symmetry

MDPI

Article

Entity Relation Extraction Based on Entity Indicators

Yongbin Qin [1,†], Weizhe Yang [1,†], Kai Wang [1], Ruizhang Huang [1], Feng Tian [2], Shaolin Ao [1] and Yanping Chen [1,*]

[1] College of Computer Science and Technology, Guizhou University, Guiyang 550025, China; ybqin@gzu.edu.cn (Y.Q.); wzyang.shin@gmail.com (W.Y.); gs.kaiwang20@gzu.edu.cn (K.W.); rzhuang@gzu.edu.cn (R.H.); alyanyi@foxmail.com (S.A.)

[2] School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China; fengtian@mail.xjtu.edu.cn

* Correspondence: ypench1@gzu.edu.cn

† These authors contributed equally to this work.

Abstract: Relation extraction aims to extract semantic relationships between two specified named entities in a sentence. Because a sentence often contains several named entity pairs, a neural network is easily bewildered when learning a relation representation without position and semantic information about the considered entity pair. In this paper, instead of learning an abstract representation from raw inputs, task-related entity indicators are designed to enable a deep neural network to concentrate on the task-relevant information. By implanting entity indicators into a relation instance, the neural network is effective for encoding syntactic and semantic information about a relation instance. Organized, structured and unified entity indicators can make the similarity between sentences that possess the same or similar entity pair and the internal symmetry of one sentence more obviously. In the experiment, a systemic analysis was conducted to evaluate the impact of entity indicators on relation extraction. This method has achieved state-of-the-art performance, exceeding the compared methods by more than 3.7%, 5.0% and 11.2% in F1 score on the ACE Chinese corpus, ACE English corpus and Chinese literature text corpus, respectively.

Keywords: relation extraction; entity indicators; entity pair; neural networks

check for updates

Citation: Qin, Y.; Yang, W.; Wang, K.; Huang, R.; Tian, F.; Ao, S.; Chen, Y. Entity Relation Extraction Based on Entity Indicators. *Symmetry* **2021**, *13*, 539. https://doi.org/10.3390/sym13040539

Academic Editor: Vladik Kreinovich

Received: 10 March 2021
Accepted: 23 March 2021
Published: 25 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Relation extraction is one of the fundamental information extraction (IE) tasks that aims to identify the semantic relationship between two named entities in a sentence [1]. For example, given a sentence "Steve Jobs was the co-founder of Apple Inc.", an employee relation instance can be identified between "Steve Jobs" and "Apple Inc.". Finding the relation between two named entities has a wide range of applications, e.g., clinical decision support [2], drug discovery [3] and economic management [4]. This task is seen as foundational to support other natural language processing (NLP) tasks, such as knowledge graph construction [5], question answering [6] and natural language understanding [7]. Therefore, relation extraction has received extensive research attention [8]. Currently, the neural network is the most popular method to support relation extraction, where multilayer stacked architecture is adopted to support the designated feature transformation, e.g., convolutional neural network (CNN) [9], recurrent neural network (RNN) [10] and attention mechanism [11]. This approach has the advantage of extracting high-order abstract features from raw inputs, avoiding the effort required for the manual generation of designed features. The main problem for relation extraction is that a sentence often contains several named entities. Because relation types are asymmetrical, every entity pair in a sentence should be considered a possible relation instance. This consideration leads to a serious data imbalance problem. Furthermore, all entity pairs share the same context, weakening the discriminability of the features to predicate a relation instance. Therefore, obtaining entity position information is highly important for a neural network to concentrate on the considered entity pair.

In related works, several neural-network-based strategies have been proposed to encode the entity position information. Position embedding is a traditional method [12] in which for every word in a sentence, its coordinates relative to the two relation arguments (named entities) are embedded into a vector; it is concatenated with the word embedding to encode the position information. The multichannel method is another technique [13] to notice the entity pair in relation extraction. In this method, a sentence is divided into five channels. Each channel adopts an independent lookup table, enabling the neural network to learn different representations for the same word. The shortest dependency path between the named entities are also widely adopted for encoding position information of relation instances [14]. In other methods, Zhang et al. [15] used semantic information of patterns and flexible match method to constrain the structure of relation instances. Zheng et. al. [16] exploited the subsentences between two named entities.

A relation is mentioned in a sentence relevant to two named entities. The entity pair plays a central role in relation representations. Entity pairs have three characteristics. First, the structural information of a relation mention is determined by the entity pair. It precisely divides a sentence into five parts. Each part has specific expression patterns. Second, the semantic expression of a relation mention depends on the entity pair. Every word in a relation mention semantically depends on the named entities. Third, semantic information of the entity pair is important for relation extraction. For example, some relation types only appear in specific named entity types.

Motivated by these characteristics, in this paper, rather than directly implementing a neural network on a sentence, task-related entity indicators are designed for a neural network to capture the position information of named entities. This approach enables a deep neural network to concentrate on the task-relevant information and ignore the effect of the irrelevant named entities. This strategy has also been discussed by Soares et al. [17], Zhang et al. [18] and Zhong et al. [19]. In this paper, we extend this notion further by encoding positional, syntactic and semantic information into entity indicators, making a deep neural network concentrate on the task-relevant information. This method is also effective for capturing syntactic and semantic information of a relation instance. The contributions of this paper include the following:

1 Entity indicators are designed to support relation extraction. Several types of entity indicators are proposed in this paper. These indicators are effective for capturing the semantic and structural information of a relation instance.
2 The entity indicators are evaluated based on three public corpora, providing a systematic analysis of these indicators in supporting relation extraction. A performance comparison showed that our method considerably outperforms all compared works.

The rest of this paper is organized as follows. Section 2 discusses the related works on relation extraction. In Section 3, entity indicators are introduced. Section 4 evaluates the entity indicators based on three public corpora. The conclusion is given in Section 5.

2. Related Works

Most of the early research methods on relation extraction can be categorized into feature-based methods [20] and kernel-based methods [21]. In feature-based models, a shallow architecture (e.g., Support Vector Machine (SVM) [22] or Maximum Entropy (ME) [20]) is adopted to make a prediction based on categorical features. For example, Kambhatla et al. [20] combined lexical features, syntactic features and semantic features for relation extraction. Kia Dashtipour et al. [23] proposed a scalable system for Persian Named Entity Recognition (PNER) that was developed by combining Persian grammatical rule-based approach with SVM. Minard et al. [24] used external resources, e.g., stems of the words, and VerbNet's classes. Because structural information of relation instances is important to support relation extraction, Chen et al. [25] proposed a feature assembly method to generate combined features. In addition, Liu et al. [26] proposed a convolutional tree kernel-based method that incorporates semantic similarity measures. Panyam et al. [27] employed a graph kernel to encode edge labels and edge directions. The extendibility of

kernels is limited because manually designed distance functions (kernel) are used to compute the similarities between relation instances. Furthermore, generation of dependency trees heavily depends on external toolkits, which are also error-prone.

Because neural networks have the advantage of automatically extracting high-order representation from relation instances, they are widely used to support relation extraction. For example, Leng et al. [28] designed a deep architecture fusing both word level and sentence level information. Liu et al. [9] introduced an early model based on a neural network, where a convolutional neural network (CNN) was designed to support relation extraction. In relation extraction, the entity position is important information to point to the entity pair. Zeng et al. [29] utilized relative distance between the current word and the target entity pair to encode position features under a CNN model. To utilize contextual features between the named entities, Zeng et al. [12] proposed a piecewise convolutional neural network (PCNN). Li et al. [30] combined the PCNN and attention mechanism for relation extraction. Zhang et al. [31] applied CNN model to shape classification in image processing. Wang et al. [32] used a bidirectional shortest dependency path (Bi-SDP) attention mechanism to capture the dependency information of words. To capture semantic dependencies in a relation instance, a long short-term memory (LSTM) model with attention mechanism was also applied [33,34]. Based on a graph neural network, Zhao et al. [35] proposed an N-gram Graph LSTM (NGG LSTM) model to capture sentence structure information. Chen et al. [13] used a multichannel deep neural network to partition a whole sentence into five parts, enabling the neural network to learn different representations for the same word.

Instead of investigating position embedding, Zhang et al. [18] utilized entity indicators to identify the start and end of the entity pair under a recurrent neural network. In this approach, indicators are placed on both sides of the entity pair to point to entity positions. An attention-based bidirectional long short-term memory (Att-BLSTM) model was proposed for obtaining contextual semantic dependencies from long texts [33]. Entity indicators were also used as the position indicators in this method. Based on the pretrained language model BERT [36], Soares et al. [17] placed four entity indicators on both sides of two named entities. The outputs of these four entity marker representations are concatenated as the relation representation. Zhong et al. [19] used entity type information to mark the entity pair and to learn the relation representation. Huang et al. [37] proposed a knowledge graph enhanced transformer encoder architecture to handle the semantic gap problem between word embeddings and entity embeddings.

The study on named entity recognition (NER) is important for relation extraction. Due to relation extraction being based on the correct position of entity pair in the sentences, the study on multiple languages NER is becoming more and more popular. McDonough et al. [38] studied the specified scope geographic NER about modern french. Isozaki et al. [39] studied Japanese named entity recognition based on a rule generator and decision tree learning. Medical NER in Swedish and Spanish has also been studied [40]. In summary, the ability to capture position information about named entities is highly important for supporting relation extraction and the study on relation extraction can also extend to multiple languages. In feature-based models, features are combined to capture the position information. The shortcoming of feature-based models is that manually designed rules should be used to generate combined features. Furthermore, comparing with distributed representations of words, categorical features are not effective to encode semantic information of words. In kernel-based models, dependency trees are used to model the sentence structure. The main problem of kernel-based models is that generating dependency trees depends strongly on external toolkits, which are also error-prone. This process also depends on manually designed distance functions for computing the similarities between the relation instances. The neural network is the most popular technique for relation extraction. Position embedding, multichannel, PCNN and entity markers were proposed to capture the position information. The shortcoming of these methods is that they cannot simultaneously capture both position information and semantic information.

3. Methodology

A relation instance is defined as a triad $I = \langle r, a_1, a_2 \rangle$ that consists of a relation mention (r) and two arguments (a_1, a_2). A relation mention is a sentence or a clause that mentions a relation, e.g., $r = \langle w_1, w_2, \cdots, w_N \rangle$. Arguments refer to named entities in the relation mention.

In a deep neural network, word embedding is implemented to transform the one-hot representation of a word into a dense semantic space. The embedding operation is represented as:

$$[H_1^e, H_2^e, \cdots, H_N^e] = Embedding(r) \tag{1}$$

The output is referred as $\mathbf{H}^e$. The superscript e represents that it is a hidden layer outputted by an embedding layer.

Let $W_c \in R^{L \times K}$ be a filter. Then, the convolutional operation is defined as $H_i^c = f_c(W_c \cdot [H_i^e, \cdots, H_{i+K}^e] + b)$, where f_c denotes a nonlinear function, and b is a bias term. Then, the implementation of a convolutional operation through the $\mathbf{H}^e$ is represented as:

$$[H_1^c, H_2^c, \cdots, H_{N-K+1}^c] = Conv(\mathbf{H}^e). \tag{2}$$

The output of the convolutional layer is referred to as $\mathbf{H}^c \in R^{(N-K+1) \times N}$, representing a high-order abstract representation generated from the local patches of the input. It is often followed with a pooling layer to collect the salient features. A padding operation can be designed to generate a matrix $\mathbf{H}^c$ with the same column size as $\mathbf{H}^e$.

If the dependency between the inputs is introduced, the recurrent operation can be defined as $H_i^r = f_r(U_r \cdot H_i^e + V_r \cdot H_{i-1}^r + b)$, where f_r is a nonlinear function, and U_r, V_r and b are the parameters of the recurrent operation.

The recurrent operation indicates that the output H_i^r depends on both the input H_i^e and the previous state H_{i-1}^r. Implementing a recurrent operation on $\mathbf{H}^e$, the output $\mathbf{H}^r$ can be formalized as:

$$[H_1^r, H_2^r, \cdots, H_N^r] = Recu(\mathbf{H}^e) \tag{3}$$

The recurrent network has the advantage of capturing the dependency information in a sentence.

In a deep neural network, the CNN and RNN are mainly used to support designed abstract feature transformation. They usually follow a fully connected layer ($Conn(\cdot)$) and an output layer ($Softmax(\cdot)$), carrying out a global adjustment and outputting normalized results.

For the CNN and RNN, their abilities to capture structural information are different. In the CNN, the parameters W_c and b are shared across the whole input matrix $\mathbf{H}^e$. Due to the vanishing gradient problem, the recurrent network weakly captures the long-distance dependence in a sentence. Let w_i and w_j ($i + K < j$) be two named entities in a relation mention r, where K is the size of the convolutional filter. Because the distance between w_i and w_j is larger than K, the convolutional operation cannot capture the dependency between them. In the recurrent operation, the semantic information of w_i is propagated as $H_j^r = f_r(U_r \cdot H_j^e + V_r \cdot (\cdots (f_r(U_r \cdot H_{i+1}^e + V_r \cdot H_i^r + b) \cdots)) + b)$, where the influence of w_i vanishes considerably. In this condition, both the CNN and RNN are unable to capture the dependency between the entity pair. Furthermore, named entities can be composed of any words. A neural network can be easily bewildered when learning a relation representation without position information for the named entities.

In this paper, entity indicators are proposed for capturing position information of a relation instance. Prior to introducing the entity indicator method, two traditional strategies for getting entity positions are presented, which will be compared with the entity indicator strategy.

Position Embedding: Let i and j denote the start positions of two named entities e_1 and e_2 in a sentence, respectively. Then, the coordinate of a word with position k is computed

as $[k-i, k-j]$. The vector is embedded into a higher-dimension vector, concatenated with the word embedding and fed into a neural network. This process is formalized as follows:

$$\mathbf{H}^e = [\cdots, Embedding(w_k) \oplus Embedding([k-i, k-j]), \cdots]\Big|_{k=1}^{N} \tag{4}$$

Let N, L and D represent the length of a relation mention, the dimension of word embedding and the dimension of position embedding, respectively. Then, $\mathbf{H}^e \in R^{N\times(L+D)}$ is the output of word embedding concatenated with position embedding.

Multichannel: In the multichannel method, a relation instance $r = [w_1, \cdots, w_i, \cdots, w_{i+s}, \cdots, w_j, \cdots, w_{j+t}, \cdots, w_N]$ can be partitioned into five parts by two named entities.

(1) $r^1 = [w_1, \cdots, w_{i-1}]$ (2) $r^2 = [w_i, \cdots, w_{i+s}]$
(3) $r^3 = [w_{i+s+1}, \cdots, w_{j-1}]$ (4) $r^4 = [w_j, \cdots, w_{j+t}]$
(5) $r^5 = [w_{j+t+1}, \cdots, w_N]$

Every part is viewed as an independent channel that uses a nonshared lookup table to transform every word in each channel into a vector representation. This approach can be formalized as follows:

$$y = Softmax\left(Conn\Big(\bigoplus_{k=1}^{5} Conv\big(Embedding^k(r^k)\big)\Big)\right) \tag{5}$$

The multiple channels enable the same word to express different semantic meanings in different channels. In the training process, these channels do not interact during recurrent propagation, enabling the neural network to learn different representations for the same word.

3.1. Entity Indicators

In a traditional relation extraction task, named entities are manually labeled and given as inputs. They have precise positions in a sentence. By inserting specific tokens next to the boundaries of named entities, it enables a deep neural network to concentrate on the considered entity pair. Each indicator is encoded into the same representation, which can be seen as the "anchor" of a relation instance. Therefore, in addition to pointing to the positions of arguments, they are beneficial for learning the dependencies between the entity indicators and words.

Let $r = \langle w_1, \cdots, w_i, \cdots, w_{i+s}, \cdots, w_j, \cdots, w_{j+t}, \cdots, w_N\rangle$ be a relation mention where $e_1 = w_i, \cdots, w_{i+s}$ and $e_2 = w_j, \cdots, w_{j+t}$ denote two named entities in r. Because of asymmetrical relation types, the relation mention r can generate two relation instances: $I_1 = \langle r, a_1 = e_1, a_2 = e_2\rangle$ and $I_2 = \langle r, a_1 = e_2, a_2 = e_1\rangle$. In a relation instance, four indicators are inserted into two sides of arguments a_1 and a_2, denoted as l_{11}, l_{12}, l_{21} and l_{22}, respectively. Then, the relation mentions for the relation instances I_1 and I_2 are revised as:

$$\begin{aligned} &\langle w_1, \cdots, l_{11}, w_i, \cdots, w_{i+s}, l_{12}, \cdots, l_{21}, w_j, \cdots, w_{j+t}, l_{22}, \cdots, w_N\rangle \\ &\langle w_1, \cdots, l_{21}, w_i, \cdots, w_{i+s}, l_{22}, \cdots, l_{11}, w_j, \cdots, w_{j+t}, l_{12}, \cdots, w_N\rangle \end{aligned} \tag{6}$$

where l_{11}, l_{12}, l_{21} and l_{22} denote predefined tokens. When a relation mention is implanted with these indicators and fed into a neural network, the network will "know" information about the positions of the arguments. Furthermore, entity indicators can encode more information, such as syntactic or semantic information. In this paper, three types of entity indicators are proposed: position indicators, semantic indicators and compound indicators. To simplify our discussion, entity indicators of arguments a_1 and a_2 are denoted as a quadruple $\langle l_{11}, l_{12}, l_{21}, l_{22}\rangle$.

Position Indicators: These indicators point to the positions of two arguments in a relation mention. To show the influence of position indicators, they are divided into

three subcategories. In the first category, all indicators use the same mark, for example, $\langle [P], [P], [P], [P] \rangle$, where "$[P]$" is a token that indicates that it is a boundary of an argument. In the second category, the beginning and end boundaries of a named entity are distinguished, e.g., $\langle [P], [/P], [P], [/P] \rangle$, where $[P]$ and $[/P]$ represent the beginning and end boundaries of an argument, respectively. In the third category, entity indicators are modified with marks about the attached arguments, e.g., $\langle [P_1], [/P_1], [P_2], [/P_2] \rangle$, representing the beginning and end boundaries relative to a_1 and a_2, respectively.

Semantic Indicators: The entity type and subtype contain important semantic information about named entities. Therefore, using the entity type and subtype as entity indicators can capture both entity positional information and entity semantic information. Entity indicators can be combined with entity types, subtypes and argument positions and are divided into four subcategories: entity type indicator, entity subtype indicator, entity type with position and entity subtype with position. For example, let "PER" and "ORG" represent the entity types person and organization, respectively. "IND" and "GOV" represent the entity subtypes individual and government, respectively. Two indicator quadruples can be generated as $\langle [PER], [/PER], [ORG], [/ORG] \rangle$ or $\langle [IND], [/IND], [GOV], [/GOV] \rangle$. To distinguish arguments between a_1 and a_2, subscripts can also be added, e.g., $\langle [PER_1], [/PER_1], [ORG_2], [/ORG_2] \rangle$, where $[/PER_1]$ means that it is an end boundary of argument a_1 with entity type person.

Compound Indicators: The above semantic indicators have shown the ability to combine semantic information and positional information (e.g., $[/PER_1]$). This strategy can be extended further to generate more complex indicators, which are referred as compound indicators. Compound indicators have a two-side effect. On the one hand, they encode more syntactic or semantic information, which is beneficial for enhancing the discriminability of a neural network. On the other hand, they also lead to a sparse representation, which disperses the significance of the indicators. In this paper, two types of compound indicators are evaluated to demonstrate their influence on the performance. The first utilizes the entity type and subtype simultaneously. For example, $[/PER_IND_1]$ combines three types of information about the entity type person (PER), subtype individual (IND) and argument position (a_1). In the second type, syntactic features are introduced to indicate the Part-of-speech (POS) tag of the left word (e.g., $[V_IND_1]$) or the POS tag of the right word (e.g., $[/N_IND_1]$), where V and N represent *verb* and *noun*, respectively.

In summary, three types and nine subtypes of entity indicators are proposed in this paper. They are listed in Table 1.

Table 1. Types of entity indicators.

TYPE	SUBTYPE	Abbreviation	Examples
Position Indicators	Dull Positions	P_D	$[P], [P]$
	Two-side Positions	P_TS	$[P], [/P]$
	Two-side-ARG Positions	P_TSA	$[P_1], [/P_1]$
Semantic Indicators	Two-side Types	S_TS_T	$[PER], [/PER]$
	Two-side Subtypes	S_TS_S	$[IND], [/IND]$
	Two-side-ARG Types	S_TSA_T	$[PER_1], [/PER_1]$
	Two-side-ARG Subtypes	S_TSA_S	$[IND_1], [/IND_1]$
Compound Indicators	Dual-types-side-ARG	C_DTSA	$[PER_IND_1], [/PER_IND_1]$
	Type-POS-side-ARG	C_PTSA	$[V_PER_1], [/V_PER_1]$

In all the indicators, the entity type and subtype are manually annotated in the employed corpus. They are widely used in the field of information extraction. To generate POS tags of words, two popular POS tools *JIEBA* and *NLTK* were adopted for Chinese and English, respectively. In addition to the above three indicator types, syntactic indicators can be used. For example, $[V_1], [/N_1]$ indicate that for the first argument, the left word is a verb and the right word is a noun. However, because POS tags are generated by external toolkits that are error-prone, they are not used independently.

3.2. Model

After entity indicators were inserted into relation mentions, they are ready for processing by a deep neural network. In this paper, we designed a simple but effective architecture to evaluate the effectiveness of entity indicators. This model is composed of an input layer, an embedding layer, a convolutional layer and an output layer. The architecture of this model is shown in Figure 1.

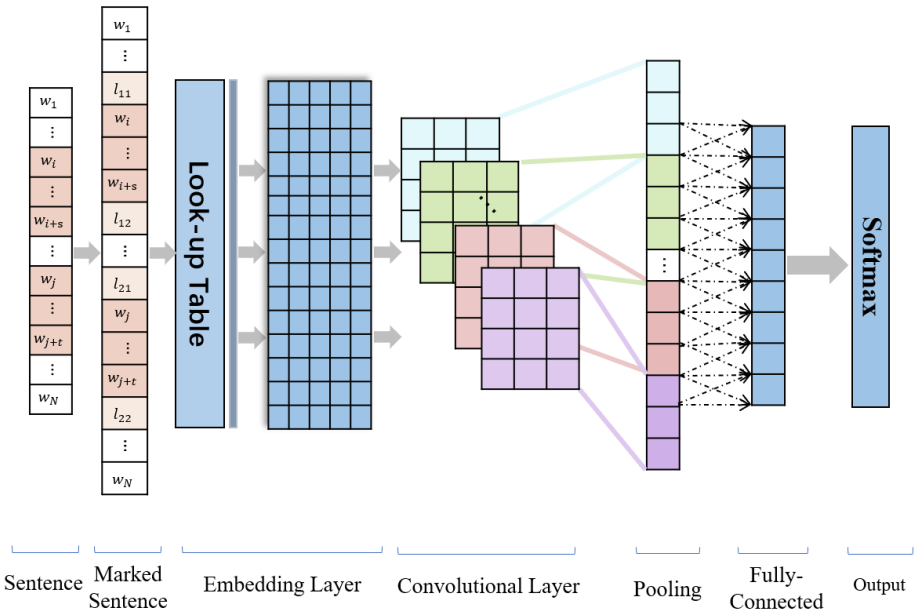

Figure 1. Framework of neural network.

In the input layer, instead of directly input an original sentence, entity indicators are implanted into relation mentions to point to the position, syntactic and semantic information of the arguments.

In the embedding layer, three approaches are adopted to support word embedding. In the first approach, a randomly initialized lookup table is adopted to support embedding. In the second approach, wiki-100 [41] and GoogleNews-vectors-negative-300 are adopted for Chinese and English word embeddings, respectively. The third model is based on BERT [36], which is pretrained with external resources by an unsupervised method. This approach is effective to capture semantic information of words. Furthermore, BERT is based on the Transformer [11], which can learn the dependency between words.

The convolutional layer performs four one-dimensional convolutional operations with a kernel shape of 3 × 1 on the output of the embedding layer. The output of this operation is a vector with size 50. The convolutional layer automatically learns abstract representation from local features. The output of the convolutional layer is fed into max-pooling and fully connected operations. The pooling layer collects salient features from the input, which reduces the parameters of the model and extends the generalizability of the model. A fully connected layer resembles the features for global regulation. In the output layer, a cross-entropy loss function is adopted to calculate the loss during the training process (Our code to implement deep learning model with entity indicators for relation extraction is available at: https://github.com/WeizheYang-SHIN/Entity_indicators_for_RE, accessed on 20 March 2021).

4. Experiments

The experiment used a NVIDIA Tesla P40 GPU for training and testing under Linux environment. In this section, three datasets the ACE 2005 Chinese corpus, ACE 2005 English

corpus [42] and Chinese literature text corpus (CLTC) [43] were adopted to evaluate the performance of entity indicators.

The ACE 2005 corpus is a classic dataset for automatic content extraction. It is collected from weblogs, broadcast news, newsgroups and broadcast conversation. The corpus was annotated with 7 entity types, 44 entity subtypes, 6 relation types and 18 relation subtypes. The Chinese corpus contains 628 documents, containing 9244 relation mentions that are used as positive instances. If two named entities have no predefined relation type, the instance should be considered as negative. To generate negative instances for training, the method in Chen et al. [25] is adopted, which generates 98,140 negative instances. The ACE English corpus generates 6583 positive instances and 97,534 negative instances.

The Chinese literature text corpus [43] is a discourse-level corpus whose articles are collected from Chinese literature. In total, seven entity types and nine relation types are manually annotated according to a gold standard. In this corpus, the entity relation only exists in four entity types: Thing, Person, Location and Organization. The corpus contains a total of 695 articles for training, 58 for validation and 84 for testing.

The experimental performance is measured by the commonly used evaluation indexes in the NLP field, namely, precision rate (P), recall rate (R) and comprehensive evaluation (F1). The total performance (referred to as "Total") is the macro-average over all positive relation types. The ACE 2005 corpus was divided into 8:1:1 for training, validation and testing in our experiments. The fixed length of a sentence is set as 100.

4.1. Performance of Entity Indicators

In this section, based on the ACE Chinese corpus, ACE English corpus and CLTC corpus, all the entity indicators in Table 1 were evaluated to demonstrate their influence on the performance. In this experiment, word embeddings were initialized by two pretrained embedding models, wiki-100 [41] for Chinese and GoogleNews-vectors-negative-300 for English. The results are shown in Table 2, where the "None" model was implemented on the relation mentions directly without any entity indicators and was implemented for comparison. "×" means that the performance is not supported because no entity subtype was annotated in the CLTC corpus.

Table 2. Performance of the entity indicators.

Entity Indicators	ACE Chinese			ACE English			CLTC		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
None	71.29	54.92	60.32	69.54	53.43	60.40	50.64	29.28	37.10
P_D	80.11	59.54	68.31	79.49	61.24	69.18	64.38	55.34	59.52
P_TS	77.42	63.25	69.16	80.80	61.36	69.75	62.45	59.20	60.78
P_TSA	78.65	60.02	68.08	80.13	60.81	69.15	61.41	56.74	58.98
S_TS_T	85.27	65.89	74.34	88.01	67.63	76.49	70.13	62.79	66.26
S_TS_S	83.90	67.89	75.05	84.90	68.55	75.85	×	×	×
S_TSA_T	84.18	71.39	77.30	84.93	70.13	76.82	75.23	72.30	73.74
S_TSA_S	85.91	67.22	75.42	87.03	69.85	77.50	×	×	×
C_DTSA	85.32	70.70	77.33	**86.75**	**71.37**	**78.31**	×	×	×
C_PTSA	**91.18**	**88.55**	**89.85**	85.92	69.94	77.11	**75.90**	**73.57**	**74.72**

Comparison of the ACE Chinese corpus and the CLTC corpus shows that the performance is higher for the ACE Chinese corpus. This is because in the CLTC corpus, sentence semantics are often expressed in a subtle and special manner [44]. Their intuitions and feelings are usually expressed through very complex and flexible sentence structures

The results show that distinguishing the beginning and end boundaries of named entities is beneficial for improving the performance (e.g., "P_D " and "P_TS"). When entity types are encoded into the entity indicators, the performance is improved considerably. All the entity indicators considerably outperform the "None" strategy. As Table 2 shows, the performance is improved with entity indicators encoding more relevant information.

When the POS tags are used in the indicator "C_PTSA ", the ACE Chinese corpus shows an impressive improvement, outperforming the " C_DTSA" by more than 12% in F1 score. An improvement was also observed for the CLTC corpus. On the other hand, on the

English corpus, the performance is unexpectedly decreased with POS tags. The reason of the difference may be that English is an alphabetic language, where many adjacent words are function words (e.g., articles, auxiliary words), which are more ambiguous and have little lexical meaning, possibly affecting the performance.

4.2. Comparison with Other Strategies

In this experiment, the entity indicator method was compared with the position embedding and multichannel methods to demonstrate their ability to capture the entity positions of a relation instance. The performance on the three corpora is shown in Table 3.

Table 3. Comparison with traditional strategies.

Data	TYPE	None			Position Emb.			Multichannel			Entity Indicator		
		P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
ACE Chinese	PHYS	39.70	45.40	42.36	53.21	33.33	40.99	55.00	25.29	34.65	85.88	83.91	84.88
	ART	64.00	20.51	31.07	66.67	35.90	46.67	70.00	26.92	38.89	95.45	80.77	87.50
	GEN-AFF	79.35	67.91	73.18	67.27	68.84	68.05	76.29	68.84	72.37	84.62	86.98	85.78
	ORG-AFF	82.01	73.46	77.50	70.82	78.20	74.32	88.70	74.41	80.93	95.22	94.31	94.76
	PART-WHOLE	79.36	73.62	76.38	66.91	76.60	71.43	78.32	75.32	76.79	87.34	88.09	87.71
	PER-SOC	83.33	48.61	61.40	65.22	62.50	63.83	73.08	52.78	61.29	98.59	97.22	97.90
	Total	71.29	54.92	60.32	65.02	59.23	61.99	73.56	53.93	62.23	91.18	88.55	89.85
ACE English	PHYS	62.96	31.29	41.80	75.00	34.94	47.70	82.61	34.97	49.14	72.86	62.58	67.33
	ART	71.05	42.19	52.94	85.29	45.31	59.18	86.11	48.44	62.00	88.37	59.38	71.03
	GEN-AFF	56.86	43.28	49.15	67.31	52.24	58.82	69.09	56.72	62.30	90.24	55.22	68.52
	ORG-AFF	70.26	74.05	72.11	84.80	78.38	81.46	79.49	83.78	81.58	90.45	87.03	88.71
	PART-WHOLE	66.67	59.77	63.03	68.82	73.56	71.11	74.67	64.37	69.14	85.71	82.76	84.21
	PER-SOC	88.89	70.00	78.32	85.51	73.75	79.19	89.71	76.25	82.43	92.86	81.25	86.67
	Total	69.54	53.43	60.40	77.79	59.70	67.55	80.28	60.75	67.76	86.75	71.37	78.31
CLTC	Ownership	26.67	4.65	7.92	50.00	9.30	15.69	63.86	61.63	62.72	82.61	66.28	73.55
	Create	17.65	7.89	10.91	39.29	28.95	33.33	71.43	52.63	60.61	65.00	68.42	66.67
	Family	49.05	68.62	57.21	57.02	69.15	62.50	81.22	85.11	83.12	89.56	86.70	88.11
	Social	50.91	25.69	34.15	45.22	47.71	46.43	70.19	66.97	68.54	71.54	85.32	77.82
	Located	56.00	83.07	66.90	59.42	83.42	69.41	82.38	94.00	87.81	90.02	97.00	93.38
	General-Special	22.12	26.04	23.92	34.29	25.00	28.92	47.32	55.21	50.96	58.82	52.08	55.25
	Near	100.00	1.96	3.85	100.00	1.96	3.85	50.00	19.61	28.17	64.29	35.29	45.57
	Use	85.71	5.56	10.43	52.17	22.22	31.17	75.96	73.15	74.53	76.34	92.59	83.68
	Part-Whole	47.66	40.05	43.52	55.47	48.15	51.55	80.79	71.06	75.62	84.96	78.47	81.59
	Total	50.64	29.28	37.10	54.76	37.32	44.39	69.24	64.37	66.72	75.90	73.57	74.72

The "None" is the same as in Table 2. In the position embedding model, the position coordinates of every word are embedded into a 25-dimensional vector. In the multichannel model [13], each channel employs an independent lookup table for word embedding that does not interact during recurrent propagation. In the entity indicator model, the "C_PTSA" encoding is used for the ACE Chinese and CLTC corpora, while the "C_DTSA" encoding is adopted for the ACE English corpus. To avoid the influence caused by external resources, all the word embeddings are initialized by randomly initialized lookup tables.

Position embedding and multichannel are two traditional strategies to support relation extraction [13,18]. A comparison of Tables 2 and 3 shows that in the CLTC corpus, the performance displays impressive improvements. However, position embedding and multichannel have little effect on the ACE corpus. This is because ACE suffers from a serious data imbalance problem. Furthermore, because the relation types of ACE are asymmetrical, every positive instance (e.g., $\langle s, e_1, e_2 \rangle$) has a corresponding negative instance (e.g., $\langle s, e_2, e_1 \rangle$). In this condition, the position information of named entities is more complex, worsening the final performance.

Compared with other strategies, entity indicators show remarkable improvements. In both the ACE corpus and the CLTC corpus, the performance is stably increased. The reason for the improvement is that named indicators encode positional information (e.g., entity boundary positions), syntactic information (e.g., POS tags) and semantic information (e.g., entity types and subtypes). Thus, they provide powerful support for relation extraction.

4.3. Evaluation on the Chinese Corpus

In the ACE Chinese corpus, the related works can be divided into those using shallow architecture and those using deep architecture. Some of them are listed in Table 4. Among the shallow architecture models, Yu et al. [45] proposed a convolutional tree kernel-based approach for relation extraction. Liu et al. [26] adopted a tree-kernel model that combines external semantic resources (HowNet) to support relation extraction. Chen [46] proposed a feature calculus strategy. In deep architecture models, Li et al. [47] proposed a lattice LSTM model that combines multigrained information for relation extraction. Chen et al. [48] designed a CNN-attention neural network model.

For comparison, the same settings as those in Chen et al. [48] were adopted in this experiment. Shallow architecture models divide the evaluation corpus into 8:2 for training and testing. In deep architecture models, the corpus is divided into 8:1:1 for training, developing and testing. In this experiment, two strategies are adopted to initialize the lookup table for word embedding. The first is "Random-CNN", which uses a randomly initialized lookup table. In the second ("BERT-CNN"), BERT [36] is adopted to support the word embedding. The embedding layer outputs 768-dimensional vectors.

Table 4. Comparison on the ACE 2005 Chinese corpus.

Model	Method	P(%)	R(%)	F1(%)
Yu et al. [45]	Convolutional kernel based on syntax and entity semantic tree.	75.30 (×)	60.43 (×)	67.00 (×)
Liu et al. [26]	Tree-Kernel with lexical semantic resources.	81.10 (79.1)	60.00 (57.5)	69.00 (66.6)
Zhang et al. [49]	Position structures between named entities.	80.71 (×)	62.48 (×)	70.43 (×)
Chen et al. [46]	Combined features for capturing structural information.	93.01 (81.41)	89.45 (72.30)	91.20 (76.59)
Li et al. [47]	Lattice LSTM with multigrained information.	× (×)	× (×)	78.17 (×)
Chen et al. [48]	A CNN and attention architecture.	82.35 (×)	79.22 (×)	80.33 (×)
Ours	Random-CNN with the "C_PTSA" encoding.	91.18 (76.94)	88.55 (73.18)	89.85 (75.01)
	BERT-CNN with the "C_PTSA" encoding.	**95.32** (**84.34**)	**94.57** (**82.69**)	**94.94** (**83.51**)

Because kernel-based models are heavily based on error-prone parsing trees, the methods reported by Yu et al. [45] and Liu et al. [26] show lower performance. Chen et al. [46] designed a feature calculus strategy to generate combined features, achieving state-of-the-art performance. In the works of Li et al. [47] and Chen et al. [48], the performance is worsened because their neural networks are directly implemented on raw inputs and cannot utilize the positional and semantic information of named entities. Compared with the related works, entity indicators receive the highest performance, outperforming the state-of-the-art by 3% in F1 score for relation types and by 7% in F1 score for relation subtypes.

In the following, the entity indicator method was evaluated on the CLTC corpus, which was published by Peking University in 2017 [43]. It contains 695 articles for training, 58 for validation and 84 for testing. Because the training, validation and testing articles are manually divided, it is convenient for comparing between different systems. Wen et al. [44] have tested several models on this corpus, and the obtained performance characteristics are listed in Table 5, where the column "Arch." denotes neural network architectures in corresponding models.

Table 5. Comparison on Chinese literature text corpus.

Model	Arch.	Information	F1(%)
Hendrickx et al. [1]	SVM	Word embeddings, NER, WordNet, dependency parse, HowNet, POS, Google n-gram	48.9
Socher et al. [50]	RNN	Word embeddings, POS, NER, WordNet	49.1
Zeng et al. [29]	CNN	Word embeddings, position embeddings, NER, WordNet	52.4
Santos et al. [51]	CR-CNN	Word embeddings, position embeddings	54.1
Xu et al. [14]	SDP-LSTM	Word embeddings, POS, NER, WordNet	55.3
Liu et al. [52]	DepNN	Word embeddings, WordNet	55.2
Cai et al. [53]	BRCNN	Word embeddings, POS, NER, WordNet	55.6
Zhang et al. [54]	C-ATT-BLSTM	Character embedding, position embedding, entity sense	56.2
Wen et al. [44]	SR-BRCNN	Word embeddings, POS, NER, WordNet	65.9
Ours	Random-CNN	Entity indicators with the S_TSA_T encoding.	74.72
	BERT-CNN	Entity indicators with the S_TSA_T encoding.	**77.14**

In the compared models, Zhang et al. [54] proposed a multifeature fusion model, which integrates multilevel features into deep neural network models. The convolutional layer is built on the Att-BLSTM model to capture the features at the word level to obtain more structural information. Wen et al. [44] obtained the highest performance by implementing two bidirectional LSTMs on the shortest dependency path of relation instances. Because Chinese sentences are insensitive to structure, parsing a Chinese sentence is error-prone and depends strongly on external toolkits. Therefore, the performance is influenced by the parsing process. On the other hand, entity indicators are based on manually annotated named entities. They contain precise position information about relation arguments. Combined with POS tags and entity types, the entity indicators are effective for utilizing entity positional and semantic information. Compared with that of the SR-BRCNN model, the performance of this approach is improved by more than 11% in terms of the F1-score.

4.4. Evaluation on the English Corpus

In this experiment, the entity indicator is compared with several related works implemented on the ACE English corpus.

Kambhatla et al. [20] proposed a feature-based maximum entropy (ME) model. Zheng et. al. [16] used multiple CNN convolution kernels of different sizes to extract features from raw inputs. Gormley et al. [55] presented a feature-rich compositional embedding model (FCM) that combines handcrafted features and word embeddings. Zhou et al. [56] adopted an SVM model with diverse lexical, syntactic and semantic knowledge. Zhong et al. [19] used encoding of entity markers to represent the relation mention based on BERT model. Chen et al. [46] proposed a feature calculus method, where combined features are generated for capturing structural information of the sentences. The results are presented in Table 6.

The results showed that entity indicators outperformed the compared work by approximately 5% F1-score in relation types and approximately 18.5% F1-score in relation subtypes.

Table 6. Comparison on the ACE 2005 English corpus.

Model	Arch.	Method	P(%)	R(%)	F1(%)
Kambhatla et al. [20]	ME	A feature-based model.	63.50 (×)	45.20 (×)	52.80 (×)
Zheng et al. [16]	MIX-CNN	Automatically extract features based on multiple CNNs.	60.00 (×)	48.40 (×)	53.60 (×)
Gormley et al. [55]	FCM	Combine features and word embeddings.	71.52 (×)	49.32 (×)	58.26 (×)
Zhou et al. [56]	SVM	Phrase chunking information	77.20 (63.10)	60.70 (49.50)	68.00 (55.50)
Zhong et al. [19]	BERT	Entity marker embedding	× (×)	× (×)	73.10 (×)
Chen et al. [46]	SSM	Feature calculus	84.50 (71.89)	77.32 (56.06)	80.75 (63.03)
Ours	BERT-CNN	Entity indicators with the C_PTSA encoding.	**88.83** (**88.82**)	**82.78** (**75.67**)	**85.70** (**81.72**)

5. Conclusions

Unlike sentence classification that makes a prediction based on sentence representation, relation extraction should consider the semantic information between two named entities. Because a sentence often contains several named entities that share the same context, directly making a decision-based sentence representation learned from raw inputs is not effective for supporting relation extraction. In this paper, entity indicators were proposed to capture position information of a relation instance. Instead of implementation on a raw input, task-related entity indicators are inserted into each relation instance. This strategy lets the neural network "know" the position, syntactic and semantic information of the named entities about a relation instance. The uniformly structured indicators can make the similarity between sentences and the internal symmetry of one sentence more obviously. It also helps neural networks to learn semantic dependencies in a relation instance. Experiments have shown that this is a powerful approach for supporting relation extraction. In further work, the notion of using entity indicators can be extended to other NLP tasks (e.g., named entities) for a neural network to capture the structural and semantic information of a sentence.

Author Contributions: Y.Q.: formal analysis, funding acquisition, project administration and supervision; W.Y.: data curation, investigation, methodology, resources and writing—original draft; K.W.: investigation and writing—review and editing; R.H.: funding acquisition and supervision; F.T.: supervision and writing—review and editing; S.A.: data curation; Y.C.: conceptualization, formal analysis, methodology, supervision and writing—review and editing. All authors have read and agreed to the final version of the manuscript.

Funding: 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, the Key Projects of Science and Technology of Guizhou Province under Grant No. [2020] 1Z055 and Project of Guizhou Province Graduate Research Fund (Qianjiaohe YJSCXJH[2019]102).

Acknowledgments: Thanks to the editors and anonymous reviewers for their valuable suggestions and comments that made our final version of the paper more perfect.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hendrickx, I.; Kim, S.N.; Kozareva, Z.; Nakov, P.; Séaghdha, D.O.; Padó, S.; Pennacchiotti, M.; Romano, L.; Szpakowicz, S. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In Proceedings of the 5th International Workshop on Semantic Evaluation, ACL, Uppsala, Sweden, 15–16 July 2010; pp. 33–38.
2. Agosti, M.; Nunzio, G.M.D.; Marchesin, S.; Silvello, G. A relation extraction approach for clinical decision support. *arXiv* **2019**, arXiv:1905.01257. [CrossRef]
3. Zheng, S.; Dharssi, S.; Wu, M.; Li, J.; Lu, Z. Text mining for drug discovery. In *Bioinformatics and Drug Discovery*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 231–252.
4. Jabbari, A.; Sauvage, O.; Zeine, H.; Chergui, H. A french corpus and annotation schema for named entity recognition and relation extraction of financial news. In Proceedings of the LREC '20, Marseille, France, 11–16 May 2020; pp. 2293–2299.
5. Macdonald, E.; Barbosa, D. Neural relation extraction on wikipedia tables for augmenting knowledge graphs. In Proceedings of the CIKM '20, Galway, Ireland, 17–20 August 2020; pp. 2133–2136. [CrossRef]
6. Li, X.; Yin, F.; Sun, Z.; Li, X.; Yuan, A.; Chai, D.; Zhou, M.; Li, J. Entity-relation extraction as multi-turn question answering. *arXiv* **2019**, arXiv:1905.05529.
7. Han, R.; Liang, M.; Alhafni, B.; Peng, N. Contextualized word embeddings enhanced event temporal relation extraction for story understanding. *arXiv* **2019**, arXiv:1904.11942. [CrossRef]
8. Liu, K. A survey on neural relation extraction. *Sci. China Technol. Sci.* **2020**, *63*, 1971–1989. [CrossRef]
9. Liu, C.Y.; Sun, W.B.; Chao, W.H.; Che, W.X. Convolution neural network for relation extraction. In Proceedings of the DMA 2013: Advanced Data Mining and Applications, Hangzhou, China, 14–16 December 2013.
10. Li, Z.; Yang, J.; Gou, X.; Qi, X. Recurrent neural networks with segment attention and entity description for relation extraction from clinical texts. *Artif. Intell. Med.* **2019**, *97*, 9–18.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
12. Zeng, D.; Liu, K.; Chen, Y.; Zhao, J. Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015. [CrossRef]
13. Chen, Y.; Wang, K.; Yang, W.; Qing, Y.; Huang, R.; Chen, P. A multi-channel deep neural network for relation extraction. *IEEE Access* **2020**, *8*, 13195–13203.
14. Xu, Y.; Mou, L.; Li, G.; Chen, Y.; Peng, H.; Jin, Z. Classifying relations via long short term memory networks along shortest dependency paths. In Proceedings of the Proceedings of the EMNLP 2015, Lisbon, Portugal, 17–21 September 2015; pp. 1785–1794. [CrossRef]
15. Zhang, C.; Xu, W.; Ma, Z.; Gao, S.; Guo, J. Construction of semantic bootstrapping models for relation extraction. *Knowl. Based Syst.* **2015**, *83*, 128–137.
16. Zheng, S.; Xu, J.; Zhou, P.; Bao, H.; Qi, Z.; Xu, B. A neural network framework for relation extraction: Learning entity semantic and relation pattern. *Knowl. Based Syst.* **2016**, *114*, 12–23.
17. Soares, L.B.; FitzGerald, N.; Ling, J.; Kwiatkowski, T. Matching the blanks: Distributional similarity for relation learning. *arXiv* **2019**, arXiv:1906.03158. [CrossRef]
18. Zhang, D.; Wang, D. Relation classification via recurrent neural network. *arXiv* **2015**, arXiv:1508.01006.
19. Zhong, Z.; Chen, D. A frustratingly easy approach for joint entity and relation extraction. *arXiv* **2020**, arXiv:2010.12812.
20. Kambhatla, N. Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. In Proceedings of the ACL (07 2004), Barcelona, Spain, 21–26 July 2004. [CrossRef]
21. Zelenko, D.; Aone, C.; Richardella, A. Kernel methods for relation extraction. *J. Mach. Learn. Res.* **2003**, *3*, 1083–1106.
22. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567.
23. Dashtipour, K.; Gogate, M.; Adeel, A.; Algarafi, A.; Howard, N.; Hussain, A. Persian named entity recognition. In Proceedings of the 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), Oxford, UK, 26–28 July 2017; pp. 79–83. [CrossRef]
24. Minard, A.-L.; Ligozat, A.-L.; Grau, B. Multi-class svm for relation extraction from clinical reports. In Proceedings of the International Conference Recent Advances in Natural Language Processing, Varna, Bulgaria, 2–4 September 2011. [CrossRef]
25. Chen, Y.; Zheng, Q.; Chen, P. Feature assembly method for extracting relations in chinese. *Artif. Intell.* **2015**, *228*, 179–194.
26. Liu, D.; Hu, Y.; Qian, L. Exploiting lexical semantic resource for tree kernel-based chinese relation extraction. In Proceedings of the NLPCC, Beijing, China, 31 October–5 November 2012; pp. 213–224. [CrossRef]
27. Panyam, N.C.; Verspoor, K.; Cohn, T.; Kotagiri, R. Asm kernel: Graph kernel using approximate subgraph matching for relation extraction. In Proceedings of the ALTA 2016, Perth, Australia, 21–28 May 2016; pp. 65–73. [CrossRef]
28. Leng, J.; Jiang, P. A deep learning approach for relationship extraction from interaction context in social manufacturing paradigm. *Knowl. Based Syst.* **2016**, *100*, 188–199.
29. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the COLING'14, Dublin, Ireland, 23–29 August 2014. [CrossRef]

30. Li, Y.; Nee, M.; Li, G.; Chang, V. Effective piecewise cnn with attention mechanism for distant supervision on relation extraction task. In Proceedings of the 5th International Conference on Complexity, Future Information Systems and Risk 2020 (COMPLEXIS 2020), Prague, Malta, 8–9 May 2020. [CrossRef]
31. Zhang, C.; Zheng, Y.; Guo, B.; Li, C.; Liao, N. Scn: A novel shape classification algorithm based on convolutional neural network. *Symmetry* **2021**, *13*, 1651. [CrossRef]
32. Wang, H.; Qin, K.; Lu, G.; Luo, G.; Liu, G. Direction-sensitive relation extraction using bi-sdp attention model—Sciencedirect. *Knowl. Based Syst.***2020**, *198*, 105928. [CrossRef]
33. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 2, pp. 207–212.
34. Lee, J.; Seo, S.; Choi, Y.S. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. *Symmetry* **2019**, *11*, 785. [CrossRef]
35. Zhao, L.; Xu, W.; Gao, S.; Guo, J. Cross-sentence n-ary relation classification using lstms on graph and sequence structures. *Knowl. Based Syst.* **2020**, *207*, 106266. [CrossRef]
36. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
37. Huang, W.; Mao, Y.; Yang, Z.; Zhu, L.; Long, J. Relation classification via knowledge graph enhanced transformer encoder. *Knowl. Based Syst.* **2020**, *206*, 106321. [CrossRef]
38. McDonough, K.; Moncla, L.; van de Camp, M. Named entity recognition goes to old regime france: geographic text analysis for early modern french corpora. *Int. J. Geogr. Inf. Sci.* **2019**, *33*, 2498–2522. [CrossRef]
39. Isozaki, H. Japanese named entity recognition based on a simple rule generator and decision tree learning. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, France, 6–11 July 2001; pp. 314–321.
40. Weegar, R.; Pérez, A.; Casillas, A.; Oronoz, M. Deep medical entity recognition for swedish and spanish. In Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Madrid, Spain, 3–6 December 2018; pp. 1595–1601. [CrossRef]
41. Dong, C.; Zhang, J.; Zong, C.; Hattori, M.; Di, H. Character-based lstm-crf with radical-level features for chinese named entity recognition. In *Natural Language Understanding and Intelligent Applications*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 239–250. [CrossRef]
42. Walker, C.; Strassel, S.; Medero, J.; Maeda, K. Ace 2005 multilingual training corpus. *Linguist. Data Consort. Phila.* **2006**, *57*, 45.
43. Xu, J.; Wen, J.; Sun, X.; Su, Q. A discourse-level named entity recognition and relation extraction dataset for chinese literature text. *arXiv* **2017**, arXiv:1711.07010.
44. Wen, J.; Sun, X.; Ren, X.; Su, Q. Structure regularized neural network for entity relation classification for chinese literature text. *arXiv* **2018**, arXiv:1803.05662.
45. Liu, D.; Zhao, Z.; Hu, Y.; Qian, L. Chinese semantic relation extraction based on syntax and entity semantic tree. *J. Chin. Inf. Process.* **2010**, *24*, 11–21.
46. Chen, Y.; Zheng, Q.; Chen, P. A set space model for feature calculus. *IEEE Intell. Syst.* **2017**, *32*, 36–42. [CrossRef]
47. Li, Z.; Ding, N.; Liu, Z.; Zheng, H.; Shen, Y. Chinese relation extraction with multi-grained information and external linguistic knowledge. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4377–4386.
48. Chen, Y.; Wang, G.; Zheng, Q.; Qin, Y.; Huang, R.; Chen, P. A set space model to capture structural information of a sentence. *IEEE Access* **2019**, *7*, 142515–142530.
49. Zhang, P.; Li, W.; Hou, Y.; Song, D. Developing position structure-based framework for chinese entity relation extraction. *ACM Trans. Asian Lang. Inf. Process.* **2011**, *10*. [CrossRef]
50. Socher, R.; Pennington, J.; Huang, E.H.; Ng, A.Y.; Manning, C.D. Semi-supervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Scotland, UK, 27–31 July 2011; pp. 151–161.
51. Santos, C.N.d.; Xiang, B.; Zhou, B. Classifying relations by ranking with convolutional neural networks. *arXiv* **2015**, arXiv:1504.06580.
52. Liu, Y.; Wei, F.; Li, S.; Ji, H.; Zhou, M.; Wang, H. A dependency-based neural network for relation classification. *arXiv* **2015**, arXiv:1507.04646.
53. Cai, R.; Zhang, X.; Wang, H. Bidirectional recurrent convolutional neural network for relation classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1, pp. 756–765.
54. Zhang, J.; Hao, K.; Tang, X.s.; Cai, X.; Xiao, Y.; Wang, T. A multi-feature fusion model for chinese relation extraction with entity sense. *Knowl. Based Syst.* **2020**, *206*, 106348.
55. Gormley, M.R.; Yu, M.; Dredze, M. Improved relation extraction with feature-rich compositional embedding models. *arXiv* **2015**, arXiv:1505.02419.
56. Zhou, G.; Su, J.; Zhang, J.; Zhang, M. Exploring various knowledge in relation extraction. In Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05), Ann Arbor, MI, USA, 25–30 June 2005; pp. 427–434.