Beyond Word Attention: Using Segment Attention in Neural Relation Extraction

Bowen Yu[1,2], **Zhenyu Zhang**[1,2], **Tingwen Liu**[1 *], **Bin Wang**[3], **Sujian Li**[4] and **Quangang Li**[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Xiaomi AI Lab, Xiaomi Inc., Beijing, China
[4]Key Laboratory of Computational Linguistics, Peking University, MOE, China
{yubowen, zhangzhenyu1996, liutingwen, liquangang}@iie.ac.cn, wangbin11@xiaomi.com,
lisujian@pku.edu.cn

Abstract

Relation extraction studies the issue of predicting semantic relations between pairs of entities in sentences. Attention mechanisms are often used in this task to alleviate the inner-sentence noise by performing soft selections of words independently. Based on the observation that information pertinent to relations is usually contained within segments (continuous words in a sentence), it is possible to make use of this phenomenon for better extraction. In this paper, we aim to incorporate such segment information into neural relation extractor. Our approach views the attention mechanism as linear-chain conditional random fields over a set of latent variables whose edges encode the desired structure, and regards attention weight as the marginal distribution of each word being selected as a part of the relational expression. Experimental results show that our method can attend to continuous relational expressions without explicit annotations, and achieve the state-of-the-art performance on the large-scale TACRED dataset.

1 Introduction

There has been significant historic interest in relation extraction (RE), which aims to extract semantic relationships between two target entities from plain text. Regarding such task as a simple text classification problem is undesirable because of the inner-sentence noise [Liu *et al.*, 2018]. To explain the influence of word-level noise, we consider the sentence in Figure 1 as an example. The sub-sentence "*Edsel Ford, the only child of Henry Ford*" keeps enough words to express the relation *children*, and the other words could be regarded as noise that may hamper the extractor's performance.

To alleviate the influence of word-level noise within sentences, many efforts have been devoted to get rid of irrelevant words [Xu *et al.*, 2015; Zhang *et al.*, 2017; Zhang *et al.*, 2018; Liu *et al.*, 2018], especially the recent state-of-the-art attention-based methods [Zhang *et al.*, 2017; Lee *et al.*, 2019]. Specifically, current attention scheme used in RE computes the attention score for each word to indicate how well the word can express the relation between the two entities. This mechanism can be viewed as the process of performing soft selections of individual words independently, regardless of the rich dependencies among the words that describe the relation.

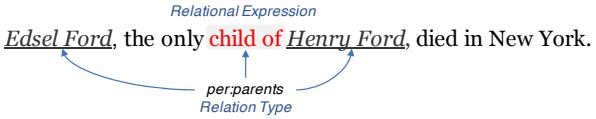

Figure 1: An example modified from the TACRED dataset. The relational expression is highlighted with a box and points to its corresponding entity pair.

As we can see from the example given in Figure 1, the relational expression associated with target entities may be in the form of a segment structure, where the segment refers to a consecutive subsequence of words in sentence. The phrase "*child of*" is the relational expression of entity pair (*Edsel Ford, Henry Ford*) and represents the relation "*children*". It is difficult for model to distinguish between "*children*" and "*parents*" if its attention layer only focuses on the individual word "*child*" rather than the segment "*child of*". Furthermore, we sample out 200 examples from the standard dataset and annotate the relational expression of each sentence manually. We find that half of the relational expressions are in the form of segment and longer than 2 words, which means that accurately extracting and modeling such segment information can be extremely crucial. While one might argue that these dependencies can be learned implicitly by a deep model with substantial amount of data, we believe it is still useful to provide such shallow structural information as prior knowledge.

To capture such relational expressions, we incorporate a layer that is analogous to conditional random fields (CRF) [Lafferty *et al.*, 2001] in the attention modeling process. As a result, our attention mechanism could provide a probabilistic framework for calculating the weights of words globally conditioned on the full sentence. In other words, the weight of one word is expected to impact on the weights of neighboring words. This can be viewed as an extension of the standard attention mechanism, and we call such a novel attention mechanism segment attention in this paper.

Our approach views the attention mechanism as a linear-

*Corresponding Author

chain CRF over a set of latent variables whose edges encode the desired structure. Concretely, we assume that there is a one-to-one match between variables and words. Each binary variable indicates whether its corresponding word is part of a relational expression or not (two states in our problem: selected word and non-selected word). Then we regard the marginal distribution of each word being selected as the attention weight. This distribution can be calculated efficiently in linear-time using the forward-backward algorithm. Furthermore, we introduce two additional regularizers to ensure our model attends to continuous regions. The first regularizer, which named as transition regularizer, discourages frequent transitions between different states and aims to achieve continuous identical states. The second regularizer called sparse regularizer tries to focus on few words that really matter and return sparse weight distribution.

Specifically, our model consists of four layers: a position-aware input layer aims to take position information into consideration, a BiLSTM layer [Graves *et al.*, 2013] that runs through the words in the sentence sequentially to get contextual information for each word, and a segment attention layer works as parameterized pooling to distill the relation information and learn a representation of the given sentence, which is fed to the final classification layer.

To summarize, our contributions are as follows:

- We propose a novel segment attention based sequence model (SA-LSTM) for RE task, which is capable of learning relational expressions and capturing dependencies between target entities and their relations.
- Experiments are conducted on the TACRED dataset. Results show that our model achieves the state-of-the-art performance on the fully-supervised RE task.
- We conduct qualitative analyses to understand how our model works with the help of segment attention, including evaluation of the extracted relational expressions.

2 Related Work

There are several studies for solving relation extraction task. Early methods used handcrafted features through a series of NLP tools or manually designing kernels [Rink and Harabagiu, 2010]. These approaches use high-level lexical and syntactic features obtained from NLP tools and manually designing kernels, but the classification models relying on such features suffer from error propagation of the tools.

Recent studies have found neural models effective in relation extraction, deep neural networks have outperformed previous models using handcraft features. Zeng *et al.*[2014] employed a deep convolutional neural network for extracting lexical and sentence level features. Santos *et al.* [2015] proposed model for learning vector of each relation class using ranking loss to reduce the impact of artificial classes. Zhou *et al.* [2016] used bidirectional recurrent neural network to learn long-term dependency between entity pairs.

Apart from neural models over word sequences, incorporating dependency trees into neural models has also been shown to improve relation extraction performance by capturing long-distance relations. Xu *et al.* [2015] generalized the idea of dependency path kernels by applying a LSTM network over the shortest dependency path between entities. Liu *et al.* [2015] first applied a recursive network over the subtrees rooted at the words on the dependency path and then applied a CNN over the path. Zhang *et al.* [2018] applied a combination of pruning strategy and graph convolutions to the dependency tree. The resulting model achieved best performance on the TACRED dataset.

More recently, some researchers have proposed attention-based models which can focus to the most important semantic information in a sentence. Zhou *et al.* [2016] combined attention mechanisms with BiLSTM. Xiao and Liu [2016] splited the sentence into two entities and used two attention-based BiLSTM hierarchically [21]. Wang *et al.* [2016] proposed attention-based CNN using word level attention mechanism that is able to better determine which parts of the sentence are more influential. Zhang *et al.* [2017] employed a position-aware attention mechanism over LSTM outputs, and showed that it outperforms several CNN and dependency-based models by a substantial margin. Du *et al.* [2018] proposed a 2-D matrix-based attention mechanism, which contains multiple vectors, each focusing on different aspects of the sentence. Our model is inspired by structural attention network [Kim *et al.*, 2017] which extends the standard attention to directly model structural dependencies between nearby input elements. Wang and Lu [2018] also applied this sort of architecture to aspect-based sentiment analysis. Different from previous attention model designed for relation extraction, our model is capable of learning phrase-like features and capture reasonable segments as relational expressions.

3 Methodology

Figure 2 gives an illustration of our SA-LSTM model. Next, we detail all components sequentially from bottom to top.

3.1 Input Layer

The input layer of the sentence encoder aims to embed both semantic information and positional information of words into their input embeddings.

Word embedding is able to capture the meaningful semantic regularities of words [Turian *et al.*, 2010]. We use pre-trained d_w-dimensional word embeddings [Pennington *et al.*, 2014] as the basic features.

Position embedding is proposed by [Zeng *et al.*, 2014], which is used to embed the relative distances of each word to the two entities into two d_p-dimensional vectors. By concatenating the distance embeddings for the current word w_i to the both head and tail entities, we get a unified position embedding $\mathbf{p}_i \in \mathbb{R}^{d_p \times 2}$.

For each word w_i, we concatenate its word embedding $\mathbf{w}_i$ and position embedding $\mathbf{p}_i$ to build its input embedding $\mathbf{x}_i = [\mathbf{w}_i; \mathbf{p}_i] \in \mathbb{R}^{d_w + d_p \times 2}$

3.2 BiLSTM Layer

A BiLSTM layer is adopted to capture the contextual information for each word. For simplicity, we denote the operation

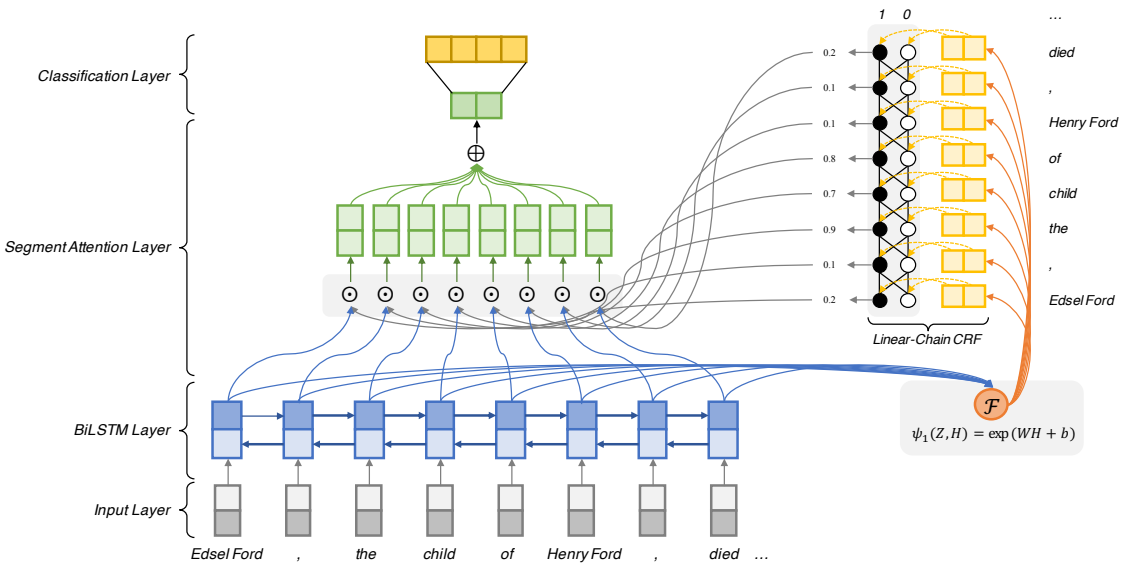

Figure 2: Model architecture shown with an example sentence "*Edsel Ford, the child of Henry Ford, died in New York*". The model encodes the contextual information for each word using BiLSTM. Based on its hidden states, the internal segment attention layer performs soft selections of a consecutive sequence of words by giving higher weights to more relevant contexts.

of an LSTM unit on $\mathbf{x_i}$ as LSTM($\mathbf{x_i}$). Thus, the contextualized word representation is obtained as follows:

$$\mathbf{h}_i = [\overrightarrow{\text{LSTM}}(\mathbf{x}_i); \overleftarrow{\text{LSTM}}(\mathbf{x}_i)], i \in [1, n] \quad (1)$$

where $\mathbf{h}_i \in \mathbb{R}^{2\times d_h}$ and d_h indicates the dimension of hidden state for LSTM. In doing so, we can efficiently make use of past features (via forward states) and future features (via backward states) for a specific time. Moreover, we use $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n\}$ to denote all the word representations generated for the input sentence.

3.3 Segment Attention Layer

Based on the observation that it is usually a segment rather than individual words scattered in the sentence that form meaningful information, we incorporate a segment attention layer to perform soft selections of a sequence of words.

Similar to the standard attention used in RE, our segment attention is also a linear combination of the input representations where the weight scalar is between [0, 1] and represents how much attention should be focused on each input. The key difference between our scheme and previous attention mechanism used in [Zhou *et al.*, 2016] and [Zhang *et al.*, 2017] is the calculation of attention weights. Specifically, we introduce a discrete latent binary variable $z \in \{0, 1\}$ for each word. This variable indicates whether its corresponding word is part of a relational expression or not. Under this definition, the representation of the given sequence $\mathbf{m}$ is defined as the expectation of hidden states with the probability that its corresponding word is selected. Equation 2 gives a general form of this function.

$$\mathbf{m} = \sum_i p(z_i = 1|\mathbf{H})\mathbf{h}_i \quad (2)$$

In order to derive $p(z_i = 1|\mathbf{H})$, we incorporate linear-chain conditional random fields (CRF) to specify the dependencies between these latent variables. For a random variable over data sequences $\mathbf{H}$, and a random variable over corresponding label sequences $\mathbf{z}$, CRF provide a probabilistic framework for calculating the probability of $\mathbf{z}$ globally conditioned on $\mathbf{H}$ [Lafferty *et al.*, 2001]. $\mathbf{H}$ and $\mathbf{z}$ may have a natural graph structure. Formally, we use $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n\}$ to represent a generic input sequence where $\mathbf{h}_i$ is the BiLSTM hidden state of the i-th word. $\mathbf{z} = [z_1, \cdots, z_n]$ represents a generic sequence of labels for $\mathbf{H}$. The probabilistic model for sequence CRF defines a family of conditional probability $p(\mathbf{z}|\mathbf{H})$ over all possible label sequences $\mathbf{z}$ given $\mathbf{H}$ with the following form:

$$p(\mathbf{z}|\mathbf{H}) = \frac{1}{Z(\mathbf{H})} \prod_{c\in C} \psi(\mathbf{z_c}, \mathbf{H}) \quad (3)$$

$$Z(\mathbf{H}) = \sum_{\mathbf{z}'\in\mathcal{Z}} \prod_{c\in C} \psi(\mathbf{z}'_\mathbf{c}, \mathbf{H}) \quad (4)$$

where $\mathcal{Z}$ denotes the set of possible label sequences $\mathbf{z}$, $Z(\mathbf{H})$ is the normalization constant that makes the probability of all state sequences sum to one, $\mathbf{z}_c$ indicates the subset of $\mathbf{z}$ given by individual clique c and $\psi(\mathbf{z_c}, \mathbf{H})$ is the potential function of this clique defined as:

$$\prod_{c\in C} \psi(\mathbf{z_c}, \mathbf{H}) = \prod_{i=1}^{n} \psi_1(z_i, \mathbf{h}_i) \prod_{i=1}^{n-1} \psi_2(z_i, z_{i+1}) \quad (5)$$

We define two types of feature functions: vertex feature ψ_1 and edge feature ψ_2. Vertex feature $\psi_1(z_i, \mathbf{h}_i)$ represents the mapping from the input $\mathbf{h}_i$ to output z_i through a single layer neural network. Edge feature $\psi_2(z_i, z_{i+1})$ models the transition from i-th state to i+1-th for a pair of consecutive time steps. Note that this transition matrix is position independent.

$$\psi_1(z_i, \mathbf{H}) = \exp(\mathbf{W}^v_{z_i} \cdot \mathbf{h}_i + b) \quad (6)$$

$$\psi_2(z_i, z_{i+1}) = \exp(\mathbf{W}^t_{z_i, z_{i+1}}) \quad (7)$$

Here $\mathbf{W}^v \in \mathbb{R}^{2\times 2d_h}$ maps context representation to the feature score of each latent state, $\mathbf{W}^t \in \mathbb{R}^{2\times 2}$ is a transition matrix defined for each pair of latent state and $\mathbf{W}^t_{z_i, z_{i+1}}$ is the transition score from z_i to z_{i+1}.

As mentioned above, our purpose is to compute the marginal probability $p(z_i = 1|\mathbf{H})$ at each position in the sequence which can be computed by a dynamic programming inference procedure similar to the forward-backward procedure for HMM [Lafferty *et al.*, 2001]. We can define the "forward values" of the i-th timestep $\alpha_i(z|\mathbf{H})$ by setting $\alpha_1(z|\mathbf{H})$ equal to the probability of starting with state z, $z \in \{0, 1\}$, and then iterate as follows:

$$\alpha_{i+1}(z|\mathbf{H}) = \sum_{z' \in \{0,1\}} \alpha_i(z'|\mathbf{H})\psi_1(z, \mathbf{h}_{i+1})\psi_2(z', z) \quad (8)$$

The "backward values" $\beta_i(z|\mathbf{H})$ can be defined similarly. After that, we calculate the marginal probability of each word being a part of the relational expression given the whole sentence sequence by:

$$p(z_i = 1|\mathbf{H}) = \frac{\alpha_i(1|\mathbf{H}) * \beta_i(1|\mathbf{H})}{Z(\mathbf{H})} \quad (9)$$

Thus we can compute the final representation $\mathbf{m}$ used for classification by combining both Equation 2 and Equation 9. Furthermore, given the conditional probability of the state sequence defined by a CRF in Equation 3, we can also obtain the most probable labeling sequence using a Viterbi decoding algorithm, corresponding to the word sequence with a label of 1 denoting its corresponding word is part of a relational expression and 0 denoting the non-informative word. In other words, the latent relational expressions can be extracted explicitly by maximizing Equation 3, which allows us to have an intuitive understanding of our model behavior and evaluate segment attention from another aspect.

3.4 Classification Layer

To compute the output distribution $p(r)$ over relation labels, a linear layer followed by a softmax is applied to the representation $\mathbf{m}$, which represents a summary of the input sequence:

$$p(r|\mathbf{m}) = \mathrm{softmax}(\mathbf{W}_\mathrm{r} \cdot \mathbf{m} + \mathrm{b}_\mathrm{r}) \quad (10)$$

where $\mathbf{W}_r \in \mathbb{R}^{2\times d_h}$ maps the relation vector $\mathbf{m}$ to the feature score for each relation label and b_r is a bias term.

3.5 Objective Function

After incorporating the segment attention into the BiLSTM, our final model is illustrated in Figure 2. The attention-based BiLSTM component is associated with the cross entropy loss of relation extraction. The loss function is given below:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} -y_i \log p(y_i) \quad (11)$$

In addition, based on the observation that relational expressions are usually segments rather than disconnected words, we hold that frequent transitions between different states should be discouraged. Inspired by [Wang and Lu, 2018], we introduce the transition regularizer to encourage the state to stay the same.

$$\Omega_t = \max(0, \mathbf{W}^t_{1,0} - \mathbf{W}^t_{1,1}) + \max(0, \mathbf{W}^t_{0,1} - \mathbf{W}^t_{0,0}) \quad (12)$$

Specifically, it enforces the transition feature value between different states to be smaller than the one between the same state. The second regularizer tries to enforce the model to attend to few words that really matter and return sparse weight distribution:

$$\Omega_s = \sum_{i=1}^{n} p(z_i = 1|\mathbf{H}) \quad (13)$$

The final objective function of our model is defined as:

$$L(\theta) = J(\theta) + \lambda_1\Omega_t + \lambda_2\Omega_s \quad (14)$$

where λ_1 and λ_2 are hyperparameters that control the weights of each regularizer.

4 Experiments

4.1 Dataset and Metric

We conduct experiments on the recently widely used benchmark TACRED dataset introduced in [Zhang *et al.*, 2017], which is the currently largest supervised dataset for relation extraction. It contains over 106k entity pairs collected from the TAC KBP evaluations 2009–2014. TACRED includes 41 relation types and a special *no_relation* class indicating that the relation expressed in the sentence is not among the 41 types. Entities in TACRED are replaced by corresponding entity types, subjects classified into person and organization, and objects categorized into 16 fine-grained classes (*e.g.* date, location, title). We evaluate the models using the official scorer in terms of the Macro-F1 score. For fair comparisons, we report the test score of the run with the median validation score among 5 randomly initialized runs following the evaluation protocol used in [Zhang *et al.*, 2017].

4.2 Implementation Details

Following popular choices and previous work, we employ the "entity mask" strategy where we replace each subject (and object similarly) entity with a special *SUBJ-<NER>* token. we also adopt the "multi-channel" strategy by concatenating the input word embeddings with part-of-speech (POS) and named entity recognition (NER) embeddings. We run Stanford CoreNLP [Manning *et al.*, 2014] to obtain the POS and NER annotations. These strategies are also used by the methods that we compare against.

We use the 300 dimension Glove embeddings [Pennington *et al.*, 2014] to initialize word embeddings. we randomly initialize the POS, NER and position embeddings with 30-dimension vectors, by drawing from a normal distribution with $\mu = 0.0$ and $\sigma = 0.01$. Dropout with $p = 0.5$ used after the input layer and before the classifier layer. λ_1 and λ_2 are chosen from [0,0.2] via grid search. For LSTM, we set the hidden dimension size to 300 and use 2-layer stacked BiLSTM. The model is trained using stochastic gradient descent for 30 epochs with the initial learning rate of 1 and the weight decay of 0.5. All the hyper-parameters are tuned on the validation set.

4.3 Comparison Models

We compare our model against the following baseline models for relation extraction.

Pattern: The core component of TAC KBP 2015 winning system, which uses a total of 4,528 surface patterns and 169 dependency patterns to extract relations [Angeli *et al.*, 2015].

LR: It is trained on 2 million bootstrapped examples and uses a comprehensive feature set [Zhang *et al.*, 2017].

CNN-PE: It builds a convolutional model to learn sentence features, and uses position vectors to indicate the relative distances of current word to two entities [Zeng *et al.*, 2014].

PCNN: Based on CNN-PE, it replaces the max-pooling operation with piece-wise max-pooling and achieves improved results [Zeng *et al.*, 2015].

SDP-LSTM: It applies a neural sequence model iteratively along the shortest dependency path between target entities [Xu *et al.*, 2015].

Tree-LSTM: It is a recursive model [Tai *et al.*, 2015] that generalizes the LSTM to arbitrary dependency tree structures.

PA-LSTM: It employs a position-aware attention mechanism over LSTM outputs, and outperforms several CNN and dependency-based models [Zhang *et al.*, 2017].

C-GCN: It applies a combination of pruning strategy and graph convolutions to the dependency tree, which is a state-of-the-art on the TACRED dataset [Zhang *et al.*, 2018] .

SA-LSTM: Segment attention layer is employed on top of the LSTM. This is the main model of this paper.

Note that C-GCN proposed a path-centric pruning strategy to empirically remove irrelevant content [Zhang *et al.*, 2018]. To fairly evaluate models, we also implement PA-LSTM augmented with the shortened sentences used in C-GCN, namely PA-LSTM+D. In a similar way, we can get the SA-LSTM+D.

4.4 Results

Experimental results are shown in Table 1. From the results, we can observe that: (1) With the same pruned input, our model outperforms state-of-the-art method with a relative improvement of 1.2%, which indicates that linear-chain CRF can work well as a attention mechanism for relation extraction. (2) SA-LSTM achieves higher precision and recall than PA-LSTM, which shows that modeling the dependencies between adjacent words can improve the model performance. (3) Given more precise shortened sentences, our SA-LSTM model still significantly outperforms the PA-LSTM, indicating that our proposed model can consistently benefits from the more precise input. (4) Comparing SA-LSTM+D with C-GCN, we can see that the gain mainly comes from improved recall. We hypothesize that this is because the C-GCN may suffer from the parser errors by modeling the tree structure directly while SA-LSTM+D just use the dependency tree to remove irrelevant words.

In addition, though pattern-based method also uses phrase patterns to capture the relation expressions which is similar to our motivation, SA-LSTM outperforms Pattern significantly. It shows that human-designed features are very limited as compared to neural models.

System	P	R	F_1
Pattern† [Angeli *et al.*, 2015]	**85.3**	23.4	36.8
LR† [Zhang *et al.*, 2017]	72.0	47.8	57.5
CNN-PE‡ [Zeng *et al.*, 2014]	68.2	55.4	61.1
PCNN‡ [Zeng *et al.*, 2015]	67.4	57.3	62.0
SDP-LSTM† [Xu *et al.*, 2015]	66.3	52.7	58.7
Tree-LSTM† [Tai *et al.*, 2015]	66.0	59.2	62.4
PA-LSTM† [Zhang *et al.*, 2017]	65.7	64.5	65.1
PA-LSTM+D‡	67.2	65.0	66.0
C-GCN† [Zhang *et al.*, 2018]	69.9	63.3	66.4
SA-LSTM	68.1	65.7*	66.9*
SA-LSTM+D	69.0	**66.2***	**67.6***

Table 1: Results on TACRED test set. bold marks highest number among all models. † marks results reported in [Zhang *et al.*, 2017] and [Zhang *et al.*, 2018]; ‡ marks results produced with our implementation. $*$ marks statistically significant improvements over C-GCN with $p < 0.01$ under a bootstrap test.

Model	Dev F_1
Best SA-LSTM	67.8
– Position embedding	64.5
– Transition regularizer	66.7
– Sparse regularizer	67.4
– Segment attention	57.1
– Pre-trained embeddings	65.3
– BiLSTM Layer	58.4

Table 2: An ablation study of the best SA-LSTM model on TACRED dev set. Scores are median of 5 models.

5 Analyses

5.1 Ablation Study

To study the contribution of each component in the SA-LSTM model, we run an ablation study on the TACRED dev set (see also Table 2). From these ablations, we find that: (1) The entire segment attention contributes about 10.7% F_1 score. (2) When we remove the position embedding and only use word embedding as input, the score drops by 3.3%, which indicates that it is important to let segment attention aware of position information. (3) Removing the transition regularizer hurts the result by 1.1% F_1 score. The performance also slightly degrades without sparse regularizer. Intuitively, segment attention can naturally recall more instances since the model captures a sequence of words rather than individual words. However, high recall also results in low precision, so these two regularizers can help model balance between precision and recall. (4) Segment attention usually performs better when coupled with BiLSTM since it is easier to model the interactions if contextual information is encoded in hidden representation.

5.2 Case Study

We compare our method with PA-LSTM on some cases, as shown in Table 3. As demonstrated by the first example, PA-

	Example	Predicted relation	True relation
PA-LSTM	SUBJ-PER SUBJ-PER, the son of Israel's first astronaut, OBJ-PER OBJ-PER, died in his home yesterday.	children	parents
SA-LSTM	SUBJ-PER SUBJ-PER, the son of Israel's first astronaut, OBJ-PER OBJ-PER, died in his home yesterday.	parents	
PA-LSTM	Prosecutors had accused SUBJ-PER, 22, then a student at OBJ-ORG OBJ-ORG, and her boyfriend Raffaele.	employee of	schools attended
SA-LSTM	Prosecutors had accused SUBJ-PER, 22, then a student at OBJ-ORG OBJ-ORG, and her boyfriend Raffaele.	schools attended	

Table 3: Output of SA-LSTM and PA-LSTM on samples from the TACRED test set, with words highlighted according to the attention weights produced by SA-LSTM and PA-LSTM. The third column for each example is the predicted result of corresponding model and the forth column is the gold standard.

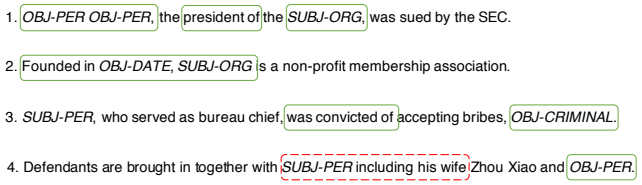

Figure 3: Visualization of the extracted relational expressions using Viterbi decoding algorithm. Green box indicates the model attend to the true relational expressions as desired while the extracted wrong segment is highlighted with red dotted box.

LSTM only focuses on the individual word "son" and cannot capture the full relational expression "the son of", so it is difficult for PA-LSTM to distinguish between relation *children* and *parents*. Thanks to the segment attention, our SA-LSTM model can successfully attend to the desired segment and make a correct decision.

In the second example, our proposed model successfully detects the relation phrase "*a student at*" while PA-LSTM only attend to the single word "*at*", lose the information that the subject entity is a student. Hence, it is not surprising that PA-LSTM wrongly marks this instance as relation *employee_of*. From these examples, we can observe that the proposed model is capable of capturing shallow structural information so as to perform relation extraction.

To give people some intuitive sense about how our models perform, we sample out some instances and use Viterbi decoding algorithm to extract the relation expressions explicitly (Figure 3). As we can see from the first two examples, the segment attention precisely detects the subject entity, object entity and the relation phrase between them. The third example shows that the model can successfully attend to the desired segment even if it is not adjacent to two target entities. Sometimes, our model also fails to attend to the right span as the forth example shows. Our segment attention identifies "*SUBJ-PER including his wife*" as a relational expression because "his wife" is a highly confusing trigger phrase and close to the subject entity, thus the model tends to capture them as a whole while no relation hold between the entity pair.

5.3 Error Analysis

Although the proposed method outperforms the state-of-the-art systems, we also observe several failure cases. The following is a typical example of a wrongly classified sentence: "*the SUBJ-ORG's annual conference in OBJ-LOC OBJ-LOC*". This sentence is wrongly classified as belonging to the *no relation* category, while the ground-truth label is *stateorprovince_of_headquarters*. The phrase "*annual conference in*" does not appear in the training data, and moreover is used metaphorically, making it difficult for the model to recognize the semantic connection.

Another common issue is that there are multiple relations in a sentence, such as the following ones: "*SUBJ-PER, the only child of OBJ-PER and his wife, Rosaile*". The model fails to attend to the right span because two relations hold simultaneously and their relational expressions are close to each other, so the attention mechanism tends to assign similar weights to the two phrases when extract the relation between *SUBJ-PER* and *OBJ-PER*. It would be interesting to see if designing more precise path pruning strategy can improve the performance since reducing the noise from input would hopefully further alleviate the burden of attention layer.

6 Conclusion

In this paper, we propose a novel model that learns the latent relational expressions based on the segment attention layer for relation extraction. By incorporating a linear-chain CRF into the attention layer, our model is capable of capturing the dependencies between target entities and their relations. Experiments on standard TACRED dataset show that our proposed model outperforms a strong feature-based classifier and all baseline neural models. We further compare the visualized attention of our model with the baseline model to show how segment attention layer affects the model. In the future, we will conduct research on how to design more sophisticated attention mechanism to address some of the existing challenges in relation extraction, such as multiple relations per sentence. The source code of this paper can be obtained from https://github.com/yubowen-ph/segment.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This research is supported by the National Key Research and Development Program of China (grant No.2016YFB0801003) and the Strategic Priority Research Program of Chinese Academy of Sciences (grant No.XDC02040400).

References

[Angeli *et al.*, 2015] Gabor Angeli, Victor Zhong, Danqi Chen, Arun Tejasvi Chaganty, Jason Bolton, Melvin Jose Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D Manning. Bootstrapped self training for knowledge base population. In *TAC*, 2015.

[Du *et al.*, 2018] Jinhua Du, Jingguang Han, Andy Way, and Dadong Wan. Multi-level structured self-attentions for distantly supervised relation extraction. *arXiv preprint arXiv:1809.00699*, 2018.

[Graves *et al.*, 2013] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[Kim *et al.*, 2017] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. Structured attention networks. *arXiv preprint arXiv:1702.00887*, 2017.

[Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[Lee *et al.*, 2019] Joohong Lee, Sangwoo Seo, and Yong Suk Choi. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. *arXiv preprint arXiv:1901.08163*, 2019.

[Liu *et al.*, 2015] Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. A dependency-based neural network for relation classification. *arXiv preprint arXiv:1507.04646*, 2015.

[Liu *et al.*, 2018] Tianyi Liu, Xinsong Zhang, Wanhao Zhou, and Weijia Jia. Neural relation extraction via inner-sentence noise reduction and transfer learning. *arXiv preprint arXiv:1808.06738*, 2018.

[Manning *et al.*, 2014] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[Rink and Harabagiu, 2010] Bryan Rink and Sanda Harabagiu. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 256–259. Association for Computational Linguistics, 2010.

[Santos *et al.*, 2015] Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*, 2015.

[Tai *et al.*, 2015] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[Turian *et al.*, 2010] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[Wang and Lu, 2018] Bailin Wang and Wei Lu. Learning latent opinions for aspect-level sentiment classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Wang *et al.*, 2016] Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. 2016.

[Xiao and Liu, 2016] Minguang Xiao and Cong Liu. Semantic relation classification via hierarchical recurrent neural network with attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1254–1263, 2016.

[Xu *et al.*, 2015] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1785–1794, 2015.

[Zeng *et al.*, 2014] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation classification via convolutional deep neural network. 2014.

[Zeng *et al.*, 2015] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, 2015.

[Zhang *et al.*, 2017] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, 2017.

[Zhang *et al.*, 2018] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*, 2018.

[Zhou *et al.*, 2016] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212, 2016.