electronics

Article

Construction of an Event Knowledge Graph Based on a Dynamic Resource Scheduling Optimization Algorithm and Semantic Graph Convolutional Neural Networks

Xing Liu [1,2], Long Zhang [1,2,*], Qiusheng Zheng [1,2], Fupeng Wei [3,*], Kezheng Wang [4], Zheng Zhang [5], Ziwei Chen [1,2], Liyue Niu [1,2] and Jizong Liu [1,2]

1 The Frontier Information Technology Research Institute, Zhongyuan University of Technology, Zhengzhou 450007, China; xl@zut.edu.cn (X.L.); czw@zut.edu.cn (Z.C.); 9715@zut.edu.cn (L.N.); 9935@zut.edu.cn (J.L.)
2 Henan Key Laboratory on Public Opinion Intelligent Analysis, Zhengzhou 450007, China
3 School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China
4 Lenovo Group Solution Center, Beijing 100085, China; wangkz1@lenovo.com
5 School of Computer and Software, Nanyang Institute of Technology, Nanyang 473004, China; zhangzheng@nyist.edu.cn
* Correspondence: zhanglong@zut.edu.cn (L.Z.); weifupeng@ncwu.edu.cn (F.W.)

Abstract: Presently, road and traffic control construction on most university campuses cannot keep up with the growth of the universities. Campus roads are not very wide, crossings do not have lights, and there are no full-time traffic management personnel. Teachers and students are prone to forming a peak flow of people when going to and from classes. This has led to a constant stream of traffic accidents. It is critical to conduct a comprehensive analysis of this issue by utilizing voluminous data pertaining to school traffic incidents in order to safeguard the lives of faculty and students. In the case of domestic universities, fewer studies have studied knowledge graph construction methods for traffic safety incidents. In event knowledge graph construction, the reasonable release and recycling of computational resources are inefficient, and existing entity–relationship joint extraction methods are unable to deal with ternary overlapping and entity boundary ambiguity problems in relationship extraction. In response to the above problems, this paper proposes a knowledge graph construction method for university on-campus traffic safety events with improved dynamic resource scheduling algorithms and multi-layer semantic graph convolutional neural networks. The experiment's results show that the proposed dynamic computational resource scheduling method increases GPU and CPU use by 25% and 9%. On the public dataset, the proposed data extraction model's F1 scores for event triples increase by 1.3% on the NYT dataset and by 0.4% on the WebNLG dataset. This method can help the relevant university personnel in dealing with unexpected traffic incidents and reduce the impact on public opinion.

Keywords: public opinion; knowledge graph; graph convolutional neural network; resource scheduling; traffic safety

Citation: Liu, X.; Zhang, L.; Zheng, Q.; Wei, F.; Wang, K.; Zhang, Z.; Chen, Z.; Niu, L.; Liu, J. Construction of an Event Knowledge Graph Based on a Dynamic Resource Scheduling Optimization Algorithm and Semantic Graph Convolutional Neural Networks. *Electronics* **2024**, *13*, 11. https://doi.org/10.3390/electronics13010011

Academic Editor: Fabio Grandi

Received: 10 November 2023
Revised: 29 November 2023
Accepted: 5 December 2023
Published: 19 December 2023

1. Introduction

Along with the increasing number of students and faculty in schools at home and abroad, schools have entered a period of rapid development. In the case of domestic universities, the expansion in the number of university teachers and students also introduces complex traffic safety risks [1–4]. If there is a traffic accident, it will not only hurt people, but also make it harder to control online public opinion [5]. Traffic safety events happen quickly, and the approaches by which they are managed are marked by diversity, multifactoriality, and variability. Traditional decision makers may be hampered by their reliance on prior emergency response experience, poor timeliness, and insufficient understanding of theoretical decision making. Therefore, responding to campus traffic emergencies with conventional management techniques is a challenge. There has been an exponential increase in current news and information regarding traffic safety incidents that occur on university campuses [6–8]. To improve the intelligent emergency management of on-campus traffic safety incidents at universities, it is important to know how to look at the issue using a lot of unstructured on-campus traffic safety event data. Intelligent decision management is important to ensure the life safety of students and teachers at universities [9].

In university on-campus traffic safety events, knowledge graphs are a significant source of technical support for fusing heterogeneous data from numerous sources. A knowledge graph is essentially a heterogeneous network graph composed of entity nodes and edges, which can show the complex relationship between entities and exhibit rich semantic structure information [10,11]. Knowledge graph application scenarios mainly include user profiles, intelligent retrieval, event-assisted decision making, and inference. The problem of semantic ambiguity and information irregularity has been a difficult task to overcome in the past in the semantic retrieval of campus traffic safety events and incident decision making. However, knowledge graphs for the construction of domain knowledge and the representation of graph learning are effective solutions to the aforementioned issues. Recently, academics have focused more on modeling knowledge graph representations of emergency traffic events and building correlation links between events. For example, Zhu et al. target the limited intelligence and accuracy of current emergency event evolution prediction algorithms. They proposed a method for anticipating municipal rail transit emergencies using knowledge graphs and relational graph neural networks, which ultimately supports rail transit emergency management decision making [12]. Sun et al. targeted the fact that existing methods have difficulty recognizing traffic events and are ineffective at extracting event-associated features. They proposed that the traffic knowledge graph and target detection method be used to recognize traffic events on Weibo. This method can help a city's traffic management department identify traffic problems and quickly inform people about them. It can also help people make decisions [13].

The current project of constructing a knowledge graph for university traffic safety events has three problems: (a) Knowledge graph construction requires many modeling algorithms and a large amount of computational resources [14]. Data capture and entity–relationship co-extraction model training require more memory and computation. How to understand the variation in the demand for resources during the construction of a knowledge graph of on-campus traffic safety events, while achieving the reasonable release and recycling of resources, poses a new challenge. (b) Little research has been conducted on how to construct knowledge graphs for events that promote traffic safety on university campuses. Very few studies have been conducted on how to make ontological models for knowledge graphs of on-campus traffic events. This makes it difficult to construct knowledge graphs of on-campus traffic events [15]. (c) The core contents of knowledge graph construction of on-campus traffic safety events in higher education are entity–relationship triples. Existing methods for extracting both entities and relationships do not perform well at dealing with triple overlap problems and the blurring of entity boundaries in relationship extraction. This can lead to errors in the extraction process, redundant knowledge information, and other problems.

In dealing with the above problems, the three main parts of the innovations we propose include the following:

1. Targeting the dynamic demand of resources for multi-scenario tasks during knowledge graph construction. For problem (a), we propose a resource management and scheduling technique based on virtualization technology using Kernel-based Virtual Machine–Quick Emulator (KVM-QEMU) and Kubernetes technology. KVM-QEMU virtualization technology enables the virtualization of hardware resources and improves the efficiency of resource utilization. Kubernetes container orchestration technology has a united scheduling feature that can use Graphics Processing Unit (GPU) resources at the same time to make scheduling work well with different types

of resources, like Central Processing Unit (CPU) and GPU. The proposed dynamic computational resource scheduling method increases GPU and CPU use by 25% and 9%.

2. To improve the accuracy and standardization of information extracted from online social media for on-campus traffic safety events, we propose a general ontology model for knowledge graphs of traffic safety events in universities to solve problem (b). The model can provide standard definitions and construct multi-source knowledge structures so that knowledge graphs regarding traffic safety events in universities can be made. In addition, the methodology takes full account of the dynamic spatial and temporal information between events. We construct a knowledge graph ontology model of university traffic events based on multi-source heterogeneous data.
3. To adapt to the needs of online public opinion scenarios, for problem (c), we propose a joint extraction method for entity relations based on graph convolutional neural networks. The method first fuses the global semantic dependency analysis graph embedding information with syntactic analysis graph embedding information to further improve the accuracy of the recognition of distant entities. Next, a multi-layer semantic graph convolutional neural network is constructed to find deeper, semantically hidden knowledge about how entities are related. Finally, a multi-feature fusion attention mechanism is designed to enhance the accuracy of model triple classification. The method effectively enhances the problem of entity boundary ambiguity in triple overlapping and relation extraction. Among them, the F1 values were improved by 1.3% and 0.4% on the NYT and WebNLG English datasets, respectively.

2. Related Work

2.1. Virtualization Management and Scheduling Technology

Virtualization, as a method of provisioning resources, abstracts and transforms various physical resources (Servers, Storage, Networks, etc.). To run numerous virtual machines on physical servers, we employ KVM-QEMU virtualization technology and Kubernetes container technology [16]. These servers then help with big computer jobs like graph computation and knowledge triple extraction [17]. Among them, KVM-QEMU is a kernel-level virtualization technology that can better utilize different operating systems and hardware resources effectively and improve the performance of hardware virtualization. Libvirt is designed as a driver-oriented architecture consisting of Libvirtd services, API libraries, and Virsh command-line management tools. It enables effective management of different types of VMs by calling idle API libraries, and different types of VMs are logically separated from each other. Kubernetes container technology is a container orchestration platform for scheduling, deploying, and managing containers, among other things [18]. Kubernetes container technology facilitates the detection and allocation of GPU resources. The hot-plugging feature of kubernetes is used to access and manage GPU resources using a plug-in extension mechanism. Then, kubernetes technology is used to determine the scheduling result of the pod based on the user's GPU request and finally allocate computer storage resources to it.

2.2. Ontology Modeling Methodology

In recent years, there has been an increase in the number of studies conducted by both domestic and foreign researchers, with the aim of constructing a knowledge graph of security incidents. The event knowledge graph application perspective encompasses several types of events, such as aviation security events, flooding events, city train traffic events, and financial emergencies. Knowledge graphs can help improve the quality of ontology construction by providing basic information at the data layer for ontology construction. However, there are fewer studies on the construction of knowledge graph ontologies for traffic safety events in universities. Constructing ontologies is used a lot in fields like knowledge graphs, information retrieval, and more. Based on what they cover, ontology model construction can be divided into two groups: general ontology construction and domain ontology construction [19,20]. Previous studies have mostly focused on domain knowledge graph ontology modeling, so this paper proposes a way for constructing a domain ontology. Two basic methods of domain ontology construction exist: human construction using domain expert knowledge and automatic or semi-automatic construction using AI techniques. Wang et al. utilized ontology technology to extract, structure, and depict textual knowledge related to fire emergency management, with the aim of assisting in fire emergency management. In turn, an ontology model for fire emergency management, Fire Emergency Ontology (FEO), is constructed. Intelligent management and fire emergency response speed are improved by the model [21].

2.3. Entity Relationship Joint Extraction Model

The role of entity–relationship co-extraction is to recognize entities and correspondences in a particular text. The traditional pipeline approach has problems extracting long-distance relational dependencies between entities and attenuating feature information between subtasks, resulting in redundant entities and not enough extracted relationships. This study focuses on exploring the prevailing deep learning approaches used to simultaneously extract entities and relations [22–24]. The traditional pipeline approach shortcomings can be solved this way. Zhu et al. target existing approaches to problems with nested entities and overlapping relationships. He proposed a single-stage joint entity relationship extraction method based on an enhanced sequence labeling strategy. The method improved NYT and WebNLG $F1$ values by 0.5–2.1% [25]. Clara Vania et al. presented a systematic study on the use of natural language inference to improve document-level relational extraction with distant supervision. This method reduces pricey annotations. Results show that the natural language inference filtering method improves relational extraction. This method reduces the $F1$ value metric gap by 2.3%, as compared to model training on a manually labeled dataset [26]. Document-level relationship extraction methods by Ding et al. do not properly incorporate sentence context, document topic, and entity pair similarity when generating nodes, resulting in low performance. She proposed a document-level relationship extraction model for enhanced entity representation, which is implemented better than existing models [27]. This experiment enhances the latest model by utilizing a graph-based convolutional neural network, resulting in an improvement in the extraction accuracy of triple groups.

3. Methodology

The technical flowchart for the construction of the knowledge graph of traffic safety events in university is shown in Appendix A.1.

This part deals with the enhancement of three modules: the Dynamic Computing Resource Algorithm Optimization Design Module, the Event Ontology Design Module, and the Event Information Extraction Design Module. The strategies for improvement are outlined as follows.

3.1. Virtualized Resource Management and Scheduling Design

The section on the Virtualized Resources and Scheduling Technology Architecture flowchart is shown in Appendix A.2.

Existing methods have not been able to effectively use GPU and CPU computing resources. We virtualize hardware using KVM-QEMU and kubernetes to optimize computing resources for model training.

The original kubernetes default scheduling algorithm leads to a balanced utilization of GPU and CPU resources that needs to be improved [28]. It does not accurately reflect node resource utilization, and the final node score does not account for how resource usage affects scheduling priority. To solve this problem, this paper proposes an improved dynamic resource scheduling method. In a kubernetes cluster, the yaml file for every pod defines the resource allocation. This study classifies pods as GPU- and CPU-consuming, based on the GPU and CPU usage of resources.

As an example, the resource e(R_e) on a node that is consumed by this pod is calculated as follows Equation (1):

$$R_e = \frac{E_e}{T_e} \tag{1}$$

where E_e is the pod's request for resource $e(R_e)$, T_e is the total amount of resources on a particular node e. Resource $e(R_e)$ includes both GPU and CPU resources, and R_e yields the percentage of consumption of GPU or CPU resources.

The node with a high score is selected as the deployment node of the pod, and the formula is as follows Equations (2) and (3):

$$\alpha + \beta = 1 \tag{2}$$

$$Score = 10 - |\alpha U_{gpu} - \beta U_{cpu}| * 10 \tag{3}$$

where U_{gpu} and U_{cpu} represent the average utilization of the node's GPU and CPU, respectively, over a certain period of time. Equations (2) and (3) adjust the resource load usage of a node by introducing the weight parameters α and β. To reflect the GPU and CPU resource loads of this node for different consumption types of pods. When *Score* is larger, it shows that the difference between the GPU and CPU use of the node for the pod is smaller. After deploying the pod, the node's GPU and CPU use balance, so the pod's priority increases.

3.2. Ontology Modeling Design

In this study, the ontology model is used to construct a knowledge graph of traffic safety events in universities [29–31]. This is accomplished by combining the distinct characteristics of the information about these events. Consider the elements involved in a traffic incident as an organic, dynamic whole. A top-down method is also used to construct a conceptual model for the knowledge graph of traffic safety events on university campuses [32]. The knowledge graph ontology of university traffic safety events includes entity type $O(E)$, attribute type $O(S)$ and relationship type $O(R)$, as shown in Equation (4).

$$H_{KG} = \{O(E), O(R), O(S)\} \tag{4}$$

The ontology for traffic safety events at the university is constructed as shown in Figure 1. It shows a knowledge graph of university traffic safety events. It is made up of four main parts: basic event characteristics, accident types, modeling methods, and event handling measures. In turn, semantic knowledge associations between its core elements are constructed. First, the four groups listed above were divided at the ontological level. Then, domain-specific splits were made based on the perceptual level of each ontological class. In this paper, the ontology layer composition is as follows: (1) find the event ontology's specialization domain; (2) use the top-down method to define the event ontology hierarchy; (3) define event category properties; and (4) create instances.

Subsequently, formulate the entities and attributes of the knowledge graph pertaining to road safety incidents occurring within the university. Firstly, in this paper, the knowledge graph of university traffic safety events is mainly designed with seven kinds of entities and attribute labels. The various entities and attribute labels are defined as shown in Table A1. Table A1 content details in Appendix A.3. Then, 10 relational mappings were designed, as shown in Table A2. Table A2's content is detailed in Appendix A.4. Secondly, through the definition of entities, attributes, and relationships, unstructured data originating from WeChat's official public numbers, Weibo, etc., is transformed into the structured format required for the construction of event knowledge graphs. Lastly, inter-entity relationships and characteristics can be used to create a single identification of traffic safety knowledge in the university. This will make it easier to connect the different pieces of knowledge about traffic events that happen at the university.

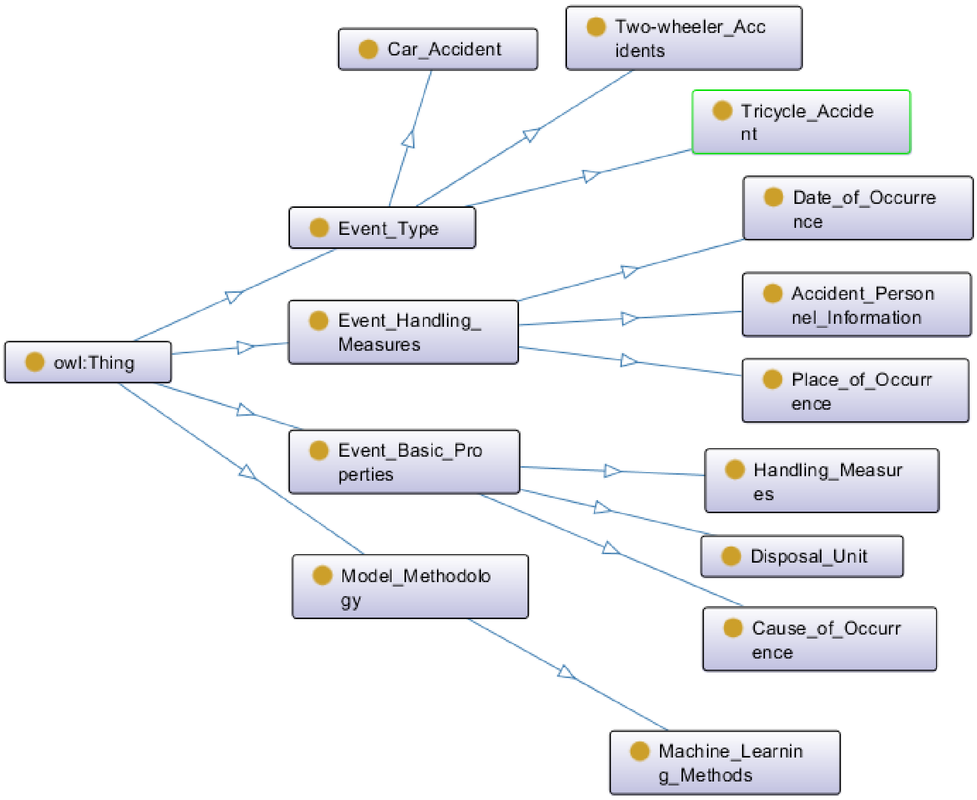

Figure 1. Ontology layer segmentation of the knowledge graph for traffic safety events at the university.

3.3. Event Extraction Model Design

Most of the existing knowledge graph triple extractions deal with text in terms of characters. However, the semantics of a single character are compared to a word. The character method contains different semantic information, which is prone to ambiguity and overlapping problems with the extracted triples. Texts often include many triples that overlap, making extraction difficult. If the model cannot handle triple group overlapping problems, it will not adapt to many datasets, resulting in serious limits and extraction errors. The extraction effect of event triples has a significant correlation with entity distance, and weak correlation between distant entities can lead to fuzzy entity boundaries and low relation extraction recall. Thus, this study uses the OneRel model [33] as the basis model. We propose the event extraction model Multi-Layer Semantic Graph Convolutional Entity-Relationship Joint Extraction (MLSRel) for the university on-campus traffic safety event knowledge graph. The model design of this paper is shown in Figure 2. First, the crawler technique is used to process the unstructured data information as an input to the MLSRel model. Second, based on the OneRel model, the global dependency semantics and syntactic graph embedding representation of sentences are incorporated in the initial vector generation stage. Additionally, Bidirectional Encoder Representations from Transformers (BERT) and Bi-directional Long Short-Term Memory (Bi-LSTM) networks were used to extract semantic information from the original text [34,35]. This stage improves relational extraction by learning the spanning sentence syntactic structure features. Next, a multilayer semantic graph convolutional neural network, Multilayer Graph Convolution Network (MultiGCN), is constructed to learn global semantic and syntactic graph embedding representation information to capture deeper semantic hidden information about entity relations [36]. Immediately after that, the learned graph embedding semantic vector G_e and the original text vector H_e are spliced to obtain a new sequence vector V_n. The new sequence vectors go through a graph mixing and pooling layer to capture the global range of semantic information. A multi-feature fusion attention mechanism is then designed to enhance the accuracy of model triple classification. This method is able to assign a high weight to the candidate entities in the entity extraction stage. This method improves the accurate recognition of entities at a distance. Finally, it goes through the $softmax()$ layer to improve the accuracy of recognizing triples and reduce the problem of overlapping triples and blurring of entity boundaries in the extraction process.

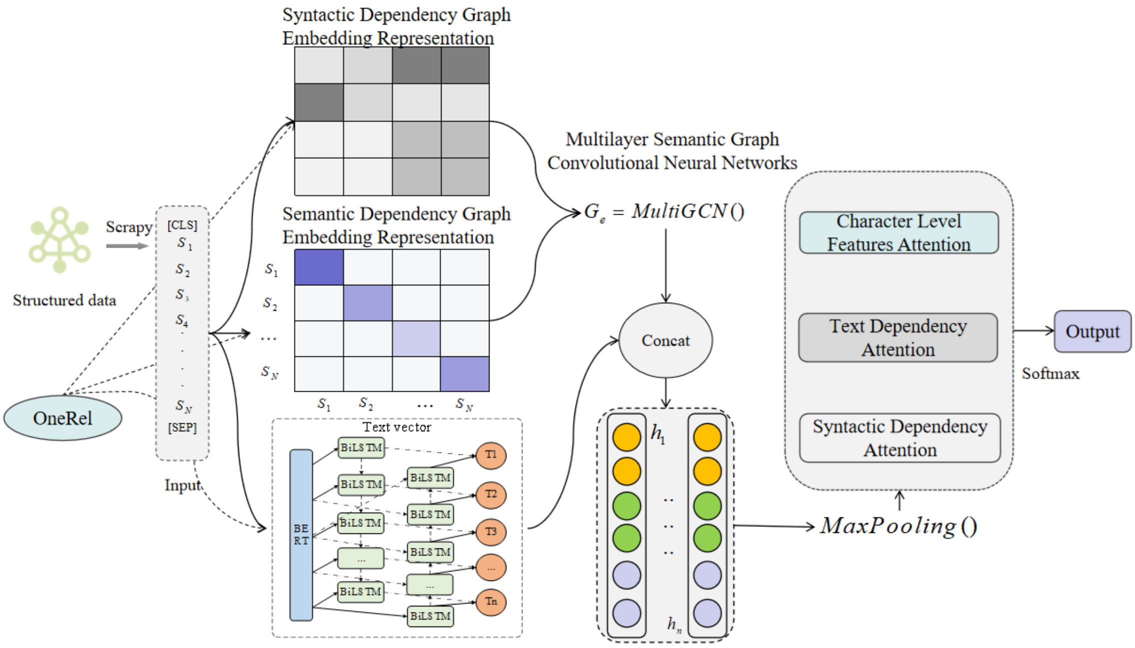

Figure 2. MLSRel overall model diagram.

Definition: first, given a training sentence $S = \{w_1, w_2, w_3, \ldots, w_L\}$, L in the training sentence S represents the length of the sentence sequence. Assume that the sentence receives a set Y of target triples as:

$$Y = \{(h_1, r_1, t_1), \ldots, (h_n, r_n, t_n) | h_n, r_n \in E, r_n \in R\} \tag{5}$$

In Equation (5), h_n, t_n represents the n-th head entity and tail entity. r_n represents the n-th relationship between entity pairs, E and R denotes the set of entities and relationships.

Input Sequence Encoding: In this paper, we first encode this training phrase with the BERT model to obtain the word vector representation corresponding to the sentence, as shown in Equation (6):

$$H_e = [h_1, h_2, h_3, \ldots, h_L] = BERT[w_1, w_2, w_3, \ldots, w_L]\{h_L \in R^d\} \tag{6}$$

where L in Equation (6) represents the number of characters contained in the sentence, d denotes the embedding dimension. $BERT$ represents the pre-trained model, which contains 12 hidden layers. Each of which has a size of 768. The training sentence S is encoded to obtain the input vector H_e.

3.3.1. Graph Embedding Representation Improvement

In this paper, we first use the Harbin Institute of Technology (HIT) Language Technology Platform (LTP) tool to preprocess the input sentences [37], and then obtain the word labeling information and syntactic dependency information of the preprocessed sentence sequences. Syntactic dependency analysis can identify the syntactic structure of a sentence or the dependencies between words in a sentence. More textual feature information can be captured through the construction of global semantic dependency syntactic graphs.A semantic–syntactic dependency graph is defined as $G_g = \{V_1, V_2, E_1, E_2\}$, where V_1 and V_2

represents the set of semantic and syntactic nodes in the graph, and E_1 and E_2 represent the set of edges in the semantic and syntactic graphs.

3.3.2. Multi-Layer Semantic Graph Convolutional Network Design

A multilayer semantic graph convolutional neural network is constructed to learn the topology in the global semantic and syntactic dependency graph which, in turn, yields the embedding vector representation of the text G_e. The computational process is shown in Equation (7):

$$G_e = MultiGCN(W_S(h_M + h_N) + b_S) \tag{7}$$

In Equation (7), W_S and b_S represent the weight parameter matrix and trainable parameter matrix. $MultiGCN()$ represents a multilayer graph convolutional neural network, which captures higher-order neighborhood information between word nodes. h_M and h_N represent the embedding representation vectors of semantic dependency analysis graphs and syntactic analysis graphs. The structure of the multilayer semantic graph convolutional neural network is shown in Figure 3:

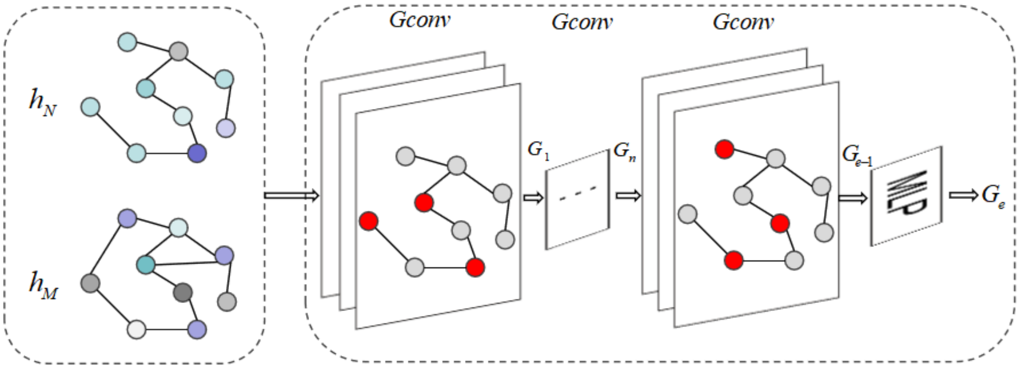

Figure 3. Structure of multilayer semantic graph convolutional neural network.

The resulting global semantic and syntactic dependency representation G_e is then spliced with the input vector H_e obtained from Equation (6) to obtain the spliced sequence representation vector V_n, as shown in Equation (8).

$$V_n = [G_e; H_e] \tag{8}$$

However, in different relationships, words and triples in a sentence are more relevant. Therefore, a hybrid graph pooling operation is used to capture the global scope information to obtain the vector representation H_g.

$$H_g = (MaxPooling(V_1, V_2, \ldots, V_n) + \frac{1}{|n|} \sum_{n \in G_g} V_n)/2 \tag{9}$$

where $V_1, V_2, \ldots, V_n$ in Equation (9) represents the embedded representation of the text. $MaxPooling()$ represents maximum pooling, and the second half of Equation (9) represents mean pooling.

3.3.3. Multi-Feature Fusion Attention Mechanism Design

Also, to remove noisy data and improve classifier accuracy, we assign corpus text weight coefficients using an attention method guided by word-level features and textual and syntactic dependence fusion features. In the case of the k-th layer, the feature representations are obtained as Z^k and M^k, respectively.

$$r = V_i(soft\max(\omega^{\mathrm{T}} \tanh(V_i)))^{\mathrm{T}} \tag{10}$$

$$Z^k = \tanh(r) \tag{11}$$

where the tanh function is used in Equation (11) to transform the spliced vectors to between [−1, 1], ω represents the trained parameter vectors, and the $softmax()$ function is normalized.

$$L_i{}^k = \tanh(w^k[V_i, H_g{}^k] + b^k) \tag{12}$$

$$\alpha_i{}^k = \frac{\exp(L_i{}^k)}{\sum\limits_i \exp(L_i{}^k)} \tag{13}$$

$$M^k = \sum_i \alpha_i{}^k V_i{}^T \tag{14}$$

In Equations (12)–(14), w^k and b^k represent the model parameters learned by the $k-1$-th layer of the attention mechanism. V_i and $H_g{}^k$ denote the inputs of the $k-1$-th layer of the attention mechanism, and stands for the i-th fused feature and the output of the $k-1$-th layer of the fused feature. Finally, drawing on the gating mechanism, the multi-feature vector fusion is denoted as D^k to achieve the purpose of complementary advantages.

$$C = \sigma(W_l{}^1 \tanh(W_l{}^2 Z^k + W_l{}^3 M^k)) \tag{15}$$

$$D^k = C \cdot Z^k + (1 - C) \cdot M^k \tag{16}$$

The above Equations, (15) and (16), represent the sigmoid activation function, W_l, representing the weight parameters learned by the self-training of the model. The vector C has the same number of dimensions as Z^k and M^k. This vector can dynamically assign weights to different features, thus avoiding information redundancy. Finally, the output layer uses $softmax$ to obtain the label of each character of the sentence, and outputs the final result. Due to sparsity in the training dataset, this study computes loss during training using the cross-entropy loss function at the global semantic and syntactic dependency graph embedding representation layer and the multi-feature attention mechanism layer. Weights are then assigned to each layer of loss, after which the modeled losses are summed up and calculated, as shown in Equation (17):

$$\text{Loss}_{total} = Loss_g + \alpha Loss_a \tag{17}$$

In this case, the Adam optimization algorithm is used in Equation (17), which is able to adapt to the problem of sparse gradient or large noise in the gradient. $Loss_g$ represents the loss of the embedding representation layer of the global semantic syntactic dependency graph. α represents the weight of the loss of the multi-feature attention mechanism layer. $Loss_a$ represents the loss of the multi-feature attention mechanism layer.

4. Experiment

4.1. Experimental Datasets and Evaluation Metrics

Experimental dataset: in order to evaluate the performance of the method in this paper, two publicly available datasets, NYT [38] and WebNLG [39], are used in the experiments to validate the effectiveness of the MLSRel model, as shown in Table 1.

Table 1. Statistics of the dataset.

Dataset	Number of Samples in the Training Set	Number of Samples in the Validation Set	Test Set Sample Size	Types of Predefined Relationships
NYT	56,195	5000	5000	24
WebNLG	5019	500	703	171

NYT dataset: This is derived from the corpus labeled by the New York Times. The relationships between the entities in the dataset are found by linking and referring to relationships in the external Freebase knowledge base, along with a remotely supervised relationship extraction algorithm. It contains 56,195 sentences in the training set, 5000 sentences in the validation set, 5000 sentences in the test set, and 24 predefined relations in the dataset.

WebNLG dataset: The WebNLG dataset is constructed for natural language generation tasks. The WebNLG training set contains 5019 sentences, the validation set contains 500 sentences, the test set contains 703 sentences, and the dataset contains 171 predefined relations.

Resource scheduling method test environment: this paper uses two kubernetes 1.22.0 servers for application deployment and scheduling experiments on the two servers deployed Ubuntu 20.04.2, Docker 20.10.14, the number of GPU cards is two, the memory is 256 GB.

Evaluation metrics: The experiments in this paper use the commonly used entity-relationship joint extraction evaluation metrics to evaluate the performance of the involved models, and the evaluation metrics mainly include the precision rate P, the recall rate R, and the $F1$ score. The calculation equations are shown in (18)–(20):

$$P = \frac{TP}{TP + FP} \tag{18}$$

$$R = \frac{TP}{TP + FN} \tag{19}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{20}$$

In Equations (18)–(20), TP represents the number of correctly predicted triples, FP represents the number of incorrectly predicted triples, and FN represents the number of true triples that were not correctly predicted. P represents the degree of accuracy of prediction in the results of positive samples. R represents the probability that a positive sample will be predicted as a positive sample. The $F1$ score represents the harmonic mean between P and R. In this paper, a triple is predicted correctly only when it is formed.

4.2. Experimental Environment Setting

The experimental development environment used in this paper is a Windows operating system with central processing (CPU) using an Intel (R) CoreTM i5-7200U CPU at 2.50 GHz and 12 GB of memory. Experiment running environment: Ubuntu operating system, graphics processor (GPU) using NVIDIA Tesla V100, 24G RAM. The experimental development framework is Pytorch, a deep learning framework, and the programming language is Python.

4.3. Experimental Comparison Model

The study compares its pairs to the newest joint extraction model of entity relations to demonstrate the model's efficacy. The comparison model is: CopyMTL [40], WDec [41], CasRel [42], TpLinker [43], SPN4RE [44], ENPAR [45], OneRel [33].

4.4. Experimental Results and Analysis

The experimental data for the improved dynamic resource scheduling method is shown in Table 2. Most of the environments where the models attempted in this paper are run experimentally are GPUs, so the tests are performed with Pod as the GPU type. This means that more GPU resources are needed than CPU resources. The experimental proof demonstrates that the use of the improved scheduling algorithm results in a notable reduction in the overall resource imbalance within the cluster. Additionally, there is a more equitable distribution of GPU and CPU resource utilization.

Table 2. Comparative effectiveness of resource utilization before and after improvements.

Node Type	Resource Type	Default Algorithm Resource Utilization	Improved Algorithm Utilization
Master	GPU	50%	**75%**
	CPU	18%	**27%**
	Random access memory (RAM)	6%	**18%**
Node1	GPU	50%	50%
	CPU	31%	26%
	Random access memory (RAM)	12%	**21%**

The comparative effectiveness of the model proposed in this paper with the baseline model mentioned above is demonstrated in Table 3 using the NYT and WebNLG English datasets. The majority of the baseline model results are obtained from the relevant literature. "-" is used to indicate the absence of a corresponding result for the model, while bold font highlights the best result.

Table 3. Comparative effectiveness of models on NYT and WebNLG English datasets.

Model	NYT Datasets			WebNLG Datasets		
	P (%)	R (%)	$F1$ (%)	P (%)	R (%)	$F1$ (%)
CopyMTL	0.727	0.692	0.709	0.578	0.601	0.589
WDec	0.843	0.764	0.802	-	-	-
CasRel	0.897	0.895	0.896	0.934	0.901	0.918
TpLinker	0.913	0.925	0.919	0.918	0.920	0.919
SPN4RE	0.933	0.917	0.925	0.931	0.936	0.934
ENPAR	0.936	0.920	0.928	0.934	0.916	0.925
OneRel	0.928	**0.929**	0.928	0.941	0.944	0.943
Ours	**0.958**	0.924	**0.941**	**0.945**	**0.951**	**0.947**

As can be seen in Table 3 above, the model shown in this paper excavates deeper feature information after the multi-layer semantic syntactic dependency graph convolutional neural network and multi-feature fusion attention mechanism. This makes the model better at expressing itself compared to other models. On the NYT dataset and WebNLG dataset, the $F1$-value of the model proposed in this paper improves by 1.3% and 0.4%, respectively, compared to the baseline model OneRel. Our proposed model, MLSRel, has better performance improvement for entity-relationship triple group extraction in sentences. It is easier to deal with the problem of overlapped triples with the model. The experiment results are also looked at to show that the model proposed in this study does not enhance the NYT dataset as much as the other models. It is possible that this is due to the fact that the BERT model has a low representational learning ability on this dataset, which causes errors spread among the training tasks. This, in turn, leads to the joint extraction of triples, which does not result in an improvement in performance, but instead causes error triples to appear.

4.5. Ablation Experiment

In this paper, we base our model on OneRel et al. [33]. Ablation experiments are performed on the NYT and WebNLG datasets for the method proposed in this paper. The results are shown in Table 4. Row 1 of Table 4 shows the experimental results for the baseline model, rows 2 and 3 show the experimental results with the addition of global semantic dependency vectors and syntactic dependency vectors, row 4 shows the experimental results with the addition of the multi-feature attention mechanism, and row 5 shows the experimental results with the addition of global semantic syntactic dependency vectors and the multi-feature attention mechanism.

Table 4. Comparative results of ablation experiments.

Model	NYT Datasets *F*1 (%)	WebNLG Datasets *F*1 (%)
Baseline	0.928	0.943
$+G_S$	0.930	0.936
$+G_Y$	0.933	0.940
$+A_{MF}$	0.940	0.946
$+G_S+G_Y$	0.937	0.945
$+G_S+G_Y+A_{MF}$	**0.941**	**0.947**

After adding the Global Semantic Dependency Vector ($+G_S$), Global Syntactic Dependency Vector ($+G_Y$), and Multi-Feature Attention Mechanism ($+A_{MF}$), respectively, the model's value on two of the datasets improved. However, it decreased on the WebNLG dataset. The reason for this analysis may be that, although the global semantic features of the English characters in the sentence are fully considered, it is not sufficiently characterized when mining the association information of multiple semantics of head/tail entities at a deeper level. It can bring noisy information to the model and affect the performance of triple extraction. After adding the Global Semantic Dependency Vector ($+G_S$) and Global Syntactic Dependency Vector ($+G_Y$) at the same time, the $F1$-values on the two datasets are 93.7%, 94.5%, respectively, and the $F1$-values are all improved. It shows that the proposed fused global semantic and syntactic dependency graph embedding vectors can effectively improve the extraction triples. Finally, after adding Global Semantic Dependency Vector ($+G_S$), Global Syntactic Dependency Vector ($+G_Y$), and Multi-feature Attention Mechanism ($+A_{MF}$) at the same time, the values on the two datasets are improved by 1.3%, 0.4%, respectively. It is shown that the proposed fusion of global semantic and syntactic dependency graph embedding vectors and a multi-feature attention mechanism can help the baseline model improve English triples extraction. The reason for this analysis may be that semantic and syntactic dependency analysis is able to uncover semantic features between word granularity and provide deeper information about the dependencies between words. The constructed multi-feature attention mechanism is able to capture the correlation between the word granularity level and the target entity to further mine the sentence-level semantics. The above strategy can effectively solve the problems of triples overlapping and the existence of entity boundary ambiguity in relation extraction.

4.6. Question Analysis

In this paper, we have designed a knowledge Q&A system based on the campus traffic event knowledge graph, which can quickly and effectively assist the managers on the university campus to deal with traffic safety events and improve the level of intelligent emergency management. The Q&A system can also adopt different answer selection strategies according to different types of events.

4.7. Graph Visualization

Based on the risk keywords (university traffic accidents, etc.), taking the universities in Henan Province as an example, we collected and organized real textual information about university on-campus traffic risk events from the official government website, Weibo, and Baidu news webpage. We constructed a comprehensive knowledge graph of on-campus traffic safety events to assist the relevant departments in their oversight. The visualization diagram of the knowledge graph part of the university traffic safety events is shown in Figure 4.

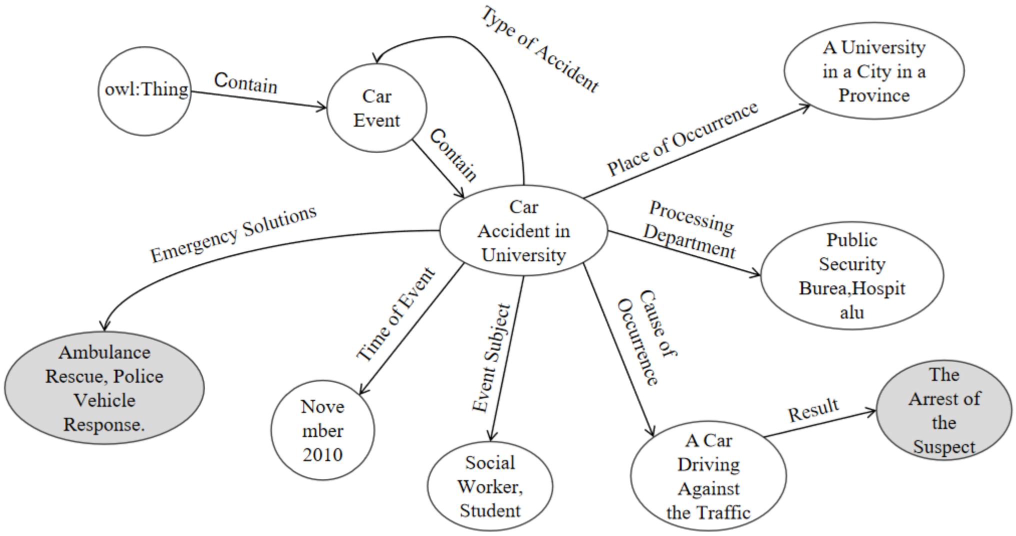

Figure 4. Visualizing the knowledge graph of traffic safety events in universities.

5. Conclusions

The frequent occurrence of campus safety events has brought serious security risks to the safety of teachers and students. This paper proposes a knowledge graph construction method for campus traffic events for the safety of students and teachers, in which the proposed event knowledge ontology model provides a canonical definition for the construction of university traffic safety event knowledge graphs, the improved MLSRel model effectively solves the triples overlapping problem and the fuzzy entity boundary problem in relationship extraction, and the method is able to protect the safety of students and teachers. However, this paper's model has limitations. For example, complex logical relationships between multiple events (cause and effect, inversion, etc.) express the need for consideration. The constructed knowledge graph of traffic safety events on university campuses lacks explainability. We will keep working to make the model structure better so it can handle complex relational situations. We will also look into event knowledge graph methods that are easy to understand.

Author Contributions: Conceptualization, X.L., Q.Z. and L.Z.; methodology, X.L., L.Z. and Z.Z.; software, X.L., K.W. and Z.C.; validation, Z.C. and J.L.; formal analysis, L.Z.; investigation, X.L. and Z.C.; resources, L.Z. and K.W.; data curation, X.L. and Z.Z.; writing—original draft preparation, X.L. and L.N.; writing—review and editing, L.Z.; visualization, L.Z. and K.W.; supervision, Z.Z., L.N. and F.W.; project administration, L.Z.; funding acquisition, L.Z., Z.Z. and F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Research Projects of Henan Higher Education Institutions, grant number No. 22B520054; Henan Province International Science and Technology Cooperation Project, grant number No. 232102520004; Key Scientific Research Projects of Colleges and Universities in Henan Province, grant number No. 23A520031; Songshan Laboratory Pre-research Project, grant numberYYJC032022021; and Natural Science Foundation of Zhongyuan University of Technology, grant number K2023MS021, P. R. China.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Kezheng Wang was employed by the company Lenovo Group Solution Center. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

Appendix A.1. Architecture Design of Knowledge Graph Construction for Traffic Safety Events in University

The detailed model architecture of this paper is shown in Figure A1. The graph has physical, computing, storage, virtualization, and graph construction layers. The physical resource layer is the hardware server. The computing and storage resource layer mainly includes Mysql, Neo4j databases, GPU and CPU computing resources. Mysql stores structured data, and Neo4J stores knowledge-based triple-group graph data [46]. GPU and CPU computing resources can provide computing support for data acquisition, ontology construction, and entity and relationship extraction model training [47]. The virtualization layer manages and schedules resources using KVM-QEMU, and kubernetes. It boosts resource efficiency and overcomes model training's memory shortage on a single device. Containerization technology can effectively reduce the difficulty for users to construct knowledge graphs. The graph construction layer is mainly for data processing and obtaining knowledge about graph triples, etc.

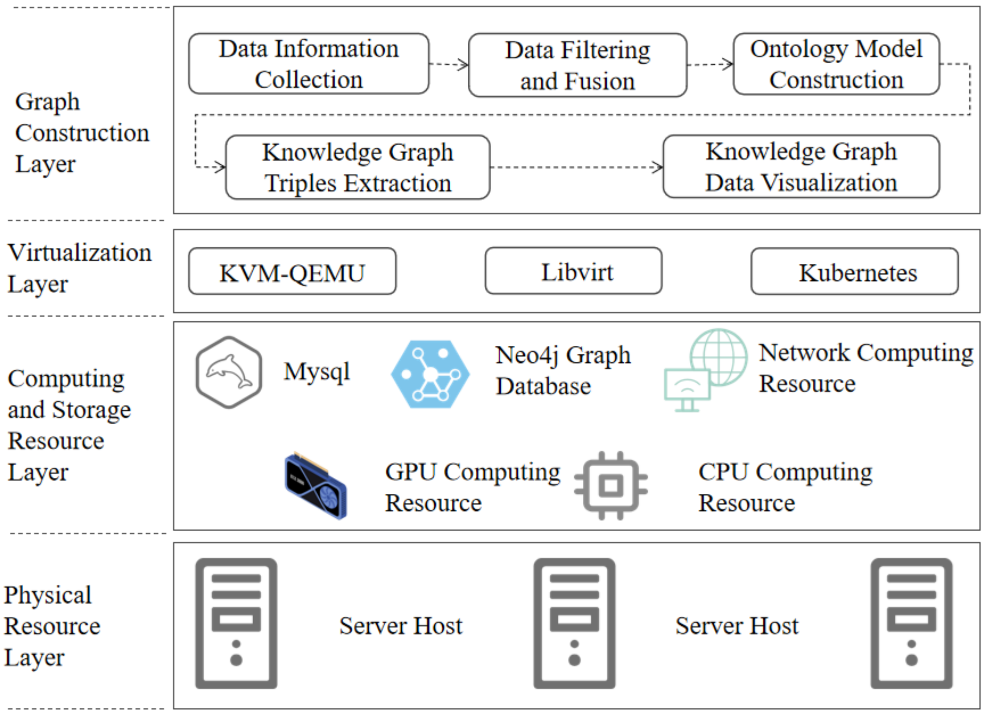

Figure A1. Architecture diagram of knowledge graph construction for traffic safety events at university.

Appendix A.2. Architecture Design of Virtualized Resources and Scheduling Technology

As shown in Figure A2, we first use KVM-QEMU technology to achieve complete virtualization of hardware resources. Specifically, you need to configure a virtualized resource pool environment for KVM-QEMU. KVM is loaded into the Libvirt kernel in the form of a driver module, making the Libvirt kernel an efficient virtual machine monitor and QEMU for device virtualization. Loading the hypervisor and Hyper-visor controls virtual resources after constructing the virtualized resource pool [48]. Immediately following this, the Device plugin enables kubernetes to properly schedule pods to GPU and CPU resource nodes based on constraints. Every pod can contain one or more containers. When kubelet finds that a successfully scheduled pod requests GPU (CPU) resources, it makes an allocation request to the device plugin to allocate GPU (CPU) resources for the pod.

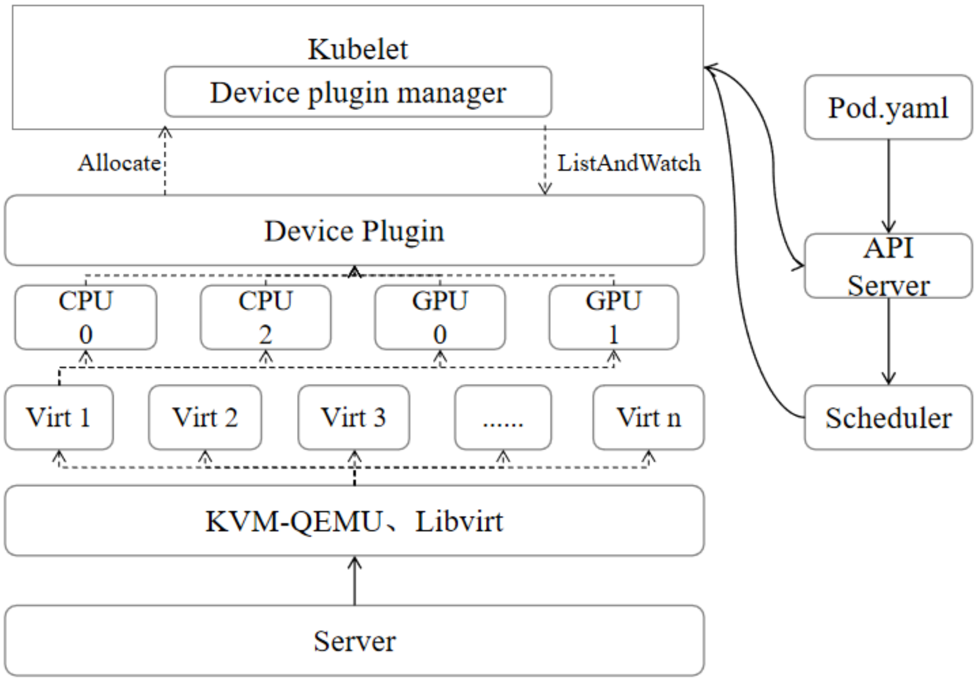

Figure A2. Architecture diagram of virtualized resources and scheduling technology.

Appendix A.3. Design of Knowledge Graph Entities and Attribute Labels for Traffic Safety Events in the University

Table A1. Design of knowledge graph entities and attribute labels for traffic safety events in the university.

Serial Number	Entity and Attribute Tag Name	Entity and Attribute Label Meaning	Example
1	Location	Geographic location of the event	A university in a city in a province
2	Time	Time of event	November 2010
3	Source of information	Relevant websites where the event was reported	Dahe Network
4	Event subject	People information	Third-year university student
5	Event data	Number of persons involved in the event, reasons for the event, etc.	1 victim, car driving against traffic, etc.
6	Emergency solutions	Emergency response plan for traffic safety events	Ambulance rescue, police vehicle response, etc.
7	Model methodology	Modeling methods that are relevant to the study of traffic safety events	Machine learning and other methods

Appendix A.4. Relationship Types and Samples of Knowledge Graphs for Traffic Safety Events in the University

Table A2. Relationship types and samples of knowledge graphs for traffic safety events in the university.

Serial Number	Relationship Type Name	Example
1	Published content	<Dahernet, Published Content, Detailed Information on Traffic Safety Events on University Grounds>
2	Date of occurrence	<An event, Time of occurrence, November 2010>
3	Place of occurrence	<An event, Place of occurrence, A university in a city in a province>
4	Event objects	<An event, Object of the event, Member of the community/student>
5	Cause of occurrence	<An event, The reason for it, A car driving against the traffic>
6	Type of accident	<An event, Type of accident, Automobile accident>
7	Processing department	<An event, Handling department, Public Security Bureau>, <An event, Handling department, Hospital>
8	Result	<An event, The result of the process, The arrest of the suspect>
9	Reference plan	<An event, A reference plan, A measure to deal with a past case>
10	Model methodology	<An event, Modeling methods, Machine learning, Statistical analysis, etc.>

Appendix A.5. NYT and WebNLG Dataset Download Links

NYT: https://github.com/davidsbatista/Annotated-Semantic-Relationships-Datasets/blob/master/datasets/DataSet-IJCNLP2011.tar.gz (accessed on 16 June 2023).

WebNLG: https://github.com/fuzihaofzh/webnlg-dataset (accessed on 15 June 2023).

References

1. Sun, L.L.; Liu, D.; Chen, T.; He, M.T. Road traffic safety: An analysis of the cross-effects of economic, road and population factors. *Chin. J. Traumatol.* **2019**, *22*, 290–295. [CrossRef] [PubMed]
2. Wang, L.; Wang, Y.; Shi, L.; Xu, H. Analysis of risky driving behaviors among bus drivers in China: The role of enterprise management, external environment and attitudes towards traffic safety. *Accid. Anal. Prev.* **2022**, *168*, 106589. [CrossRef] [PubMed]
3. Gan, L.; Ye, B.; Huang, Z.; Xu, Y.; Chen, Q.; Shu, Y. Knowledge graph construction based on ship collision accident reports to improve maritime traffic safety. *Ocean. Coast. Manag.* **2023**, *240*, 106660. [CrossRef]
4. Cvitković, I.; Vilke, S.; Krpan, L.; Brlek, P. Removing regulatory features of traffic control in school zones. *Transp. Res. Procedia* **2022**, *60*, 228–234. [CrossRef]
5. Yan, S.; Su, Q.; Gong, Z.; Zeng, X. Fractional order time-delay multivariable discrete grey model for short-term online public opinion prediction. *Expert Syst. Appl.* **2022**, *197*, 116691. [CrossRef]
6. Sayed, A. Use of Machine Learning and Natural Language Processing to Enhance Traffic Safety Analysis. Ph.D. Thesis, The University of Wisconsin-Milwaukee, Milwuakee, WI, USA, 2022.
7. Zhang, X.; Zhang, J. Classification and topic tracking of college students' cybersecurity education based on the internet. *J. Comput. Methods Sci. Eng.* 2023, *in press*.
8. Telima, M.; El Esawey, M.; El-Basyouny, K.; Osama, A. The use of crowdsourcing data for analyzing pedestrian safety in urban areas. *Ain Shams Eng. J.* **2023**, *14*, 102140. [CrossRef]
9. Schwalbach, J.; DeAngelis, C.A. School sector and school safety: A review of the evidence. *Educ. Rev.* **2022**, *74*, 882–898. [CrossRef]
10. Chen, X.; Jia, S.; Xiang, Y. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.* **2020**, *141*, 112948. [CrossRef]
11. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [CrossRef]
12. Zhu, G.; Zhang, M.; Xi, Y. Knowledge graph-based prediction of evolutionary outcomes of urban rail transit emergencies. *J. Electron. Inf.* **2023**, *45*, 949–957.
13. Sun, X.; Mengm Y.; Wang, W. Recognition of microblog traffic events based on knowledge graph and target detection. *Data Anal. Knowl. Discov.* **2020**, *4*, 136–147.

14. Song, Y.; Li, W.; Dai, G.; Shang, X. Advancements in Complex Knowledge Graph Question Answering: A Survey. *Electronics* **2023**, *12*, 4395. [CrossRef]
15. Munir, S.; Jami, S.I.; Wasi, S. Towards the modelling of Veillance based citizen profiling using knowledge graphs. *Open Comput. Sci.* **2021**, *11*, 294–304. [CrossRef]
16. Ge, Y.; Tian, Y.C.; Yu, Z.G.; Zhang, W. Memory sharing for handling memory overload on physical machines in cloud data centers. *J. Cloud Comput.* **2023**, *12*, 1–20. [CrossRef]
17. Li, J.Y.; Du, K.J.; Zhan, Z.H.; Wang, H.; Zhang, J. Distributed differential evolution with adaptive resource allocation. *IEEE Trans. Cybern.* **2022**, *53*, 2791–2804. [CrossRef] [PubMed]
18. Carrión, C. Kubernetes scheduling: Taxonomy, ongoing issues and challenges. *ACM Comput. Surv.* **2022**, *55*, 1–37. [CrossRef]
19. ALshalabi, H.; Tiun, S.; Omar, N.; Ali Alezabi, K.; Al-Aswadi, F.N. The Effectiveness of Arabic Stemmers Using Arabized Word Removal. *Int. J. Inf. Sci. Manag.* **2022**, *20*, 87–102.
20. Spoladore, D.; Sacco, M.; Trombetta, A. A review of domain ontologies for disability representation. *Expert Syst. Appl.* **2023**, *228*, 120467. [CrossRef]
21. Wang, F.; Yang, J.; Xu, L.L. Research on ontology construction for fire emergency management. *J. Intell.* **2020**, *39*, 914–925.
22. Wang, T.; Zheng, L.; Lv, H.; Zhou, C.; Shen, Y.; Qiu, Q.; Li, Y.; Li, P.; Wang, G. A distributed joint extraction framework for sedimentological entities and relations with federated learning. *Expert Syst. Appl.* **2023**, *213*, 119216. [CrossRef]
23. Upadhyay, P.; Balalau, O.; Manolescu, I. *Open Information Extraction with Entity Focused Constraints*; Findings of the Association for Computational Linguistics; EACL: Dubrovnik, Croatia, 2023; pp. 1255–1266.
24. Yang, J.; Wu, Z.; Wu, R. Micro-Expression Spotting Based on VoVNet, Driven by Multi-Scale Features. *Electronics* **2023**, *12*, 4459. [CrossRef]
25. Zhu, X.; Zhou, G.; Chen, J.; Lu, J.; Xiang, Y. A single-stage joint entity relationship extraction method based on enhanced sequence labeling strategy. *Comput. Sci.* **2023**, *50*, 184–192.
26. Lee, C.V.G.E.; Pierleoni, A. Improving Distantly Supervised Document-Level Relation Extraction Through Natural Language Inference. *DeepLo* **2022**, *2022*, 14.
27. Ding, X.; Zhou, G.; Lu, J.; Chen, J. Research on document-level relationship extraction methods for enhanced entity representation. *Comput. Sci.* **2023**, *50*, 157–162.
28. Omotehinwa, T.O. Examining the developments in scheduling algorithms research: A bibliometric approach. *Heliyon* **2022**, *8*, e09510. [CrossRef] [PubMed]
29. Abbasi, R.; Martinez, P.; Ahmad, R. An ontology model to represent aquaponics 4.0 system's knowledge. *Inf. Process. Agric.* **2022**, *9*, 514–532. [CrossRef]
30. Polenghi, A.; Roda, I.; Macchi, M.; Pozzetti, A.; Panetto, H. Knowledge reuse for ontology modelling in Maintenance and Industrial Asset Management. *J. Ind. Inf. Integr.* **2022**, *27*, 100298. [CrossRef]
31. Li, Z.; Liu, X.; Wang, X.; Liu, P.; Shen, Y. Transo: A knowledge-driven representation learning method with ontology information constraints. *World Wide Web* **2023**, *26*, 297–319. [CrossRef]
32. Agrawal, G.; Deng, Y.; Park, J.; Liu, H.; Chen, Y.-C. Building Knowledge Graphs from Unstructured Texts: Applications and Impact Analyses in Cybersecurity Education. *Information* **2022**, *13*, 526. [CrossRef]
33. Shang, Y.M.; Huang, H.; Mao, X. Onerel: Joint entity and relation extraction with one module in one step. In Proceedings of the AAAI Conference on Artificial Intelligence 36, Vancouver, BC, Canada, 22 February–1 March 2022; pp. 11285–11293.
34. Liu, Y.; Wei, S.; Huang, H.; Lai, Q.; Li, M.; Guan, L. Naming entity recognition of citrus pests and diseases based on the BERT-BiLSTM-CRF model. *Expert Syst. Appl.* **2023**, *234*, 121103. [CrossRef]
35. Bokolo, B.G.; Liu, Q. Deep Learning-Based Depression Detection from Social Media: Comparative Evaluation of ML and Transformer Techniques. *Electronics* **2023**, *12*, 4396. [CrossRef]
36. Ali, A.; Zhu, Y.; Zakarya, M. Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. *Neural Netw.* **2022**, *145*, 233–247. [CrossRef] [PubMed]
37. Yang, X.; Li, W.; Chen, Y.; Guo, Y. Construction of a COVID-19 Pandemic Situation Knowledge Graph Considering Spatial Relationships: A Case Study of Guangzhou, China. *ISPRS Int. J.-Geo-Inf.* **2022**, *11*, 561. [CrossRef]
38. Lai, T.; Cheng, L.; Wang, D.; Ye, H.; Zhang, W. RMAN: Relational multi-head attention neural network for joint extraction of entities and relations.*Appl. Intell.* **2022**, *52*, 3132–3142. [CrossRef]
39. Zhao, K.; Xu, H.; Cheng, Y.; Li, X.; Gao, K. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowl.-Based Syst.* **2021**, *219*, 106888. [CrossRef]
40. Zeng, D.; Zhang, H.; Liu, Q. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. In Proceedings of the AAAI Conference on Artificial Intelligence 34, New York, NY, USA, 7–12 February 2020; pp. 9507–9514.
41. Nayak, T.; Ng, H.T. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In Proceedings of the AAAI Conference on Artificial Intelligence 34, New York, NY, USA, 7–12 February 2020; pp. 8528–8535.
42. Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; Chang, Y. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–8 July 2020; pp. 1476–1488.
43. Wang, Z.; Yang, L.; Yang, J.; Li, T.; He, L.; Li, Z. A Triple Relation Network for Joint Entity and Relation Extraction. *Electronics* **2022**, *11*, 1535. [CrossRef]

44. Sui, D.; Zeng, X.; Chen, Y.; Liu, K.; Zhao, J. Joint entity and relation extraction with set prediction networks. *IEEE Trans. Neural Netw. Learn. Syst.* 2023, *in press*. [CrossRef]
45. Wang, Y.; Sun, C.; Wu, Y.; Zhou, H.; Li, L.; Yan, J. ENPAR: Enhancing entity and entity pair representations for joint entity relation extraction. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, Kyiv, Ukraine, 19–23 April 2021; pp. 2877–2887.
46. Sun, L.; Dou, Y.; Yang, C.; Zhang, K.; Wang, J.; Philip, S.Y.; He, L.; Li, B. Adversarial attack and defense on graph data: A survey. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 7693–7711. [CrossRef]
47. Liang, C.; Yang, J.; Du, R.; Hu, W.; Tie, Y. Non-Uniform Motion Aggregation with Graph Convolutional Networks for Skeleton-Based Human Action Recognition. *Electronics* **2023**, *12*, 4466. [CrossRef]
48. Monteil, J.B.; Iosifidis, G.; DaSilva, L.A. Learning-based Reservation of Virtualized Network Resources. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 2001–2016. [CrossRef]