

Article

Contextual Semantic-Guided Entity-Centric GCN for Relation Extraction

Jun Long ^{1,2}, Lei Liu ², Hongxiao Fei ², Yiping Xiang ², Haoran Li ², Wenti Huang ^{3,*} and Liu Yang ^{2,*}¹ School of Software, Xinjiang University, Urumqi 830046, China; junlong@csu.edu.cn² School of Computer Science and Engineering, Central South University, Changsha 410083, China; lei153@csu.edu.cn (L.L.); hxfei@csu.edu.cn (H.F.); 194712189@csu.edu.cn (Y.X.); 194712136@csu.edu.cn (H.L.)³ School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

* Correspondence: wthuang@hnust.edu.cn (W.H.); yangliu@csu.edu.cn (L.Y.)

Abstract: Relation extraction tasks aim to predict potential relations between entities in a target sentence. As entity mentions have ambiguity in sentences, some important contextual information can guide the semantic representation of entity mentions to improve the accuracy of relation extraction. However, most existing relation extraction models ignore the semantic guidance of contextual information to entity mentions and treat entity mentions in and the textual context of a sentence equally. This results in low-accuracy relation extractions. To address this problem, we propose a contextual semantic-guided entity-centric graph convolutional network (CEGCN) model that enables entity mentions to obtain semantic-guided contextual information for more accurate relational representations. This model develops a self-attention enhanced neural network to concentrate on the importance and relevance of different words to obtain semantic-guided contextual information. Then, we employ a dependency tree with entities as global nodes and add virtual edges to construct an entity-centric logical adjacency matrix (ELAM). This matrix can enable entities to aggregate the semantic-guided contextual information with a one-layer GCN calculation. The experimental results on the TACRED and SemEval-2010 Task 8 datasets show that our model can efficiently use semantic-guided contextual information to enrich semantic entity representations and outperform previous models.

Keywords: graph convolutional network; relation extraction; machine learning; natural language processing

MSC: 68T50



Citation: Long, J.; Liu, L.; Fei, H.; Xiang, Y.; Li, H.; Huang, W.; Yang, L. Contextual Semantic-Guided Entity-Centric GCN for Relation Extraction. *Mathematics* **2022**, *10*, 1344. <https://doi.org/10.3390/math10081344>

Academic Editor: Victor Mitrana

Received: 10 March 2022

Accepted: 13 April 2022

Published: 18 April 2022

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



Copyright: © 2022 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 an important task in natural language processing (NLP) which aims to predict the semantic relations between entities. It extracts special events or information in unstructured text. For example, it can extract events, institutions, and people relations from reports. Therefore, relation extraction is widely used in downstream natural language processing (NLP) tasks, such as information relation extraction [1,2], knowledge network construction [3,4], and intelligent question-answering systems [5,6].

In recent years, deep learning models have made remarkable progress in many research areas, such as convolutional neural networks (CNNs) [7], recurrent neural networks (RNNs) [8], and other neural network architectures [9], which are widely used in relation extraction tasks. These models convert words or phrases in text into low-dimensional vectors through NLP processing tools and obtain the word-level or sentence-level semantic representation through a feature extractor. Finally, the relation between the entity

pair is acquired through a specifically designed classifier. However, in entity relation extraction processing, predicates have significant meaning, which means long distances between entities and predicates will cause semantic information loss. To solve this problem, the dependency tree [10] is proposed to capture remote semantic information. To better obtain the semantic information from the dependency tree, the SDP-LSTM model [11] applies long short-term memory (LSTM) to obtain the shortest dependency path between entities. Zhang et al. [12] propose an extended graph convolutional network (GCN) to train a dependency tree with a pruning strategy to obtain important words in the shortest path. Compared with CNN and LSTM, GCN [12] can parallelly process non-Euclidean data and align trees for efficient batch training, which is used widely in image recognition [13], visual reasoning [14], and biological graph generation [15].

Although the previous results are obtained using GCN-based models, they treat textual contexts and entities equally with the graph convolutional operation. Entity representations cannot obtain semantic-guided contextual information from sentences, and the ambiguity of the entity mentions affect the relation extraction results. Therefore, the impact of semantic-guided contextual information on entity mentions in a sentence is still worth investigating. For example, in the following sentence (S1), “Donald Trump is the 45th president of the United States”, the relation between entities is “president_of”. However, in the following sentence (S2), “Donald Trump was born in the United States”, the relation between entities is “born_in”. We can observe that the entity mentions (Donald Trump and United States) have ambiguity in different sentences. The textual context can guide the semantic information of entity mentions in a sentence, such as “the president of” in S1 and “was born in” in S2; these phrases are strongly semantic-guided. Focusing the semantic information of textual contexts on entity mentions can improve the precision of the relation extraction.

To address these problems, our paper proposes a novel GCN model for relation extraction. Firstly, we propose a self-attention enhanced neural network that consists of extended LSTM with a gate mechanism and a multi-head self-attention mechanism. Both mechanisms are arranged in a parallel manner. This model can capture the long-distance dependency and concentrate on the relevance and importance of different words in a sentence to highlight the semantic information of crucial words. By combing the output of both parallel modules, we can obtain semantic-guided contextual information. The latest GCN model based on a sentence dependency tree enables global nodes to aggregate the semantic information of all nodes. Therefore, we build a dependency tree with entities as global nodes and add virtual edges to construct an entity-centric logical adjacency matrix (ELAM). This matrix enables entities to aggregate semantic-guided contextual information. Finally, we model the association between the subject and object entities, and use a difference vector as a part of the relation extraction constraint.

We evaluated the performance of the model on two popular datasets: the Semeval-2010 Task 8 dataset [16] and the TACRED dataset [17]. Our model achieves satisfactory performance on both datasets.

The main contributions of this paper are summarized as follows:

- We propose a self-attention enhanced neural network that captures long-distance dependency and concentrates on the importance and relevance of different words in a sentence to obtain semantic-guided contextual information;
- We propose a novel entity-centric logical adjacency matrix that enables entities to aggregate contextual semantic information with a one-layer GCN calculation.
- Finally, we analyze the complementary semantic-feature-capturing effect of the extended LSTM, GCN, and multi-head self-attention mechanisms.

2. Materials and Methods

In this section, we will introduce our novel relation extraction model (CEGCN). This model proposes a self-attention enhanced neural network and an entity-centric logical adjacency matrix to focus semantic-guided contextual information on entity representations in relation extraction to produce more accurate results. Figure 1 illustrates the overview of

the model. The model consists of four modules, including (1) a sequence encoding module, (2) a self-attention enhanced neural network module, (3) a semantic aggregation module, and (4) a relation extraction module.

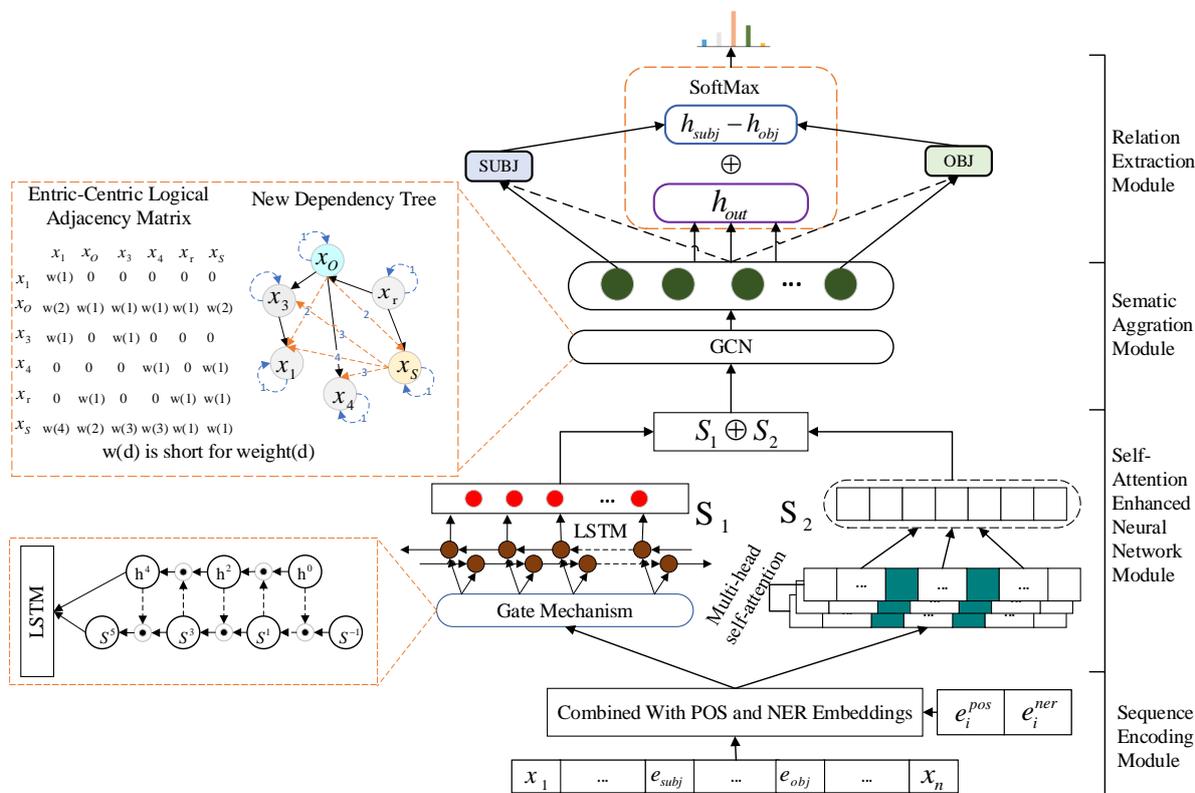


Figure 1. Model architecture diagram. The right side of the figure is the overall architecture of the model’s algorithm. The left half describes the extended LSTM with a gate mechanism and an entity-centric logical adjacency matrix (ELAM). In the gate mechanism, S^i and h_i represent the output of the i -th gating interaction, S^{-1} represents the input sentence S , and h^0 represents an initialized hidden state. In the ELAM, the nodes of x_0 and x_5 represent the subject entity and object entity, respectively, and x_r represents the root node.

2.1. Sequence Encoding Module

We define a sentence as $S = [x_1, x_2, x_3, \dots, x_n]$ with subject entity e_{subj} and object entity e_{obj} , where x_i is the i -th word and n is the length of the sentence.

First, we use GloVe [18] to map each word of the sentence to low-dimensional word vectors. The word embedding of the i -th word in S is denoted by $e_i^w \in R^{d^w}$, where d^w is the size of the word embeddings. Considering that the part of speech and the named entity recognition are the important features of each word or phrase in a sentence, we concatenate the word embedding, NER label embedding, and POS tag embedding of each word in a sentence. This approach can enrich the semantic features of each word or phrase in relation extraction models. Then, the representation of the i -th word is as follows:

$$e_i = [e_i^w, e_i^{pos}, e_i^{ner}], \tag{1}$$

where $e_i \in R^{d^w+d^p+d^n}$, d^w , d^p , and d^n denote the dimensions of the word, POS, and NER embeddings.

2.2. Self-Attention Enhanced Neural Network Module

This section introduces a self-attention enhanced neural network consisting of extended LSTM with a gate mechanism and a multi-head self-attention mechanism. Both

mechanisms are arranged in a parallel manner. We employ an extended LSTM to capture the long-distance dependency and the multi-head self-attention mechanism to concentrate on the importance and relevance of different words in a sentence. Finally, we combine the output of both modules to obtain semantic-guided contextual information as the input of the following layer.

Extended LSTM: we concatenate forward and reverse LSTM to encode the sentence features. This can efficiently capture long-distance semantic information. However, the input sentence S and previous state h_{prev} are independent and only interact in the LSTM. This model results in contextual information loss. Inspired by Gábor Melis et al. [19], we add a gate mechanism before the LSTM to afford a richer space of interaction between input S and hidden state h_{prev} . In the gate mechanism, h_{prev} and S interactions are regulated several times through a sigmoid gate. This mechanism reduces information loss during encoding, as shown in Figure 2. That is, we define the extended LSTM as $\widetilde{LSTM}(S, c_{prev}, h_{prev}) = LSTM(S \uparrow, c_{prev}, h_{prev}^\uparrow)$, where $S \uparrow$ and h_{prev}^\uparrow are defined as the highest-indexed S^i and h_{prev}^i , respectively. The formula is as follows:

$$S^i = 2\sigma(Q^i h^s i - 1_{prev}) \odot S^{i-2} \quad \text{for odd } i \in [1 \dots r], \tag{2}$$

$$h_{prev}^i = 2\sigma(R^i S^{i-1}) \odot h_{prev}^{i-2} \quad \text{for even } i \in [1 \dots r], \tag{3}$$

where Q^i and R^i are learnable weight matrices, h_{prev} is the initialization vector, and the number of rounds, $r \in N$, is a hyperparameter. Then, we feed the sentence S into the \widetilde{LSTM} to obtain contextual semantic representations:

$$h_t = \widetilde{LSTM}(S, c_{prev}, h_{prev}) \in R^{d_l}, \tag{4}$$

where d_l denotes the LSTM hidden dimension. After concatenating the forward and reverse \widetilde{LSTM} , we obtain the final hidden representation, as in Equation (4), and $\overleftrightarrow{h}_1, \overleftrightarrow{h}_t, \dots$, and \overleftrightarrow{h}_n as the output of the sequence encoding module, which obtains the semantic features:

$$\overleftrightarrow{h}_t = [\overrightarrow{h}_t, \overleftarrow{h}_t]. \tag{5}$$

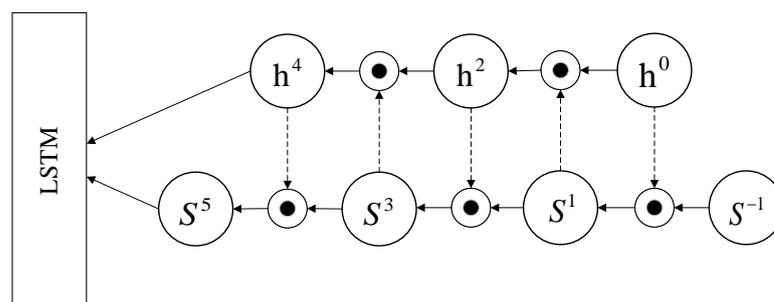


Figure 2. Gate mechanism of the extended LSTM. The previous state $h^0 = h_{prev}$ is transformed linearly, passing through the sigmoid and S^{-1} gates to produce S^1 , where S^{-1} is the representation of the input sentence S . After repeating this gating interaction five times, the final representation of sentence S^5 and the previous state h^4 are fed to the LSTM.

Multi-Head Self-Attention Mechanism: in a text sentence, each word has different importance, especially entity mentions. In semantic feature extraction processing, the relevance between different words affects the semantic information of entity mentions.

In order to reflect the relevance and importance of different words in a sentence, this paper uses a multi-head self-attention mechanism to calculate the correlations of each word. Transformer model [20] shows that the multi-head self-attention mechanism could obtain better results in sentence encoding by learning internal semantic features.

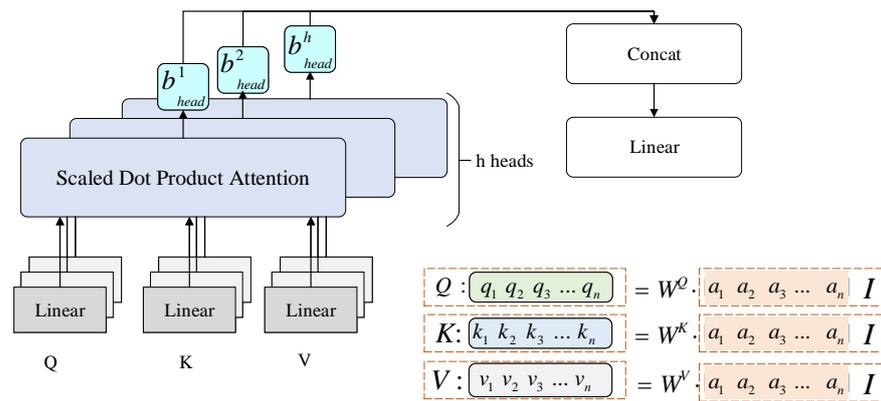


Figure 3. Multi-head self-attention module. The inputs $I = [a_1, a_2, a_3, \dots, a_n]$ are multiplied by the learnable matrices W^Q , W^K , and W^V to obtain the novel matrices Q , K , and V . Then, Q , K , and V are fed into a scaled dot-product attention to obtain the attention matrix b_i . The multi-head self-attention module performs this scaled dot-product attention h times parallelly, and concatenates each output b_i with linear transforming.

In this paper, we use scaled dot-product attention to calculate the attention weight. The input of the scaled dot-product attention consists of a query (Q), key (K), and value (V). Formally, after the encoding layer, the input representation $S = [e_1^w, e_2^w, \dots, e_n^w]$. We define $Q = K = V = S \in R^{n \times d_w}$. The hidden representation of a sentence obtained by self-attention is as follows:

$$H = Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_w}}V\right). \tag{6}$$

In the multi-head self-attention module, we linearly transform Q , K , and V before inputting them into the scaled dot-product attention, as shown in Figure 3. Instead of conducting a single self-attention, we perform it h times parallelly to jointly extract semantic features from different positions in a sentence. The i -th head is obtained from:

$$H_i = Attention(QW_i^Q, KW_i^K, VW_i^V) (i = 1, \dots, h), \tag{7}$$

where W_i^Q , W_i^K , and $W_i^V \in R^{(d \times d_m)}$ are learnable weight matrices; $d_m = m/h$. The multi-head attention module concatenates h outputs of each head’s self-attention operation. The output is denoted by:

$$A = MultiHead(Q, K, V) = W^R Concat(H_1, H_2, \dots, H_h), \tag{8}$$

where $W^R \in R^d \times d$ is a learnable weight matrix. The attention matrix A is the hidden representation of the input through the multi-head self-attention module.

We add a fully connected feed-forward network (FFN) to integrate the information extracted by the multi-head self-attention layer. The FFN consists of two linear transformations with a ReLU activation function between them. The feed-forward network is calculated as follows:

$$FFN(a_i) = \rho(a_i M_1 + \beta_1) M_2 + \beta_2, \tag{9}$$

where a_i is the output of the multi-head self-attention layer, M_1 and M_2 represent the linear transformation matrices, β_1 and β_2 are bias terms, d_q represents the dimension of the hidden layers, and ρ is the activation function (e.g., RELU). Inspired by Vaswani et al. [20], we employ layer normalization [21] and integrate the outputs of the multi-head self-attention layer and the FFN layer through a residual connection:

$$C = layerNorm(A' + FFN(A')), \tag{10}$$

where $A' = \text{LayerNorm}(A + S)$ represents the residual connection around the input sentence (S) embedding and the multi-head self-attention outputs (A). Finally, we employ a max pooling layer to obtain the final representation of S . The output is denoted by:

$$r = \text{Max}(C) = \text{Max}\{c_1, c_2, \dots, c_n\}. \quad (11)$$

2.3. Semantic Aggregation Module

Firstly, we introduce the convolutional graph network (GCN) [12] in this module. The GCN is an adaption of a convolution neural network for efficiently processing graph-structured data. Let graph $G = (V, E)$, where V represents a set of nodes and E represents a set of edges. The input of the GCN is an adjacency matrix A ; if there is an edge from node i to node j , define $A_{ij} = 1$. The convolution formula is as follows:

$$h_i^{(l)} = \rho\left(\sum_{j=1}^n A_{ij} W^{(l)} h_j^{l-1} + b^{(l)}\right), \quad (12)$$

where $h_i^{(l)}$ denotes the output vector of the i -th node after the l -th convolution operation layer. $W^{(l)}$ is a weight matrix and $b^{(l)}$ is a bias vector.

In the graph convolutional network, each convolutional layer operation fuses each node with the features of neighbor nodes. However, entities and textual contexts are considered equally important in this process. Inspired by Guo et al. [22], this paper proposes an entity-centric logical adjacency matrix (ELAM) to emphasize the impact of textual context on the entities in the dependency tree. We construct a dependency tree with entities as global nodes and add virtual edges between the entity nodes and other nodes. Then, we parse a sentence as graph-structured data on the relation extraction task through the dependency tree. The proposed model can fuse the semantic features of all nodes to the entity nodes with only a one-layer GCN convolutional operation. In addition, the information of the node itself in $h^{(l-1)}$ cannot be transmitted to $h^{(l)}$; thus, we add a self-loop for each node. The algorithm for constructing the ELAM is shown as Algorithm 1.

Algorithm 1 Construction of the entity-centric logical adjacency matrix (ELAM).

Input:

P: entity position in sentence; N: sentence length; S: target sequence.

Output:

Entity-centric logical adjacency matrix (ELAM);

- 1: Construct the sentence S as a dependency tree with entities as global nodes;
 - 2: Initialize the entity-centric logical adjacency matrix (ELAM) with all elements set to 0;
 $ELAM \in R^{N \times N}$
 - 3: Traverse all nodes of the subtree and calculate distance d between node i and root;
 - 4: To each node itself, add a self-citation and set $d = 1$;
 - 5: Set the value of the corresponding position in the matrix to $w(d)$ where
 $w(d) = \text{Weight}(d)$;
 - 6: **return** $ELAM$.
-

The $w(d)$ in Algorithm 1 represents the weight coefficient of the feature fusion between the nodes, which is calculated by the *Weight* function. The greater the weight, the shorter the distance between nodes and the richer the semantic information. We define the *Weight* function as:

$$\text{Weight}(e) = \frac{1}{e^{d-1}}, \quad (13)$$

where e is the Euler's number and d is the distance from the node to the entity. The further the distance, the less semantic information, and the lower the weight. Figure 4 illustrates the construction process of the entity-centric logical adjacency matrix.

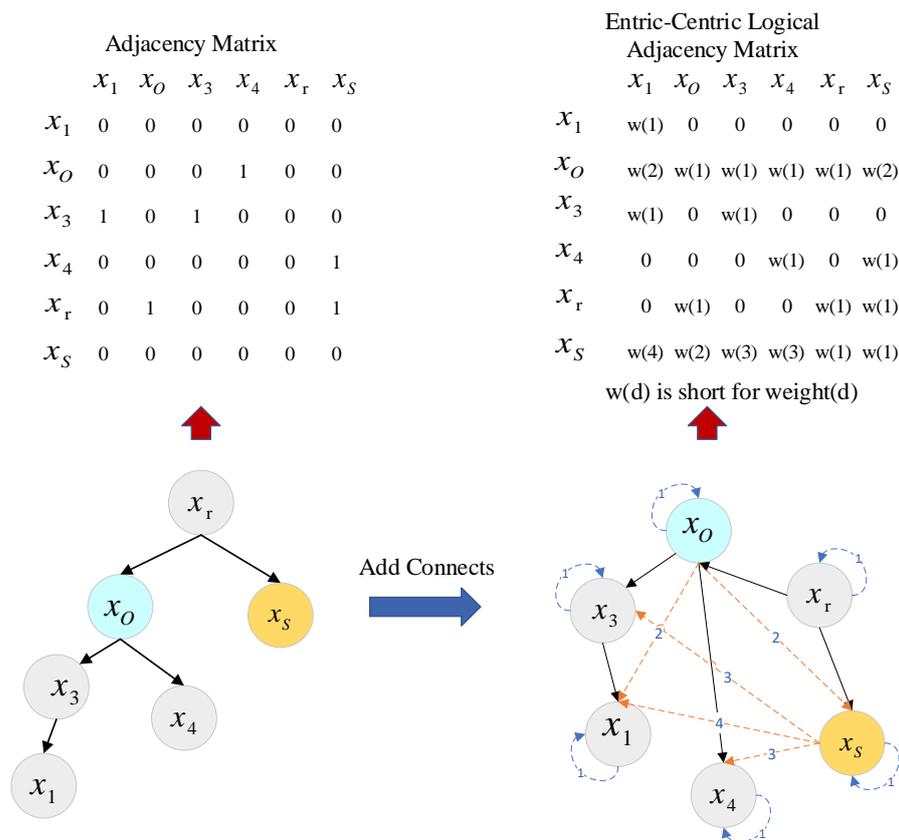


Figure 4. Construction process of the entity-centric logical adjacency matrix. The dashed lines represent the new connections between the entity nodes and the other established nodes, and the dashed line represents the self-loops of the nodes themselves. The number on the line represents the distance between the nodes. $w()$ is short for $weight()$ function. The nodes of x_O and x_S represent the subject entity and object entity, respectively.

This model has two advantages. First, it emphasizes the impact of the textual context on the semantic representation of the entities and uses enhanced semantic entity information to improve the accuracy of the relation extraction. Second, the entity-centric logical adjacency matrix can integrate k-order neighborhood information directly on a one-layer GCN and alleviate the tendency of over-smoothing in the multi-layer GCN calculation. Therefore, the paper modifies the convolution calculation as follows (Equation (14)):

$$h_i^{(l)} = \rho\left(\sum_{j=1}^n ELAM_{ij}W^{(l)}h_j^{l-1}/d_i + b^{(l)}\right) \in R^{d_w}, \tag{14}$$

where $h_1^0, h_2^0, \dots, h_n^0 = S = [e_1^w, e_2^w, \dots, e_n^w]$, d_i represents the out-degree of the node i , and d_w denotes the GCN hidden representation size; $W^{(l)} \in R^{2d_l \times d_w}$.

2.4. Relation Extraction Module

After the L-layer CEGCN calculation, the hidden representation of the sentence is $H^{(L)} = [h_1^{(L)}, h_2^{(L)}, \dots, h_n^{(L)}]$. This paper employs a *maxpool* function to reduce the hidden representation matrix from two dimensions to one dimension as d_w . The formula is as follows:

$$h = \maxpool[H^{(L)}]. \tag{15}$$

Embedding the semantic-guided contextual information into the subject and object entities can improve their association. To focus more semantic information of the textual

context on the subject entity and the object entity, we combine hidden representations of entities and the textual context in the relation extraction module. In this module, the semantic-guided contextual information from a sentence can be concentrated on the semantic entity representations. We feed the hidden representation into a *softmax* function to calculate the attention weight α . Then, the final entity representation is given by:

$$h_{entity} = \text{maxpool}(H_{entity}^{(L)}), \quad (16)$$

$$y = WH^{(L)}h_{entity} + b, \quad (17)$$

$$\alpha = \frac{\exp(y^L)}{\sum_{j=1}^L \exp(y^j)}, \quad (18)$$

$$h'_{entity} = \text{maxpool}(\alpha H_{entity}^{(L)}). \quad (19)$$

We believe modeling the association between the subject and object entities to be a significant factor in determining their relation. Lin et al. [23] propose that the entity relation r in a sentence is a subject entity to the object entity transformation ($e_{sub} + r = e_{obj}$). Their models have thoroughly employed and evaluated the difference vector of the entity pair to represent the relation between the entity pair and achieve good results. Therefore, we calculate the difference vector of the entity pair ($r = h_{sub} - h_{obj}$) as a part of the relation extraction constraint, where h_{sub} and h_{obj} are the entity vectors obtained through Equations (16)–(19). Then, we join the difference vector of the entity pair (r) and hidden layer output of the context (h_{text}) to obtain the final vector representation. The formula is as follows:

$$h_{out} = [h_{text}; r]. \quad (20)$$

Finally, this paper feeds the final vector representation into the feed-forward network (FFN) and obtains the probability distribution of the relation between entity pairs through the *softmax* function:

$$h_{final} = \text{FFN}(h_{out}), \quad (21)$$

$$p(y | h_{final}) = \text{softmax}(\text{MLP}(h_{final})) \in R^{|\mathcal{C}|}, \quad (22)$$

where $|\mathcal{C}|$ is the number of relation categories defined in the datasets. We train the model by back-propagation and employ the cross-entropy function as the loss function of the model. The cross-entropy function is defined as follows:

$$\text{Loss} = \sum_{i \in [1, L]} -\log P_{\theta}(c_i = C_i), \quad (23)$$

where c_i represents the predicted relation category and C_i represents the true relation category.

3. Experiment

3.1. Datasets

We evaluate the performance of our model on two popular relation extraction datasets: TACRED and SemEval-2010 Task 8.

TACRED: The TACRED dataset is a relation extraction dataset with 106,264 instances and 42 relation types (including 41 declared semantic relations and a “None” relation, which indicates that an entity pair has no defined relation) [17]. In the TACRED dataset, 79.5% of instances have been labeled as “no_relation”; the main predefined relations include “per:titled”, “org:employ_of”, “per:age”, “org:founded_by”, etc. Each TACRED instance is a sentence that contains an entity pair, 23 fine-grained types of entity mentions, and 1 of the 42 relation types. The type of entity mentions includes “organization”, “time”, “person”, etc.

SemEval-2010 Task 8: The SemEval-2010 Task 8 [16] dataset consists of 10,717 examples, 9 relation types, and a specific “other” type, which has been widely used in relation extrac-

tion tasks. In the SemEval-2010 Task 8 dataset, 17.6% of instances are labeled as “Other”; the main predefined relations include “Cause–Effect”, “Instrument–Agency”, “Entity–Destination”, etc. Each instance of this dataset contains two marked entities and the relation between the entity pair. The training set has 8000 instances, whereas the test set contains 2717 instances.

Based on these two datasets, we use a pre-trained 300-dimensional GloVe [18] vector to map each word of the sentence to word embeddings and initialize POS and NER label embeddings with a 30-dimension vector. The number of interactive computations in the gate mechanism is set to 5. The hidden GCN size is set to 200, the dropout rate is 0.5, and the prune $k = 1$ [12]. For the TACRED dataset, we set a learning rate of CEGCN 0.1 with a decay rate 0.95. For the SemEval-2010 Task 8 dataset, we set a learning rate of CEGCN 0.5 with a decay rate 0.9. We trained our model for 120 epochs on both datasets. We list the details of the hyperparameters of our model for both datasets in Table 1.

Table 1. Hyperparameters of the model for both datasets.

Parameters	Description	Value
d_w	word embedding	300
d_p	POS embedding	30
d_n	NER embedding	30
h_l	LSTM hidden size	100
h_g	GCN hidden size	200
d_l	CEGCN layers	3
h	attention heads	6
r	interaction rounds	5
batch	batch size	50

3.2. Performance Comparison

We use precision (P), recall (R), and F1 score (F1) to evaluate our model on the TACRED dataset and F1 score on the SemEval-2010 Task 8 dataset. For both datasets, we compare our model (CEGCN) against several competitive baselines, which contain logical regression models [24], sequence-based feature extraction models [8,25], the LSTM-based models [10,26], and graph-based models [27,28]. These baselines include the relation extraction model with a dependency tree as the input and the latest improved GCN models. To avoid effects from external enhancements, we do not employ BERT-based [29] models as the baseline.

The performance metrics of our model and all comparison models on the TACRED dataset are shown in Table 1. Four types of models are compared. (1) The logical regression (LR) models [24]: a traditional relation extraction model based on dependency trees combined with lexical information. (2) The CNN-based models [30]: these models use multi-window filters to capture the semantic features of sentences for relational extraction automatically. (3) The LSTM-based relation extraction models: these include the position-aware LSTM (PA-LSTM) [17] model, the tree-LSTM model [26], and the SDP-LSTM model [11]. The PA-LSTM model employs the position-aware attention mechanism combined with LSTM sequence encoder models. The SDP-LSTM model uses the shortest dependency path between the entity pair and the LSTM encoder. The tree-LSTM model encodes the entire tree structure to acquire the semantic information of words. (4) GCN-based relation extraction models: Zhang et al. [12] proposed the C-GCN model to apply a pruned dependency tree. The AGGCN model was proposed by Guo et al. [22] as a soft-pruning strategy based on the attention mechanism with the whole dependency tree as the GCN input. Chen et al. [27] proposed the DAGCN model, which automatically learned the neighbor importance of different points using multiple attentional components.

As shown in Table 2, we can observe that the F1 score of our model is significantly improved. Compared with other models, the precision of the CNN model achieves 75.6, but the lowest recall results in the lowest F1 score. We argue that the low recall score of

the CNN-based model is because the CNN tends to classify pre-defined relations precisely, producing the wrong prediction of undefined relation types. Moreover, compared with GCN-based models, the F1 score of our model improved by at least 0.4. In particular, compared with AGGCN, our model has a specific improvement in all three evaluation standards. The AGGCN takes the whole dependency tree as the input and employs an attention mechanism to guide the GCN. In contrast, our model concentrates on the important context rather than the whole text, improving the semantic-guided relation between entity pairs. We believe this is because our model focuses the contextual semantic information on the entity mentions in a sentence, enriching the semantic features of the entities and reducing the ambiguity of the entity mentions. The experimental results show the effectiveness of the model.

Table 2. Results on the TACRED dataset.

Model	P	R	F1
LR [24]	73.5	49.9	59.4
CNN [30]	75.6	47.5	58.3
SDP-LSTM [11]	66.3	52.7	58.7
PA-LSTM [17]	66.0	59.2	62.4
C-GCN [12]	65.7	63.3	66.4
AGGCN [22]	69.9	60.9	65.1
DAGCN [27]	70.1	63.5	66.8
CEGCN (our model)	73.4	61.8	67.2

In addition, we conducted validation experiments on the SemEval-2010 Task 8 dataset to assess the versatility of our model. As indicated in Table 3, we conducted validation experiments on some relevant dependency models. The SDP-LSTM model calculates the shortest path to the common ancestor in the dependency tree, but this only focuses on the part of the information between entities, ignoring the important words in context. The F1 score of our model is 1.7 points higher than that of SDP-LSTM. By observing the experimental results, we find our model improved the F1 score by at least 0.4, compared to other GCN-based models. Compared with the latest C-MDR-GCN model, our model could focus on essential words in context to obtain a higher F1 score. The proposed model can achieve an 86.1 F1 score and thereby outperform other models.

Table 3. Results on the Semeval-2010 Task 8 dataset.

Model	F1
LR [24]	82.2
CNN [30]	83.7
SDP-LSTM [11]	84.4
PA-LSTM (2017) [17]	84.8
C-GCN (2018) [12]	84.8
C-AGGCN (2020) [22]	85.7
C-MDR-GCN (2021) [31]	84.9
CEGCN (our model)	86.1

3.3. Ablation Study

To demonstrate the contribution of each module in the proposed framework, we perform ablation experiments on the TACRED dataset and adopt the F1 score as the standard. The results of the ablation experiments are shown in Table 4. Based on the proposed model, we introduce three different ablation models, which are described below:

- “CEGCN w/o Entity” means that we mask the entities with random tokens in the proposed model;
- “CEGCN w/o Self-Attention Enhanced NN” means that the self-attention enhanced neural network is removed;

- “CEGCN w/o ELAM” means that the entity-centric logical adjacency matrix is replaced by the ordinary adjacency matrix.

Table 4. Results on the Semeval-2010 Task 8 dataset.

Model	Dev F1
CEGCN (our model)	67.2
CEGCN w/o Entity	65.4
CEGCN w/o Self-Attention Enhanced NN	66.3
CEGCN w/o ELAM	66.5

Figure 5 indicates that the performance of the proposed model significantly drops when removing different modules. We can observe that, compared with CEGCN, the performance of the CEGCN w/o Entity decreases by 1.8. This indicates that the entities are crucial in the model. Experiments demonstrate that entity mentions obtain essential semantic information, which is necessary for relation extraction. When we remove the self-attention enhanced neural network, the performance of the CEGCN w/o Self-Attention Enhanced NN decreases by 1.0. This demonstrates the effectiveness of the self-attention enhanced neural network module. This module can obtain the semantic information of relevance and importance between different words to enrich the contextual dependencies. When we replace the entity-centric logical adjacency matrix with an ordinary adjacency matrix, the F1 score of CEGCN w/o ELAM decreases from 67.2 to 66.5. This proves that the effectiveness of ELAM can focus the semantic-guided contextual information on entities to improve the accuracy of the relation extraction. The convergence results of different models are shown in Figure 6. The smaller the train_loss, the more accurate the prediction result. The CEGCN model converges faster and obtains a lower train_loss than variant ablation study models.

Effect of Mask-Entity. Figure 5 indicates that the performance of the proposed model with masking entities is lower than without masking entities under each epoch. We can also observe that, in Figure 6, CEGCN w/o Entity converges slowly and obtains a higher train_loss. This demonstrates that entity mentions obtain essential semantic information. Enhancing the semantic representations of entities is crucial for relation extraction.

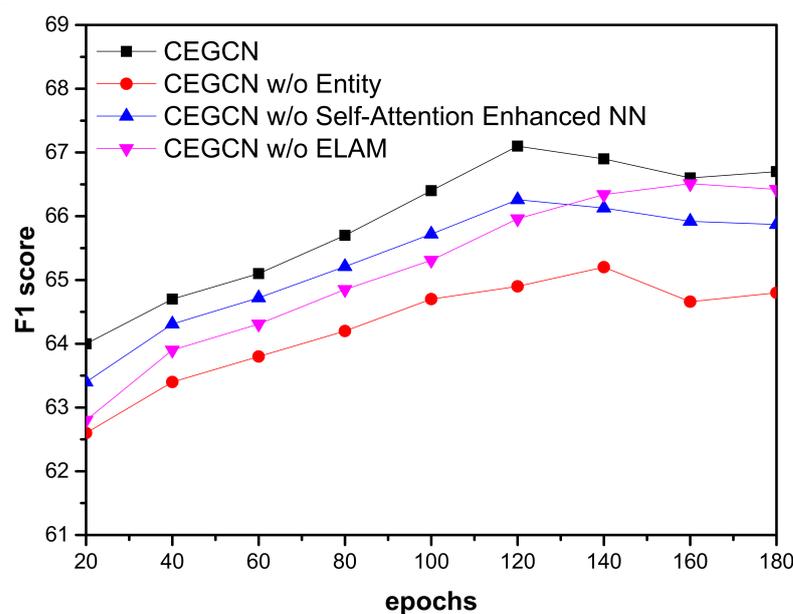


Figure 5. Experimental results in terms of F1 under different epochs for variant models of the ablation study.

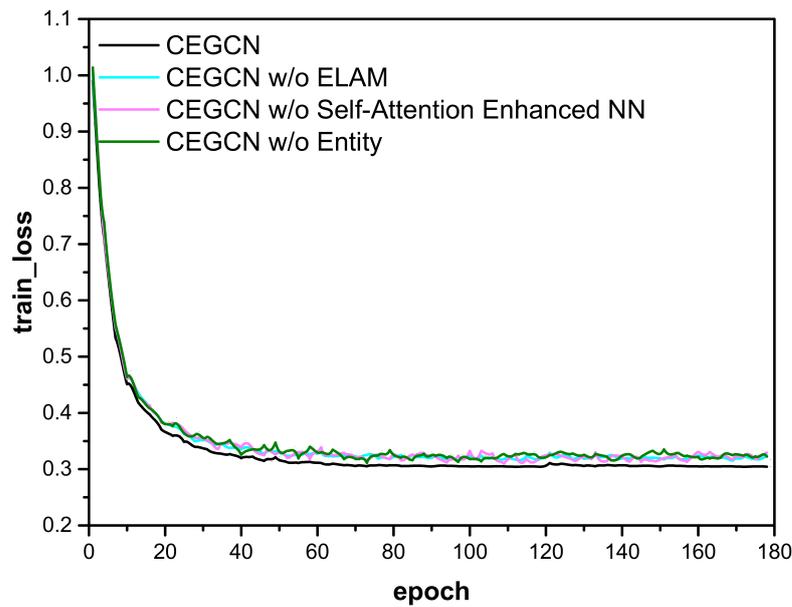


Figure 6. The train_loss for variant models of the ablation study.

Analysis of LSTM, Self-Attention, and GCN. Most natural language processing models based on deep learning use LSTM to obtain semantic information. The LSTM can capture the long-distance semantic information and enables each word to obtain the semantic features of the context. However, the input and previous state are independent and only interact in the LSTM, resulting in contextual information loss. In this model, we use a gate mechanism to solve this problem. The self-attention mechanism can help concentrate more on the relevance and importance of different words in a sentence to highlight the semantic information of key words. By combining them, we can obtain semantic-guided contextual information. Figure 7 indicates that the self-attention enhanced model can concentrate more on phrases containing predicates in different sentences; these context fragments are strong semantic-guided relations, such as “quit and later founded the hedge” in S1.

Person/org:founded_by/Organization



Figure 7. Self-attention weight distribution visualization. “Person/org:founded_by/Organization” means all sentences contain the same entity types (Person, Organization) and the same relation type (org:founded_by). The color depth expresses the degree of the attention weight distribution of the different text sequences to demonstrate the effectiveness of the self-attention enhanced model. The darker context fragments contain more important semantic information for relation.

The novel GCN models allow each word to capture the information of its dependent words directly. Focusing semantic-guided contextual information on entities can improve the representation of the relation between entities; these are complementary effects of LSTM, the self-attention mechanism, and GCN. Table 4 indicates that all three modules contribute the F1 score to the proposed model. Combining LSTM, the self-attention mechanism, and GCN enriches the representations of entities with semantic-guided information to obtain a more accurate relation between entities. Moreover, the entity-centric logical adjacency matrix enables entities to aggregate the semantic features of all nodes with a one-layer GCN. Furthermore, considering the distance of different words to entities, it calculates a fusion weight coefficient for each word to the entity; it can fuse the relevant information of the words and improve the accuracy of relation extractions.

Effect of ELAM. In our research, we insist that the entity-centric logical adjacency matrix can enrich the semantic representations of entities to improve the performance of our model. To demonstrate the effectiveness of ELAM in relation extraction tasks, we replace it with an ordinary adjacency matrix in the proposed model. We compare the F1 score and train_loss of them under different epochs. Figure 5 indicates that the CEGCN outperforms the CEGCN w/o ELAM by at least 0.7 F1 scores and reaches a peak around the 120th epoch in terms of the final F1 score. Figure 7 indicates that the self-attention enhanced model can improve the weight of important phrases in feature extraction; it can improve the semantic impact on the relation representation between entity pairs in the convolution operation. Moreover, our model converged quicker than the CEGCN w/o ELAM, as shown in Figure 6. The above has proved that ELAM can effectively aggregate the semantic-guided contextual information on entities and obtain better results in relation extraction tasks.

3.4. Effect of Hyper-Parameters

This paper introduces some hyperparameters to improve model performance. Compared with the other hyperparameters, the number of attention heads h and rounds r has a more significant impact on model performance. This section discusses the influence of two hyperparameters that affect model performance through experiments, namely, the number of attention heads h and the number of interaction rounds r in the gate mechanism of the extended LSTM.

The multi-head attention mechanism can reflect the relevance and importance of different words in a sentence. It is of great significance to select the correct number of heads for model improvement. Figure 8 shows that the model achieves its optimal performance at six heads, and the performance degrades with each additional head when using over six heads. Then, we study the number of interaction rounds r in the gate mechanism of the extended LSTM. Extended LSTM with a gate mechanism can afford more space for modeling the long-distance dependency feature, and can reduce information loss during encoding. Choosing the different numbers of r affects the model performance. In Figure 8, the comparison shows that the performance of the CEGCN model is relatively close when the r is set to 4 or 5, that the model obtains the highest score when the r is set to 5, and that the F1 score decreases when it exceeds 5.

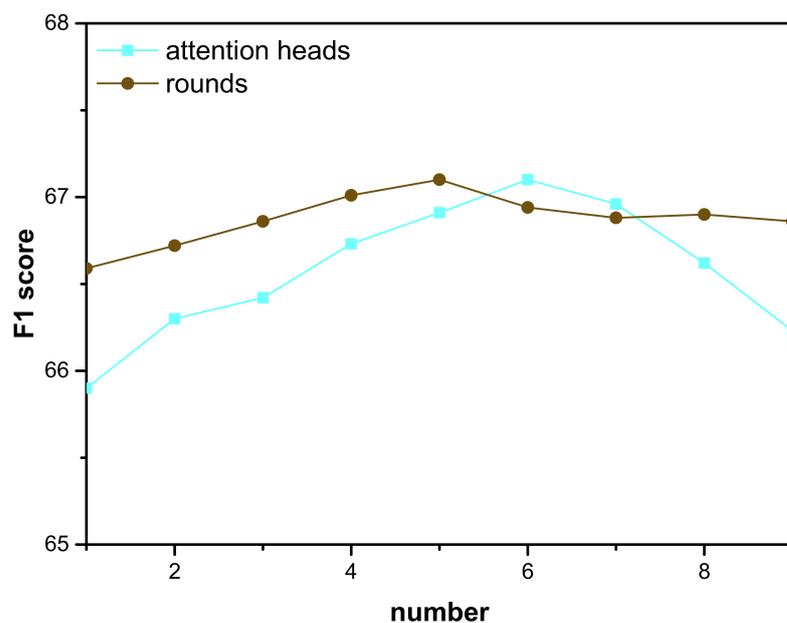


Figure 8. Experimental results on different numbers of attention heads h and different rounds r in extended LSTM.

4. Related Work

Traditional relation extraction tasks are based on feature extractors and rely on semantic features obtained from lexical resources. With the popularity of deep learning, deep learning models have been widely used in many research areas, such as intelligent Q&A systems [32], pattern recognition [33], and intelligent transportation systems [34]. In recent years, researchers mainly employed deep neural network models for relation extraction tasks [35]. Compared with classical machine learning models, the deep-learning-based models can automatically extract and learn from sentence features without complex feature extractors.

Initially, scholars tended to exploit CNN, RNN, and their improved deep learning models for relation extraction tasks. Zeng et al. [8] employed CNN to extract word-level and sentence-level features and took all of the per-trained word tokens as the input. Xu et al. [25] proposed a CNN model based on the dependency tree, parsing the sentence with a dependency tree as the input. Traditional RNNs have difficulty addressing long-term dependence; LSTM can solve this problem by adding a cell state, and gated operations can afford a richer space of interaction for the RNN. Xu et al. [11] proposed SDP-LSTM to obtain structure information through the shortest path between entities.

Dependency trees can convert text inputs into graph-structured data, and CNN and RNN models cannot efficiently process these data parallelly. Kipf and Welling et al. [12] proposed a graph convolutional network for supervised learning on graph-structured data. Hong et al. [28] proposed a relation-aware attention GCN for end-to-end relation extraction. Huang et al. [35] employed a GCN and knowledge graph enhanced transformer encoder for measuring semantic similarity between sentences and relation types. Guo et al. [22] proposed using soft attention to prune unimportant edges in the graph data dynamically. Huang et al. [36] proposed a knowledge-aware framework to highlight the keyword and relation clues and employed GCN for relation extraction. Our model exploits the advantages of GCN and enables entities to aggregate contextual semantic information with a one-layer GCN calculation.

5. Conclusions

This paper proposes a novel contextual semantic-guided entity-centric GCN model for relation extraction (CEGCN). This model combines the semantic information of relevance

and importance between different words to obtain semantic-guided contextual information. To enable entity aggregate semantic-guided contextual information, we construct a dependency tree with entities as global nodes and connect global nodes directly with other nodes. It can aggregate information from the whole tree with only a one-layer GCN calculation. In addition, our model can combine the semantic representations of the text sequence and the difference vectors of entities to constrain the relation between the entity pair, improving its performance. The experimental results on the TACRED and SemEval-2010 Task 8 datasets illustrate that this model enables the entities to obtain the semantic-guided contextual information to reduce the ambiguity of entity mentions in a sentence and outperform previous models. Finally, we find that the extended LSTM with a gate mechanism can effectively reduce information loss and complement GCN and multi-head self-attention in capturing semantic features.

Author Contributions: Conceptualization: J.L. and L.L.; experimentation and data analysis: L.L.; writing—original draft preparation: L.L.; writing—review and editing: L.L., H.F., Y.X., and H.L.; funding acquisition: W.H. and L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Joint Funds of the National Natural Science Foundation of China, under Grant No. U2003208, the National Natural Science Foundation of China, Grant No. 62177014, the Open Research Projects of Zhejiang Lab (Grant No. 2022KG0AB01), and the National Natural Science Foundation of China, under Grant No. 62172451.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural network
LSTM	Long short-term memory
GCN	Graph convolutional network
RNN	Recurrent neural network

References

1. Fader, A.; Soderland, S.; Etzioni, O. Identifying relations for open information extraction. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July 2011; pp. 1535–1545.
2. Hobbs, J.R.; Riloff, E. Information Extraction. In *Handbook of Natural Language Processing*; Chapman & Hall/CRC Press: London, UK, 2010.
3. Aviv, R.; Erlich, Z.; Ravid, G.; Geva, A. Network analysis of knowledge construction in asynchronous learning networks. *J. Asynchronous Learn. Netw.* **2003**, *7*, 1–23. [[CrossRef](#)]
4. Chen, Z.; Li, H.; Kong, S.C.; Xu, Q. An analytic knowledge network process for construction entrepreneurship education. *J. Manag. Dev.* **2006**, *25*, 11–27. [[CrossRef](#)]
5. Yih, S.W.t.; Chang, M.W.; He, X.; Gao, J. Semantic parsing via staged query graph generation: Question answering with knowledge base. In Proceedings of the Joint Conference of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015.
6. Dong, J.; Wu, R.; Pan, Y. A Low-Profile Broadband Metasurface Antenna with Polarization Conversion Based on Characteristic Mode Analysis. *Front. Phys.* **2022**, *10*, 860606. [[CrossRef](#)]
7. Hashimoto, K.; Miwa, M.; Tsuruoka, Y.; Chikayama, T. Simple customization of recursive neural networks for semantic relation classification. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1372–1376.
8. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation classification via convolutional deep neural network. In Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014), Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
9. Dong, J.; Qin, W.; Wang, M. Fast multi-objective optimization of multi-parameter antenna structures based on improved BPNN surrogate model. *IEEE Access* **2019**, *7*, 77692–77701. [[CrossRef](#)]

10. 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; pp. 207–212.
11. 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 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, 17–21 September 2015; pp. 1785–1794.
12. Zhang, Y.; Qi, P.; Manning, C.D. Graph convolution over pruned dependency trees improves relation extraction. *arXiv* **2018**, arXiv:1809.10185.
13. Chen, Z.M.; Wei, X.S.; Wang, P.; Guo, Y. Multi-label image recognition with graph convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5177–5186.
14. Li, L.; Gan, Z.; Cheng, Y.; Liu, J. Relation-aware graph attention network for visual question answering. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 10313–10322.
15. Babič, M.; Mihelič, J.; Cali, M. Complex network characterization using graph theory and fractal geometry: The case study of lung cancer DNA sequences. *Appl. Sci.* **2020**, *10*, 3037. [[CrossRef](#)]
16. Hendrickx, I.; Kim, S.N.; Kozareva, Z.; Nakov, P.; Séaghdha, D.O.; Padó, S.; Pennacchiotti, M.; Romano, L.; Szapkowicz, S. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv* **2019**, arXiv:1911.10422.
17. Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; Manning, C.D. Position-aware attention and supervised data improve slot filling. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017.
18. Santoro, A.; Raposo, D.; Barrett, D.G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; Lillicrap, T. A simple neural network module for relational reasoning. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
19. Melis, G.; Kočiský, T.; Blunsom, P. Mogrifier LSTM. *arXiv* **2019**, arXiv:1909.01792.
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
21. Kim, M.; Park, S.; Lee, W. A robust energy saving data dissemination protocol for IoT-WSNs. *KSII Trans. Internet Inf. Syst.* **2018**, *12*, 5744–5764.
22. Guo, Z.; Zhang, Y.; Lu, W. Attention guided graph convolutional networks for relation extraction. *arXiv* **2019**, arXiv:1906.07510.
23. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
24. Tsukimoto, H. Logical regression analysis: From mathematical formulas to linguistic rules. In *Foundations and Advances in Data Mining*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 21–61.
25. Xu, K.; Feng, Y.; Huang, S.; Zhao, D. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv* **2015**, arXiv:1506.07650.
26. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv* **2015**, arXiv:1503.00075.
27. Chen, F.; Pan, S.; Jiang, J.; Huo, H.; Long, G. DAGCN: Dual Attention Graph Convolutional Networks. In Proceedings of the International Joint Conference on Neural Networks (IJCNN 2019), Budapest, Hungary, 14–19 July 2019.
28. Hong, Y.; Liu, Y.; Yang, S.; Zhang, K.; Wen, A.; Hu, J. Improving graph convolutional networks based on relation-aware attention for end-to-end relation extraction. *IEEE Access* **2020**, *8*, 51315–51323. [[CrossRef](#)]
29. Shi, P.; Lin, J. Simple bert models for relation extraction and semantic role labeling. *arXiv* **2019**, arXiv:1904.05255.
30. Nguyen, T.H.; Grishman, R. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Denver, CO, USA, 5 June 2015; pp. 39–48.
31. Hu, Y.; Shen, H.; Liu, W.; Min, F.; Qiao, X.; Jin, K. A Graph Convolutional Network With Multiple Dependency Representations for Relation Extraction. *IEEE Access* **2021**, *9*, 81575–81587. [[CrossRef](#)]
32. Yu, M.; Yin, W.; Hasan, K.S.; Santos, C.D.; Xiang, B.; Zhou, B. Improved neural relation detection for knowledge base question answering. *arXiv* **2017**, arXiv:1704.06194.
33. Kong, Q.; Cao, Y.; Iqbal, T.; Wang, Y.; Wang, W.; Plumbley, M.D. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 2880–2894. [[CrossRef](#)]
34. Shi, H.; Zhao, X.; Wan, H.; Wang, H.; Dong, J.; Tang, K.; Liu, A. Multi-model induced network for participatory-sensing-based classification tasks in intelligent and connected transportation systems. *Comput. Networks* **2018**, *141*, 157–165. [[CrossRef](#)]
35. 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](#)]
36. Huang, W.; Mao, Y.; Yang, L.; Yang, Z.; Long, J. Local-to-global GCN with knowledge-aware representation for distantly supervised relation extraction. *Knowl.-Based Syst.* **2021**, *234*, 107565. [[CrossRef](#)]