEE-TRANS” is superior to “SEE-CAT”. This is consistent with our intuition that the TRANS strategy can better capture the interaction between the two entities simultaneously. In the following results, we adopt “SEE-TRANS” for further experiments and analysis.

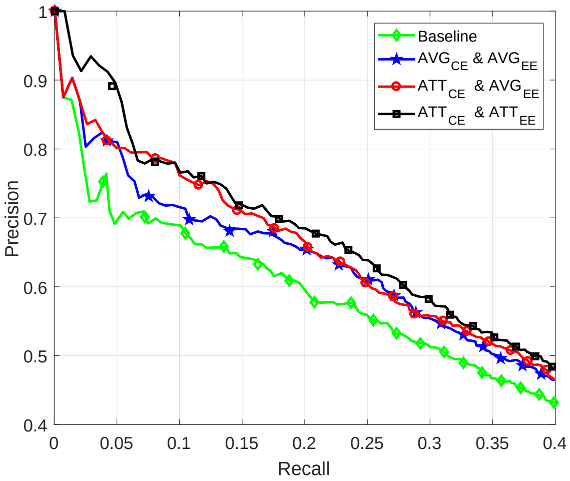

Figure 6: Effect of self-attention components.

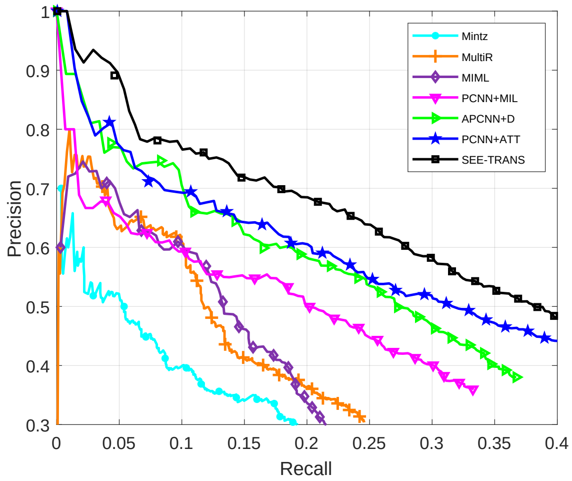

Figure 7: Comparison with previous results.

The effect of self-attention components is investigated in Figure 6. To better understand the two self-attention components used in our "SEE" approach, we replace attention with an average component, which assumes the same weight for all input vectors and simply use the averaged vector as the resulting embedding. Therefore, the "ATT_{CE}" in Figure 2 is replaced with "AVG_{CE}", and "ATT_{EE}" in Figure 3 is replaced with "AVG_{EE}".

The four precision-recall curves clearly show that both self-attention components are helpful for our model. In other words, the attention provides a flexible mechanism that allows the model to distinguish the contribution of different input vectors, leading to better global representation of instances.

Comparison with previous works is presented in Figure 7. We select six representative approaches and directly get all their results from Lin et al. (2016) and Ji et al. (2017) for comparison[6], which fall into two categories:

- Traditional discrete feature-based methods: (1) **Mintz** (Mintz et al. 2009) proposes distant supervision paradigm and uses a multi-class logistic regression for classification. (2) **MultiR** (Hoffmann et al. 2011) is a probabilistic graphical model with multi-instance learning under the "at-least-one" assumption. (3) **MIML** (Surdeanu et al. 2012) is also a graphical model with both multi-instance and multi-label learning.
- Neural model-based methods: (1) **PCNN+MIL** (Zeng et al. 2015) proposes piece-wise (three-segment) CNN to obtain sentence embeddings. (2) **PCNN+ATT** (Lin et al. 2016) corresponds to our baseline approach and achieves state-of-the-art results. (3) **APCNN+D** (Ji et al. 2017) uses external background information of entities via an attention layer to help relation classification.

From the results, we can see that our proposed approach "SEE-TRANS" consistently outperforms all other approaches by large margin, and achieves new state-of-the-art results on this dataset, demonstrating the effectiveness of leveraging syntactic context for better entity representation for distant supervision relation extraction.

Accuracy	Top 100	Top 200	Top 500	Average
Mintz	0.77	0.71	0.55	0.676
MultiR	0.83	0.74	0.59	0.720
MIML	0.85	0.75	0.61	0.737
PCNN+MIL	0.84	0.77	0.64	0.750
PCNN+ATT	0.86	0.83	0.73	0.807
APCNN+D	0.87	0.83	0.74	0.813
Baseline	0.86	0.84	0.73	0.810
SEE-TRANS	**0.91**	**0.87**	**0.77**	**0.850**

Table 1: Manual evaluation results.

Manual Evaluation

Due to existence of noises resulting from distance supervision in the test dataset under the held-out evaluation, we can see that there is a sharp decline in the precision-recall curves in most models in Figure 7. Therefore, we manually check the top-500 entity-relation tuples returned by all the eight approaches.[7] Table 1 shows the results. We can see that (1) our re-implemented baseline achieve nearly the same performance with Lin et al. (2016); (2) our proposed SEE-TRANS achieves consistently higher precision at different N levels.

Case Study

Table 2 present a real example for case study. The entity-relation tuple is (*Bruce Wasserstein*, *company*, *Lazard*). There are four sentences containing the entity pair. The baseline approach only uses the word sequences as the input, and learn the sentence embeddings for relation classification. Due to the lack of sufficient information, the *NA* relation type receives the highest probability of 0.735. In contrast, our proposed SEE-TRANS can correctly recognize the

[6] We are very grateful to Dr. Lin and Dr. Ji for their help.

[7] Please note that there are many overlapping results among different approaches, thus requiring much less manual effort.

Tuple	Sentences	Syntax-aware Entities	Baseline	SEE-TRANS
company (Bruce Wasserstein, Lazard)	1. A record profit at [Lazard], the investment bank run by [Bruce Wasserstein], said that strength in its merger advisory ...	**Bruce Wasserstein**: 1. the chairman of Lazard. 2. the current Lazard chief executive. **Lazard**: 1. the investment bank run by Bruce Wasserstein.	*NA* (0.735) *company* (0.256) *founders* (0.002)	*company* (0.650) *NA* (0.250) *founders* (0.028)
	2. The buyout executives ... huddled in a corner, and [Bruce Wasserstein], the chairman of [Lazard], chatted with richard d. parsons , the chief executive of time warner .			
	3. [Lazard], the investment bank run by [Bruce Wsserstein], said yesterday that strength in its merger-advisory ...			
	4. Along with the deals and intrigue ... maneuverings in martha 's vineyard as well as the tax strategies of the current [Lazard] chief executive [Bruce Wasserstein].			

Table 2: Case study: a real example for comparison.

relation type as *company* with the help of the rich contexts in the syntactic parse trees.

Related Work

In this section, we first briefly review the early previous studies on distant supervision for RE. Then we introduce the systems using the neural RE framework.

In the supervised paradigm, relation extraction is considered to be a multi-class classification problem and needs a great deal of annotated data, which is time consuming and labor intensive. To address this issue, Mintz et al. (2009) aligns plain text with Freebase by distant supervision, and extracts features from all sentences and then feeds them into a classifier. However, the distant supervision assumption neglects the data noise. To alleviate the wrong label problem, Riedel, Yao, and McCallum (2010) models distant supervision for relation extraction as a multi-instance single-label problem. Further, Hoffmann et al. (2011) and Surdeanu et al. (2012) adopt multi-instance multi-label learning in relation extraction, and use the shortest dependency path as syntax features of relation. The main drawback of these methods is that their performance heavily relies on a manually designed set of feature templates which are difficult to design.

Neural networks (Bengio 2009) have been successfully used in many NLP tasks such as part-of-speech tagging (Santos and Zadrozny 2014), parsing (Socher et al. 2013), sentiment analysis (Dos Santos and Gatti 2014), and machine translation (Cho et al. 2014). As for relation extraction, neural networks have also been successfully applied and achieved advanced performance for this field. Socher et al. (2012) uses a recursive neural network in relation extraction. Zeng et al. (2014) adopts an end-to-end convolutional neural network in this task, and Zeng et al. (2015) further combines at-least-one multi-instance learning and assumes that only one sentence expresses the relation for each entity pair, which doesn't make full use of the supervision information. Lin et al. (2016) proposes to use attention to select valid sentences, which shows promising results. However, sentence embeddings are used to represent relation between entities, may result in semantic shifting problem, since the relation between entities is just a small part of a sentence.

All the above work on neural networks mainly use words to generate sentence embeddings, and use them for classification. Besides the word-level information, syntax information also has been considered by some researchers, for example, Miwa and Bansal (2016) and Cai, Zhang, and Wang (2016) model the shortest dependency path as a factor for the relation between entities, but they ignore that the tree information can be used to model the syntax roles the entities played. The syntax roles are important for relation extraction. Different from the above previous studies, we enrich the entity representations with syntax structures by considering the subtrees rooted at entities.

Conclusion

In this paper, we propose to learn syntax-aware entity embedding from dependency trees for enhancing neural relation extraction under the distant supervision scenario. We apply the recursive tree-GRU to learn sentence-level entity embedding in a parse tree, and utilize both intra-sentence and inter-sentence attentions to make full use of syntactic contexts in all sentences. We conduct experiments on a widely used benchmark dataset. The experimental results show that our model consistently outperforms both the baseline and the state-of-the-art results. This demonstrates that our approach can effectively learn entity embeddings, and the learned embeddings are able to help the task of relation extraction.

For future, we would like to further explore external knowledge as Ji et al. (2017) to obtain even better entity embeddings. We also plan to apply the proposed approach to other datasets or languages.

Acknowledgments

The research work is supported by the National Key Research and Development Program of China under Grant No.2017YFB1002104, and the National Natural Science Foundation of China (61672211). This work is partially supported by the joint research project of Alibaba and Soochow University.

References

[2015] Bansal, M. 2015. Dependency link embeddings: Continuous representations of syntactic substructures. In *Proceedings of NAACL-HLT*, 102–108.

[2009] Bengio, Y. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning* 2(1):1–127.

[2013] Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*, 2787–2795.

[2016] Cai, R.; Zhang, X.; and Wang, H. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of ACL*, 756–765.

[2014] Cho, K.; Van Merriënboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, 1724–1734.

[2014] Dos Santos, C. N., and Gatti, M. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING*, 69–78.

[2005] Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL*, 363–370.

[2011] Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL*, 541–550.

[2017] Ji, G.; Liu, K.; He, S.; and Zhao, J. 2017. Distant supervision for relation extraction with sentence-level attention and entity descriptions. In *AAAI*, 3060–3066.

[2003] Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL*, 423–430.

[2015] Li, J.; Luong, M. T.; Jurafsky, D.; and Hovy, E. 2015. When are tree structures necessary for deep learning of representations? In *Proceedings of EMNLP*, 2304–2314.

[2016] Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, 2124–2133.

[2017] Lin, Z.; Feng, M.; Santos, C. N. D.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

[2013a] Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[2013b] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 3111–3119.

[2009] Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*, 1003–1011.

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

[2013] Riedel, S.; Yao, L.; McCallum, A.; and Marlin, B. M. 2013. Relation extraction with matrix factorization and universal schemas. In *HLT-NAACL*, 74–84.

[2010] Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases* 148–163.

[2014] Santos, C. D., and Zadrozny, B. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning*, 1818–1826.

[2012] Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, 1201–1211.

[2013] Socher, R.; Bauer, J.; Manning, C. D.; and Ng, A. Y. 2013. Parsing with compositional vector grammars. In *Proceedings of ACL*, 455–465.

[2014] Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

[2012] Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*, 455–465.

[2015] Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

[2014] Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, 2335–2344.

[2015] Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, 1753–1762.