



A Boundary Determined Neural Model For Relation Extraction

R.X. Tang, Y.B. Qin, R.Z. Huang, H. Li, Y.P. Chen

Ruixue Tang

1. College of Computer Science and Technology, Guizhou University
2. School of Information, Guizhou University of Finance and Economics
Guiyang 550025, China
gs.rxtang19@gzu.edu.cn

Yongbin Qin*

College of Computer Science and Technology, Guizhou University
Guiyang 550025, China
*Corresponding author: ybqin@foxmail.com

Ruizhang Huang

College of Computer Science and Technology, Guizhou University
Guiyang 550025, China
rzhuang@gzu.edu.cn

Hao Li

College of Computer Science and Technology, Guizhou University
Guiyang 550025, China
haolilhhh@foxmail.com

Yanping Chen

College of Computer Science and Technology, Guizhou University
Guiyang 550025, China
ypench@gmail.com

Abstract

Existing models extract entity relations only after two entity spans have been precisely extracted that influenced the performance of relation extraction. Compared with recognizing entity spans, because the boundary has a small granularity and a less ambiguity, it can be detected precisely and incorporated to learn better representation. Motivated by the strengths of boundary, we propose a boundary determined neural (BDN) model, which leverages boundaries as task-related cues to predict the relation labels. Our model can predict high-quality relation instance via the pairs of boundaries, which can relieve error propagation problem. Moreover, our model fuses with boundary-relevant information encoding to represent distributed representation to improve the ability of capturing semantic and dependency information, which can increase the discriminability of neural network. Experiments show that our model achieves state-of-the-art performances on ACE05 corpus.

Keywords: relation extraction, error propagation, boundary.

1 Introduction

Relation extraction aims to identify semantic relations between entities. This basic task in information extraction facilitates many more complex tasks, such as question answering [18], machine translation [22], emotion detection [20] and knowledge base construction [23]. Take a sentence from the ACE05 corpus for example (Figure 1). The entity of Person (PER) "thousands of pro-life crusaders" is associated with the geo-political entity (GPE) "Washington D.C." by a physical relation (PHY).

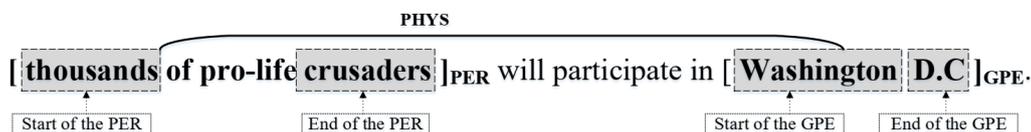


Figure 1: An example from ACE05 corpus

The previous relation extraction models can be roughly divided into two categories: pipeline models [42] and joint models [15, 16, 38]. The pipeline models extract relations in two steps, namely, entity recognition [5, 6] and relation classification [7, 43]. Ignoring the relevance between the two steps, the pipeline models tend to have the problem of error propagation. That is why the joint models came into being. The joint models extract entities and relations, and combine them into a unified framework [29, 30]. The relations are extracted based on the recognition of entity spans. Any error of entity recognition would spread to relation extraction. Therefore, the performance of relation extraction depends heavily on the effect of entity recognition.

Every entity span has a start boundary and an end boundary. For example, "Washington" is the start boundary of the entity "Washington D.C." in Figure 1. There are three advantages of boundary recognition over entity span recognition. First, the boundaries can be clearly encoded, due to their small granularity. Second, it is easy to automatically recognize boundaries, because they are dependent on local features. Third, the boundary-relevant information facilitates the learning of representation. For these advantages, the boundaries can provide a powerful support to relation extraction.

In many applications, researchers only care about the presence/absence of relations in raw texts. For instance, the protein interactions extracted from biomedical literature are important to new medical research. In public opinion analysis, it is more significant to find specific public opinions than to identify the expresser. Thus, a distinctive way of relation extraction is to extract semantic relations from sentences without needing the accurate measurement of entity spans.

For relation extraction, this paper combines boundary detection model with relation extraction model into a boundary determined neural (BDN) model. Specifically, a sequence labeling model was applied to identify boundaries. Based on the detected boundaries, the relation labels were classified with boundary pairs and relevant information. There are two prominent advantages of our model: (1) The model can generate many high-quality candidate boundaries, eliminating the need to import inaccurate entity spans into the relation extraction model; (2) By learning different representations, the model improves the capture of semantic and dependency information, which in turn improves the relation prediction.

This paper mainly makes two contributions: (1) Relation extraction was solved from a fresh perspective. Instead of entity spans, boundaries were selected as task-related cues to classify the relations. (2) The BDN model was developed to extract relations, and proved to achieve state-of-the-art results on ACE05 corpus.

2 Literature Review

Relation extraction has received growing interest in recent years. As mentioned before, pipeline models and joint models are the two major types of tools to extract relations.

Pipeline models treat entity recognition and relation extraction as two separate tasks. By pipeline models, named entities are recognized before the extraction of relations. Named entity recognition

(NER) [24, 25, 26] is usually treated as a sequence model based on neural networks like bidirectional long short-term memory with a conditional random field (CRF) layer (Bi-LSTM-CRF) or Bi-LSTM with convolutional neural network (CNN) (Bi-LSTM-CNN). The output is a flattened labeling sequence. Li et al. [26] designed the Cross-BiLSTM, which interleaves the forward and backward hidden states between LSTM layers. Recently, boundary supervision [34, 44] is introduced to entity recognition.

Meanwhile, relation extraction is normally treated as a multi-label classification task. The task can be executed under a shallow architecture or a deep learning (DL) architecture. Shallow architectures like support vector machine (SVM) and maximum entropy are often used in conjunction with feature engineering or kernel methods. As for DL architectures, the deep semantic features are widely extracted by recurrent neural networks (RNNs) [35, 36] and CNNs [17, 39, 40]. Tran et al. [36] preserved full relational information by integrating segment-level attention-based CNNs with dependency-based RNNs. Zhong and Chen [42] learned two encoders built on deep pretrained language models. Besides, the relation extraction models can be improved with texture features [1, 2, 8]. Chen et al. [8] grouped the features extracted from each sentence into multiple sets, and thus obtained the structural and semantic information required for relation extraction. Nevertheless, all relation extraction models must recognize entities first. They may suffer heavily from error propagation, for the neglect of the interaction between entity recognition and relation extraction.

Unlike pipeline models, joint models fuses the information of entities with that of relations, and identify entities and their relations under a unified framework. These models either adopt the strategy of parameter sharing or the strategy of joint decoding [3, 41]. By parameter sharing, the model shares some input features or internal hidden states. Bekoulis et al. [3] proposed a neural model with the CRF layer for NER and the sigmoid loss for multiple head prediction. Yu et al. [41] decomposed relation extraction into two subtasks, namely, the extraction of head- and tail-entities, and the extraction of their relations; the two subtasks were deconstructed into several sequence labeling problems based on tagging scheme. However, the parameter sharing models cannot capture all the interactions between entities, owing to the limitations of independent decoders.

To utilize the interaction between decoders, some joint decoding algorithms were developed to decode entities and relations all at once [13, 45, 46]. For example, Zeng et al. [45] proposed an end-to-end neural model based on sequence-to-sequence (Seq2Seq) with copy mechanism. To improve the accuracy of entity recognition, Zeng et al. [46] created a multi-task framework which adds a sequence labeling task to Zeng et al. [45] encoder. Yet the framework requires complex tags or decoding process.

To prevent mistakes from inaccurate entity spans, our model utilizes boundaries to classify relation labels instead of entity spans. The boundary-relevant information was fused to the neural network to learn different representations, such that the semantic and dependency information can be captured to improve model performance.

3 Methodology

The proposed BDN model is illustrated in Figure 2. The model obviously consists of two parts: *boundary detection* and *relation extraction*. The former part executes the sequence labeling task of generating boundary candidates from a sentence. The boundary candidates are employed as cues for relation extraction. The latter part takes boundary pairs as input to forecast relation types. This part is composed of a feature combination layer, an embedding layer, a convolution layer, a pooling layer, and a fully-connected layer (output layer).

3.1 Boundary detection

For a given sentence consisting of n tokens $T = \{t_1, t_2, \dots, t_n\}$, the word embedding of the i -th token t_i can be expressed as:

$$x_i = Emb(t_i). \quad (1)$$

where, Emb is the embedding operation. Here, the word embedding is initialized with a pretrained bidirectional encoder representations from transformers (BERT) mode [14]. Then, $x = \{x_1, x_2, \dots, x_n\}$

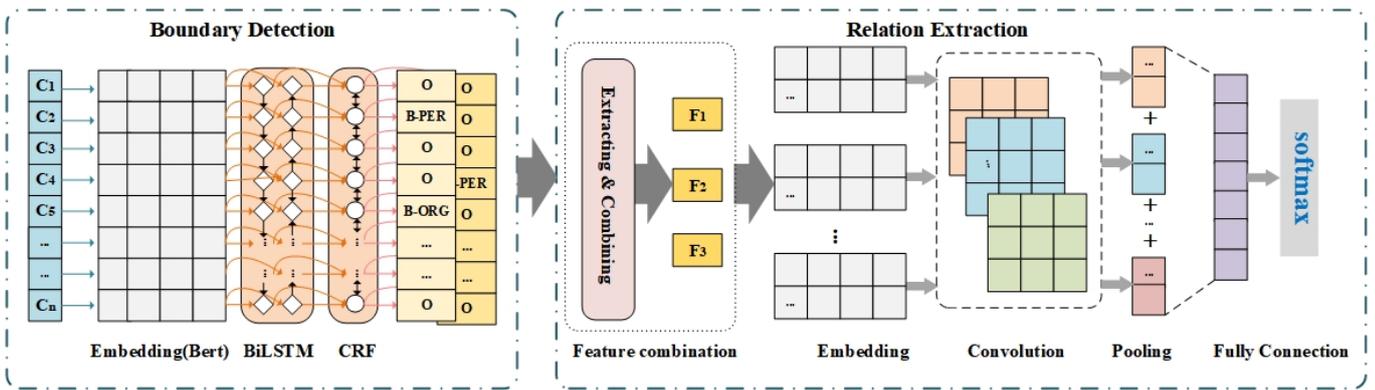


Figure 2: Framework of BDN model

is fed into a Bi-LSTM layer [19] to learn hidden states.

Each LSTM can be formalized as a function $f_{\theta}(x_{1:n}, i)$ with parameter set θ , where f is a nonlinear function (e.g., logistic sigmoid function or hyperbolic tangent function). The network can only capture important features from the input. To acquire more features, a Bi-LSTM can be applied to traverse the input data twice. The Bi-LSTM encompasses a forward LSTM and a backward LSTM f_L and f_R , with parameter sets of θ_L and θ_R , respectively. Taking token sequence $x_{1:n}$ and index i as input, the corresponding vector $\mathbf{v}_i = f_{\theta}(x_{1:n}, i)$ can be formalized as:

$$f_{\theta}(x_{1:n}, i) = h_{L,i} \circ h_{R,i}. \tag{2}$$

$$h_{L,i} = f_L(x_1, x_2, \dots, x_i). \tag{3}$$

$$h_{R,i} = f_R(x_n, x_{n-1}, \dots, x_i). \tag{4}$$

where, \circ is vector concatenation; \mathbf{v}_i is the concatenation of forward hidden representation $h_{L,i}$ and backward hidden representation $h_{R,i}$ at in the i -th step. In formula (3), an LSTM is applied on the input sequence to learned from the past context $x_{1:i}$. In formula (4), the reverse form of the input sequence is fed into the LSTM to learn the future context $x_{n:i}$.

Rather than assign an entity class label to each token, this paper predicts the boundary labels in the task. Let t_1, t_2, \dots, t_n be a sentence with start and end boundaries. The tokens are labeled by the BO tagging scheme. The boundary tokens t_i are assigned with labels $B - \varepsilon$, where ε is the set of pre-defined entity types; the non-boundary tokens are assigned with labels O .

Then, the boundary labels are predicted by feeding the corresponding representation $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)$ into a CRF layer, which is good at learning the constraint relationship between labels. The layer then outputs the final prediction label sequence $y^e = (y_1^e, y_2^e, \dots, y_n^e)$ for the input T . The optimal sequence can be obtained by the Viterbi algorithm. Then, the output label can be expressed as:

$$y^e = \arg \max P(\mathbf{v}, y'). \tag{5}$$

where, $P(\mathbf{v}, y')$ is the likelihood function of training the tags sequence \mathbf{v} ; y' is the ground-truth label sequence.

To sum up, boundary detection is considered as a sequence-labeling task. The input $x_{1:n}$ is represented as a token sequence t_1, t_2, \dots, t_n and fed into a forward LSTM and a backward LSTM. The two vectors are added to the CRF layer to decode the optimal label sequence. Then, the start boundary and an end boundary of each entity are identified separately. Every detected end boundary is greedily matched with the detected start boundary. The boundaries can be assembled into entity spans following prior work [9].

3.2 Relation extraction

3.2.1 Feature combination layer

Deep neural networks can automatically learn abstract high-dimensional semantic information from raw inputs [10]. However, the learning is limited to the local information around each word in

the sentence. For instance, CNNs cannot encode semantic information out of the filter range.

In relation extraction, a sentence usually contains several relation instances. The semantic dependencies between these instances cannot be fully encoded by deep neural networks. Inspired by Chen et al. [11, 12], our model fuses features based on the boundary-relevant information, so as to enhance the representation and capture the semantic dependencies in a sentence. Instead of processing each pair of entity spans, our model leverages each pair of boundaries and its relevant features as input.

For each input sentence and its boundary label sequence, our model processes every pair of candidate boundaries independently. Because relations are symmetric in general, every pair of candidate boundaries $\langle t_i, t_j \rangle$ can generate two relation instances: $\langle t_i, t_j, r \rangle$ and $\langle t_j, t_i, r \rangle$, where $r \in R$ and R with being the set of the relation labels, including a null label of the absence of relation between a boundary pair.

The features of each relation instance can be denoted as follows:

$$F_1 = \{z | z = TypeOf(t_i) _ TypeOf(t_j)\}. \quad (6)$$

$$F_2 = \{z | z = T _ PositionOf(t_i) _ PositionOf(t_j)\}. \quad (7)$$

$$F_3 = \{z | z = NGram(t_i, t_j)\}. \quad (8)$$

where, $_$ is concatenation; $TypeOf(t_i)$ is an indicator of boundary type (e.g., person (PER), and organization (ORG)); $PositionOf(t_i)$ maps position of t_i to an n-dimensional vector about the relative distance of the current word to t_i .

F_1 concatenates two boundary labels into a feature, and thus captures the semantic and dependency information of a relation instance. For example, in the *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*, a "Gen-Affiliation" is defined as the relationship between a facility and geo-political boundary. F_2 concatenates the input sentence and two positions into a feature, providing information for relation extraction. Each sentence can be divided by two boundaries into three parts: $[t_1, t_{i-1}]$, $[t_i, t_{j-1}]$ and $[t_j, t_n]$. F_3 concatenates three features, i.e., different representations from different channels, into a multichannel feature.

To sum up, the feature combination layer extracts the features from the input sentence, and then combines them into the final feature. These features are designed to capture the semantic dependencies in a sentence and fed into the next layer. They can be formalized as follow.

$$F = \cup_{i=1}^3 F_i = F_1 \cup F_2 \cup_{j=1}^3 F_3^j. \quad (9)$$

3.2.2 Embedding layer

For each relation instance, all the features are allocated to g groups, and each group is embedded into a vector. An independent lookup table is adopted to transform the groups into vector representations. Let m be the dimensionality of the embedded words. The embedding operation can be defined as:

$$[x_1^g, x_2^g, \dots, x_h^g] = Emb(F_g). \quad (10)$$

The output is expressed as a matrix $\mathbf{x}^g \in R^{h \times m}$. After that, each group is separately convoluted.

3.2.3 Convolution layer

The convolution layer transforms the representations into abstract features, using a convolutional filter w with a window size l . A filter can be viewed as a weight matrix $w = [w_1, w_2, \dots, w_l] (w_l \in R^{l \times k})$. Then, the convolutional operation can be described as:

$$X_i^g = f \left(w x_{i:i+k}^g + b \right) (1 \leq i + k \leq h). \quad (11)$$

where, f is an activation function (e.g., rectified linear unit (ReLU)); b is the bias. Convolution filters generally adopt a fixed window size, and thus unable to learn variable features flexibly. This process can then be replicated for various filters with different window sizes to produce a feature map:

$$[X_1^g, X_2^g, \dots, X_{h-k+1}^g] = Conv(\mathbf{x}^g). \quad (12)$$

The output of the convolution layer can be referred to as $\mathbf{X}^g \in R^{(h-k+1) \times h}$, which represents a high-dimensional abstract representation.

3.2.4 Pooling layer

Following the convolution layer, a pooling layer is arranged to determine the most useful feature in each feature representation. The max pooling over time on \mathbf{X}^g can be defined as:

$$\hat{X}^g = \max \{ X_{1:h-k+1}^g \}. \quad (13)$$

The pooling scores for every filter are concatenated into a feature representation:

$$[\hat{X}_1^g, \hat{X}_2^g, \dots, \hat{X}_s^g] = \text{Pooling}(\mathbf{X}^g). \quad (14)$$

The output of the pooling layer can be referred to as $\hat{\mathbf{X}}^g \in R^s$, where s is the number of filters in the model and \hat{X}_s is the max pooling score of the s -th filter.

3.2.5 Fully connected layer

A relation instance inputted to the model can be encoded as:

$$\hat{\mathbf{X}} = \oplus_{i=1}^g \text{Pooling}(\text{Conv}(\text{Emb}(F^g))) = \oplus_{i=1}^g \hat{\mathbf{X}}^g. \quad (15)$$

where, \oplus is the concatenation operation. The dropout strategy is adopted to prevent overfitting. The output can be described as:

$$\mathbf{o} = f'(\hat{\mathbf{X}} \cdot c) + b'. \quad (16)$$

where, \cdot is element-wise multiplication operation; c is Bernoulli random variables; b' is the bias; $f' \in R^{r \times s'}$ is the weight, with being s' equal to s plus the dimension of the features, and r being the relation type.

Finally, a softmax function is adopted to predict the probabilistic distribution for all types. Parameter θ is calculated by stochastic gradient descent (SGD), and updated by error backpropagation algorithm:

$$P(i|c, \theta) = \frac{e^{\theta_i}}{\sum_{i=1}^r e^{\theta_i}}. \quad (17)$$

4 Experiment

4.1 Datasets

Our model was evaluated on ACE 2005 corpus, using both English corpus and Chinese corpus. ACE05 defines 7 types of entities, namely, Facility (FAC), Geo-Political Entities (GPE), Location (LOC), Organization (ORG), Person (PER), Vehicle (VEH), and Weapon (WEA), and 6 types of relations between entities, namely, Artifact (ART), Gen-Affiliation (GEN-AFF), Org-Affiliation (ORG-AFF), Part-Whole (PARTWHOLE), Person-Social (PER-SOC), and Physical (PHYS).

The files from ACE05 English corpus were preprocessed and segmented by the procedure mentioned in Li's work [27]. Those from ACE05 Chinese corpus were split into a training set, a verification set, and a test set by the ratio of 6:2:2. Table 1 shows the data of each corpus. Every file was segmented into sentences. A relation instance was deemed as negative, if there is no relation between two boundaries. Since boundaries are ordered in a sentence, if a sentence has three boundaries (A, B, C), then six pairs could be generated: [A, B], [A, C], [B, C], [B, A], [C, A], and [C, B]. Since the nested entities may have the same boundaries, three pairs could be added: [A, A], [B, B] and [C, C].

Table 1: Data of each corpus

Dataset	Number of relation types	Number of entity types	Number of files	Number of sentences		
				Train	Verification	Test
ACE English	7	6	511	7,273	1,765	1,535
ACE Chinese	7	6	628	5,546	1,858	1,850

4.2 Parameter settings

The BERT base model (uncased) was taken as the base encoders for ACE corpus. The dimension of word embedding was set to 768 in BERT. The embedded data were further mapped into a 128×2 -dimensional vector in Bi-LSTM. The filter windows of 3, 4, 5, 6, and 7 were adopted with 128 feature maps. The dimensionality of word and feature were configured as 100 and 50, respectively. Learning rate, dropout probability, and batch size were set as 0.9, 0.5 and 64, respectively.

The performance of our model was measured by Precision (P), Recall (R) and F1 score (F1) to evaluate the performance [4]. The strict relation evaluation **Rel+** was adopted, where a predicted relation is considered correct if both entity span and relation type are correct. In addition, boundary evaluation is denoted as **Rel**, where the boundaries must be correct.

4.3 Baselines

Our model was compared with the following existing models:

Beam search [27], an incremental method, which adopts segment-based decoder based on semi-Markov chain and global features;

SPTree [31], which detects entities with bidirectional sequential LSTM, and decodes relations with tree-structured LSTM-RNN;

Global normalization [47], a table-filling method, which integrates syntactic information to facilitate global learning;

Att-BiLSTM [21], a BiLSTM model with attention, which does not adopt any dependency tree;

Minimum risk training [32], which recognizes entities with an RNN, extracts relations with a CNN, and trains the model by a minimum risk strategy;

Graph convolutional network (GNN) [33], which detect all entity spans, and then designs an entity-relation bipartite graph to support relation extraction;

Multi-turn question answering (QA) [28], which characterizes each entity type and relation type via a question answering template, and then extracts relations by a standard machine reading comprehension (MRC) framework;

Table-sequence encoder [38], which introduces table-sequence encoders containing two different encoders;

Span-level NER decoder [37], which predicts the relations in RE decoder with a span-level NER;

Two independent encoders [42], as a pipeline approach, which are built on span-level representations and contextual representations, respectively.

Several contrastive experiments were also carried out, including "Ours w/o AF", which uses a standard multi-class classifier to detect relation, and "Ours w/o CF", which utilizes the most useful feature: the type information.

4.4 Comparing with existing models

This subsection compares our model with the published models on the ACE English corpus. As shown in Table 2, our model outperformed the previous models in relation extraction: the advantage in F1 score was at least 1.8% on boundary evaluation (Rel) and 0.6% on strict evaluation (Rel+), when recognizing entity spans. It can be concluded that our model is exceptionally powerful in relation extraction. The main reason is that our model uses boundary-related information to learn better representations. These data, which contain priori knowledge and experience, can be transformed into the high-dimensional abstract representations, making it easier to capture semantic and dependency information. That is how our model achieved excellent performance of relation extraction.

Table 2: Comparing with existing models

Model	Entity			Rel			Rel+		
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Li and Ji [27]	85.2	76.9	80.8	68.9	41.9	52.1	65.4	39.8	49.5
Miwa and Bansal [31]	82.9	83.9	83.4	-	-	-	57.2	54.0	55.6
Zhang et al. [47]	-	-	83.6	-	-	-	-	-	57.5
Katiyar and Cardie [21]	84.0	81.3	82.6	57.9	54.0	55.9	55.5	51.8	53.6
Sun et al. [32]	83.9	83.2	83.6	64.9	55.1	59.6	-	-	-
Sun et al. [33]	86.1	82.4	84.2	68.1	52.3	59.1	-	-	-
Li et al. [28]	84.7	84.9	84.8	-	-	-	64.8	56.2	60.2
Wang and Lu [38]	-	-	89.5	-	-	67.6	-	-	64.3
Taillé et al. [37]	-	-	87.4	-	-	61.2	-	-	64.4
Zhong and Chen [42]	-	-	88.7	-	-	67.0	-	-	62.2
Ours:span	86.9	86.7	86.8	75.1	66.5	69.4	77.6	62.7	65.0
Ours:boundary	92.8	90.4	91.6	75.3	76.7	75.3	-	-	-

Table 3: Results on ACE05

Relation Type	ACE English			ACE Chinese		
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
PART-WHOLE	72.02	85.21	78.06	82.58	73.50	77.78
ORG-AFF	91.12	84.29	87.57	99.39	80.20	88.77
GEN-AFF	82.57	69.23	75.31	93.65	85.51	89.39
PHYS	67.16	58.44	62.50	96.58	92.97	94.74
PER-SOC	87.80	86.40	87.10	91.30	86.90	89.05
ART	50.86	76.72	61.17	97.27	88.56	92.71
Total	75.26	76.72	75.29	93.46	84.60	88.74

Our model also brought the highest performance, improving the F1 score by +7.7% on boundary evaluation (Rel), when predicting relation types with boundaries. The improvement is attributable to two facts: First, boundary detection performs better than entity span recognition. Our model can generate high-quality candidate boundaries to be fed into the next layer. This solves the mistakes from inaccurate entity spans, and alleviates the error propagation problem. Second, our model can capture more features between boundary pairs, making the network more discriminable. The edge of our model in F1 score over other models clearly shows its superiority.

4.5 Results on ACE05

Table 3 compares the results of our model on English corpus and Chinese corpus against those of baselines.

Our model achieved better performance on Chinese corpus than on English corpus. The performance gap was 13.5% in F1 score. The reason may be the good performance on Chinese boundary detection. Besides, Chinese and English sentences differ greatly in syntactic structure. English sentences are generally longer and stricter in word order, which is the key in sentence interpretation. By contrast, Chinese sentences are composed of informative characters, which reflect the syntactic and semantic information of the sentence.

4.6 Analysis of boundary detection

To disclose the influence of boundaries, experiments were conducted on boundary detection of English corpus and Chinese corpus. Table 4 compares the detection results of three models on start boundaries, end boundaries and entity spans. Note that the BERT was adopted to encode word semantics, BiLSTM to capture the semantic and grammatical dependencies in a sentence representation, and CRF to support detection with artificial features.

As shown in Table 4, the F1 scores of boundary detection were 4-6% higher than that of entity recognition. The possible reasons are as follows: the boundaries are clear and of small granularity, leaving no space for ambiguity; the boundary detection is more dependent on local features, and easier to be recognized automatically. It was observed that BERT-BiLSTM achieved 2% higher F1 score than the best result of BERT-CrossBiLSTM [26]. This fully demonstrates the superiority of our boundary detection model.

Table 4: Influence of boundary detection

Dataset	Model	Start Boundary			End Boundary			Spans		
		P(%)	R(%)	F(%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
	random-Bilstm	83.11	68.85	75.31	83.77	66.74	74.29	-	-	-
ACE English	Bert-CrossBilstm [26]	87.31	91.39	89.30	86.70	88.81	87.74	-	-	-
	Bert-Bilstm	92.80	90.44	91.60	92.39	89.43	90.88	86.92	86.74	86.83
	random-Bilstm	85.85	76.92	81.14	87.59	80.97	84.15	-	-	-
ACE Chinese	Bert-CrossBilstm [26]	93.04	94.12	93.58	94.35	93.63	93.99	-	-	-
	Bert-Bilstm	94.72	95.26	94.99	95.91	95.29	95.60	92.21	86.56	89.30

Table 5: Influence of feature combination

Model	ACE English			ACE Chinese		
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
S	50.82	75.94	59.64	76.41	46.36	57.04
F_1	73.17	69.42	71.13	93.48	71.29	80.31
F_2	67.73	74.92	69.88	91.40	78.87	84.53
F_3	58.21	79.06	65.84	72.29	74.14	72.81
$F_1 \cup F_2$	67.52	77.80	71.05	92.63	83.53	87.70
$F_1 \cup F_3$	82.90	67.65	74.19	83.93	84.12	83.59
$F_2 \cup F_3$	65.89	78.18	70.54	91.49	78.71	84.35
$F_1 \cup F_2 \cup F_3$	75.26	76.72	75.29	93.46	84.60	88.74

4.7 Analysis of feature combination

Table 5 lists the experimental results on the influence of boundary-relevant features on model performance. The F1 scores on English corpus and Chinese corpus stood at 59.64% and 57.04%, respectively. This is because the entity can be any word, and the sentence fails to provide sufficient information. Moreover, model F_1 increased the F1 score by +1.3% and +5.3% from the levels of models F_2 and F_3 , respectively. This means the boundary-relevant features provide critical information to support relation extraction, and type information is highly useful in improving the relation performance. In addition, models F_1 , F_2 and F_3 suffered a significant performance drop compared with the combination of all features, indicating the insufficiency of solely relying on a single feature. It can be concluded that the improvement over other models comes from the abundance of features.

4.8 Analysis of different boundary

To better understand the BDN performance, the influence of the start and end boundaries was evaluated separately on relation extraction. As shown in Figure 3, the performance of start boundary reached an F1 score of 75.29% and 88.74% on English corpus and Chinese corpus, respectively. The results were 0.6% and 1.5% better than the F1 scores with end boundary on the two corpuses, respectively. During the ablation experiment, the performance dropped deeply, but the performance of the start boundary approached that of the end boundary. The result indicates that the two boundaries have no significant difference in the impact on relation extraction. Both start boundary and end boundary can flexibly support relation extraction in place of entities.

5 Conclusion

This paper presents a BDN model, which leverages the boundaries to predict relation labels. Our model combines sequence labeling model with classification model to accurately classify relation instances. Experimental results show that the BDN can effectively extract relations, and outperform start-of-the-art baselines on ACE05 corpus. Through extensive analyses, the authors fully demonstrated the superiority of our model in boundary detection, and validated the importance of using boundary pairs as input and learning different representations from boundary-relevant information. The future research will further reduce mistakes and make more meaningful comparisons in a unified end-to-end evaluation setting.

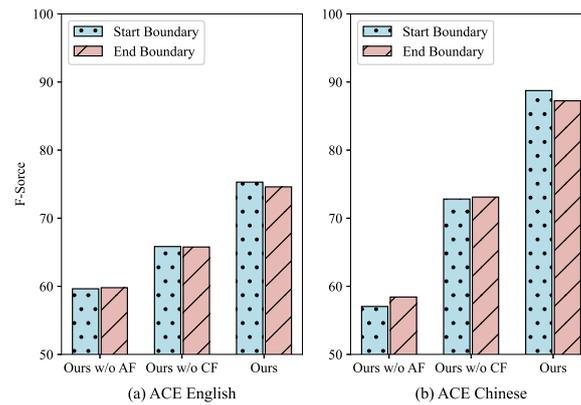


Figure 3: Influence of Different Boundary

Acknowledgements

This work is supported by the Joint Funds of the National Natural Science Foundation of China under Grant No. U1836205, the Major Research Program of National Natural Science Foundation of China under Grant No. 91746116, National Natural Science Foundation of China under Grant No. 62066007 and No. 62066008, the Major Special Science and Technology Projects of Guizhou Province under Grant No. [2017]3002 and the Key Projects of Science and Technology of Guizhou Province under Grant No. [2020]1Z055)

References

- [1] Beirami, B.A.; Mokhtarzade, M. (2020). Superpixel-based minimum noise fraction feature extraction for classification of hyperspectral images, *Traitement du Signal*, 37(5), 815-822, 2020.
- [2] Bhatele, K.R.; Bhadauria, S.S. (2020). Glioma Segmentation and Classification System Based on Proposed Texture Features Extraction Method and Hybrid Ensemble Learning, *Traitement du Signal*, 37(6), 989-1001, 2020.
- [3] Bekoulis, G.; Deleu, J.; Demeester, T.; Develder, C. (2018). Joint entity recognition and relation extraction as a multi-head selection problem, *Expert Systems with Applications*, 114, 34-45, 2018.
- [4] Bekoulis, G.; Deleu, J.; Demeester, T.; Develder, C. (2018). Adversarial training for multi-context joint entity and relation extraction, *arXiv preprint arXiv:1808.06876*, 2018.
- [5] Chiu, J.; Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns, (2016). *Transactions of the Association for Computational Linguistics*, 4, 357-370, 2016.
- [6] Cho, M.S.; Ha, J.H.; Park, C.Y.; Park, S.H. (2020). Combinatorial feature embedding based on cnn and lstm for biomedical named entity recognition, *Journal of Biomedical Informatics*, 103, 103381, 2020.
- [7] Chan, Y.S.; Roth, D. (2011). Exploiting syntactico-semantic structures for relation extraction, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.551-560, 2011.
- [8] Chen, Y.P.; Zheng, Q.H.; Chen, P. (2015). Feature assembly method for extracting relations in chinese. *Artificial Intelligence*, 228, 179-194, 2015.
- [9] Chen, Y.P.; Wu, Y.F.; Qin, Y.B.; Hu, Y.; Wang, Z.Y.; Huang, R.Z.; Cheng, Z.Y.; Chen, P. (2019). Recognizing nested named entity based on the neural network boundary assembling model, *IEEE Intelligent Systems*, 35(1), 74-81, 2019.

- [10] Chen, G.; Xu, B.; Lu, M.L.; Chen, N.S. (2018). Exploring blockchain technology and its potential applications for education, *Smart Learning Environments*, 5(1), 1-10, 2018.
- [11] Chen, Y.P.; Zheng, Q.H.; Chen, P. (2017). A set space model for feature calculus, *IEEE Intelligent Systems*, 32(5), 36-42, 2017.
- [12] Chen, Y.P.; Wang, G.R.; Zheng, Q.H.; Qin, Y.B.; Huang, R.Z.; Chen, P. (2019). A set space model to capture structural information of a sentence, *IEEE Access*, 7, 142515-142530, 2019.
- [13] Dai, D.; Xiao, X.Y.; Lyu, Y.J.; Dou, S.; She, Q.Q. Wang, H.F. (2019). Joint extraction of entities and overlapping relations using position-attentive sequence labeling, *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, pp.6300-6308, 2019.
- [14] Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805*, 2018.
- [15] Fu, T.J.; Li, P.H.; Ma, W.Y. (2019). Graphrel: Modeling text as relational graphs for joint entity and relation extraction, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp.1409-1418, 2019.
- [16] Gupta, P.; Schütze, H.; Andrassy, B. (2016). Table filling multi-task recurrent neural network for joint entity and relation extraction, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp.2537-2547, 2016.
- [17] Gorur, K.; Bozkurt, M.R.; Bascil, M.S.; Temurtas, F. (2019). GKP Signal Processing Using Deep CNN and SVM for Tongue-Machine Interface, *Traitement du Signal*, 36(4), 319-329, 2019.
- [18] Hao, Y.C.; Zhang, Y.Z.; Liu, K.; He, S.Z.; Liu, Z.Y.; Wu, H.; Zhao, J. (2017). An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1, pp.221-231, 2017.
- [19] Hochreiter S.; Schmidhuber, J. (1997). Long short-term memory, *Neural computation*, 9(8), 1735-1780, 1997.
- [20] Kulkarni, P.; Rajesh, T.M. (2021). Video based sub-categorized facial emotion detection using LBP and edge computing, *Revue d'Intelligence Artificielle*, 35(1), 55-61, 2021.
- [21] Katiyar, A.; Cardie, K. (2017). Going out on a limb: Joint extraction of entity mentions and relations without dependency trees, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 1, pp.917-928, 2017.
- [22] Liu, Y.H.; Gu, J.T; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation, *Transactions of the Association for Computational Linguistics*, 8, 726-742, 2020.
- [23] Luan, Y.; He, L.H.; Ostendorf, M.; Hajishirzi H. (2018). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction, *arXiv preprint arXiv:1808.09602*, 2018.
- [24] Li, F.; Zhang, M.S.; Tian, B.; Chen, B.; Fu, G.H.; Ji, D.H. (2018). Recognizing irregular entities in biomedical text via deep neural networks, *Pattern Recognition Letters*, 105, 105-113, 2018.
- [25] Lin, H.Y.; Lu, Y.J.; Han, X.P.; Sun, L. (2019). Sequence-to-nuggets: Nested entity mention detection via anchor-region networks, *arXiv preprint arXiv:1906.03783*, 2019.
- [26] Li, P.H.; Fu, T.J. Ma, W.Y. (2020). Why attention? analyze bilstm deficiency and its remedies in the case of ner, *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05), pp.8236-8244, 2020.

- [27] Li, Q.; Ji, H. (2014). Incremental joint extraction of entity mentions and relations, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 1, pp.402-412, 2014.
- [28] Li, X.Y.; Yin, F.; Sun, Z.J.; Li, X.Y.; Yuan, A. and Chai, D.; Zhou, M.X.; Li, J.W. (2019). Entity-relation extraction as multi-turn question answering, *arXiv preprint arXiv:1905.05529*, 2019.
- [29] Miwa, M.; Sasaki, Y. (2014). Modeling joint entity and relation extraction with table representation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp.1858-1869, 2014.
- [30] Meng, Z.; Tian, S.W.; Yu, L.; Lv, Y.L. (2020). Joint extraction of entities and relations based on character graph convolutional network and multi-head self-attention mechanism, *Journal of Experimental & Theoretical Artificial Intelligence*, 1-14, 2020.
- [31] Miwa, M.; Bansal, M. (2016). End-to-end relation extraction using lstms on sequences and tree structures, *arXiv preprint arXiv:1601.00770*, 2016.
- [32] Sun, C.Z.; Wu, Y.B.; Lan, M.; Sun, S.L.; Wang, W.T.; Lee, K.C.; Wu, K.W. (2018). Extracting entities and relations with joint minimum risk training, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp.2256-2265, 2018.
- [33] Sun, C.Z.; Gong, Y.Y.; Wu, Y.B.; Gong, M.; Jiang, D.X.; Lan, M.; Sun, S.L.; Duan, N. (2019). Joint type inference on entities and relations via graph convolutional networks, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp.1361-1370, 2019.
- [34] Tan, C.Q.; Qiu, W.; Chen, M.S.; Wang, R.; Huang, F. (2020). Boundary enhanced neural span classification for nested named entity recognition, *Proceedings of the AAAI Conference on Artificial Intelligence*, 34, pp.9016-9023, 2020.
- [35] Tripathi, A.; Jain, A.; Mishra, K.K.; Pandey, A.B.; Vashist, P.C. (2020). MCNN: A deep learning based rapid diagnosis method for COVID-19 from the X-ray images, *Revue d'Intelligence Artificielle*, 34(6), 673-682, 2020.
- [36] Tran, V.H.; Phi, V.T.; Shindo, H.; Matsumoto, Y.J. (2019). Relation classification using segment-level attention-based cnn and dependency-based rnn, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1, pp.2793-2798, 2019.
- [37] Taillé, B.; Guigue, Y.; Scoutheeten, G.; Gallinari, P. (2020). Let's stop incorrect comparisons in end-to-end relation extraction!, *arXiv preprint arXiv:2009.10684*, 2020.
- [38] Wang, J.; Lu, W. (2020). Two are better than one: Joint entity and relation extraction with table-sequence encoders, *arXiv preprint arXiv:2010.03851*, 2020.
- [39] Wang, L.L.; Cao, Z.; De Melo, G.; Liu, Z.Y. (2016). Relation classification via multi-level attention cnns. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 1, pp.1298-1307, 2016.
- [40] Yan, X.D.; Song, X.G. (2020). An image recognition algorithm for defect detection of underground pipelines based on convolutional neural network, *Traitement du Signal*, 37(1), 45-50, 2020.
- [41] Yu, B.W.; Zhang, Z.Y.; Shu, X.B.; Wang, Y.B.; Liu, T.W.; Wang, B.; Li, S.J. (2019). Joint extraction of entities and relations based on a novel decomposition strategy, *arXiv preprint arXiv:1909.04273*, 2019.
- [42] Zhong, Z.X.; Chen, D.Q. (2020). A frustratingly easy approach for joint entity and relation extraction, *arXiv preprint arXiv:2010.12812*, 2020.

- [43] Zeng, D.J.; Liu, K.; Lai, S.W.; Zhou, G.Y.; Zhao, J. (2014). Relation classification via convolutional deep neural network, *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, pp.2335-2344, 2014.
- [44] Zheng, C.M.; Cai, Y.; Xu, J.Y.; Leung, H.F.; Xu, G.D. (2019). A boundary-aware neural model for nested named entity recognition, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 357-366, 2019.
- [45] Zeng, X.R.;Zeng, D.J.; He, S.Z.; Liu, K.; Zhao, J. (2018). Extracting relational facts by an end-to-end neural model with copy mechanism, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 1, pp.506-514, 2018.
- [46] Zeng, D.J.; Zhang, H.R.; Liu,Q.Y. (2020). Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning, *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05), pp.9507-9514, 2020.
- [47] Zhang, M.S.; Zhang, Y.; Fu, G.H. (2017). End-to-end neural relation extraction with global optimization, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp.1730-1740, 2017.



Copyright ©2020 by the authors. Licensee Agora University, Oradea, Romania.

This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International License.

Journal's webpage: <http://univagora.ro/jour/index.php/ijccc/>



This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).

<https://publicationethics.org/members/international-journal-computers-communications-and-control>

Cite this paper as:

Tang, R.X.; Qin, Y.B.; Huang, R.Z.; Li, H.; Chen, Y.P. (2021). A Boundary Determined Neural Model For Relation Extraction, *International Journal of Computers Communications & Control*, 16(4), 4235, 2021.

<https://doi.org/10.15837/ijccc.2020.3.4235>