applied sciences

MDPI

*Article*

# EDET: Entity Descriptor Encoder of Transformer for Multi-Modal Knowledge Graph in Scene Parsing

**Sai Ma [1], Weibing Wan [1,*], Zedong Yu [1] and Yuming Zhao [2]**

[1] Department of Computer, Shanghai University of Engineering Science, Shanghai 201620, China; masai970@163.com (S.M.); zedongyu@sues.edu.cn (Z.Y.)
[2] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China; arola_zym@sjtu.edu.cn
* Correspondence: wbwan@sues.edu.cn

**Abstract:** In scene parsing, the model is required to be able to process complex multi-modal data such as images and contexts in real scenes, and discover their implicit connections from objects existing in the scene. As a storage method that contains entity information and the relationship between entities, a knowledge graph can well express objects and the semantic relationship between objects in the scene. In this paper, a new multi-phase process was proposed to solve scene parsing tasks; first, a knowledge graph was used to align the multi-modal information and then the graph-based model generates results. We also designed an experiment of feature engineering's validation for a deep-learning model to preliminarily verify the effectiveness of this method. Hence, we proposed a knowledge representation method named Entity Descriptor Encoder of Transformer (EDET), which uses both the entity itself and its internal attributes for knowledge representation. This method can be embedded into the transformer structure to solve multi-modal scene parsing tasks. EDET can aggregate the multi-modal attributes of entities, and the results in the scene graph generation and image captioning tasks prove that EDET has excellent performance in multi-modal fields. Finally, the proposed method was applied to the industrial scene, which confirmed the viability of our method.

**Keywords:** scene parsing; knowledge graph; multi-modality



**Citation:** Ma, S.; Wan, W.; Yu, Z.; Zhao, Y. EDET: Entity Descriptor Encoder of Transformer for Multi-Modal Knowledge Graph in Scene Parsing. *Appl. Sci.* **2023**, *13*, 7115. https://doi.org/10.3390/app13127115

Academic Editors: Jiaqi Li, Božidar Šarler, Haiping Liu and Jian Zhang

Received: 1 May 2023
Revised: 24 May 2023
Accepted: 2 June 2023
Published: 14 June 2023



## 1. Introduction

Scene parsing is a set of multiple tasks that perceive the scene and extract the semantic and internal connections in the scene. From the complexity of scene parsing tasks, it can be divided into object detection, entity relationship prediction, scene description generation, etc. Based on the information carrying capacity of knowledge graph, it can be used as the intermediate product of some scene parsing tasks. As shown in Figure 1, scene parsing was regarded as a task based on the multi-modal knowledge graph to solve sub-tasks such as entity relationship prediction and scene captioning generation.

Knowledge graph is a graph topology for storing and expressing information. It is a directed graph composed of nodes and edges that can carry information and has powerful logical representation capability to effectively and intuitively express the relationships among objects. Logically, the basic constituent elements of the knowledge graph include entities, relations, and attributes; topologically, the knowledge graph is composed of nodes and edges. This means that, in the knowledge graph, entities can exist as independent nodes, relations must be between nodes, and attributes are embedded inside entities and relations. Since knowledge graphs store and express knowledge in a way that is closer to how humans do it, they have attracted a lot of attention in the field of artificial intelligence in recent years, and various organizations have established knowledge graphs in different domains. The established knowledge graphs include general knowledge graph FB15K [1], medical knowledge graph DiaKG [2], CORD-19 [3], etc. When building knowledge graphs, entities are usually required to be objects with distinguishability and independent existence.

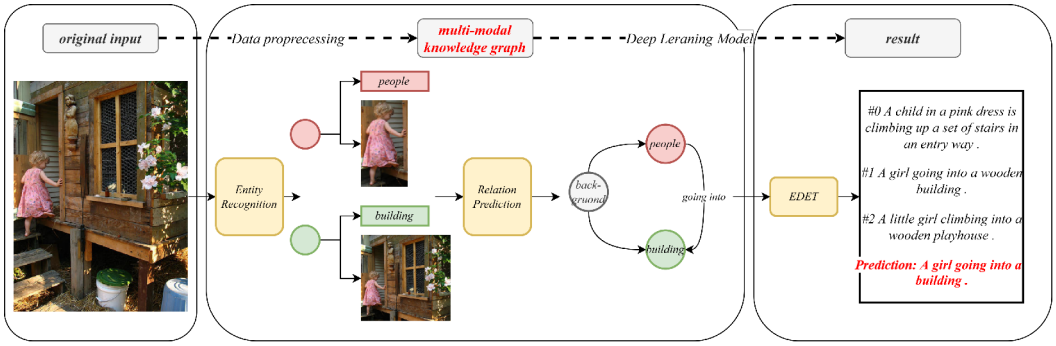



**Figure 1.** The scene parsing task can be decomposed into multiple sub-tasks based on knowledge graph.

According to the representation of information in them, they can be divided into knowledge graphs containing only a single modality such as Cora [4] and Citeseer [5] etc., and multi-modal graphs such as MMKG [6] etc. In a single modality knowledge graph, there is only one mode of information. The two single modality knowledge graphs mentioned above only contain text-modal information. The multi-modal knowledge graph contains information in various forms, and they generally contain information in text, image modalities, and even audio and video modalities. In order to enable computers to understand and apply the information in knowledge graphs, a variety of knowledge representation methods have been proposed.

Traditional knowledge representation methods can be divided into methods based on information propagation, such as GCN [7] and GAT [8], and methods based on spatial domain distance such as TransE [1], TransH [9], and TransR [10]. The former methods are applicable to small-scale graph structure data. They represent information of the nodes as the aggregation of other nodes connected to them and learn the representation of each node through the information propagation algorithm. While the latter maps the nodes into a semantic space, the relationship between nodes is represented by the distance between nodes mapped in the edge semantic space. Both types of methods have good performance on a variety of tasks, but it can be found that they treat the attributes as the linked nodes of entity nodes, which means that the knowledge representation of each entity node no longer satisfies the distinguishability and independence properties when the knowledge graph is established.

When choosing a knowledge graph as the intermediate of scene content understanding, unifying the multi-modal information into the form of a knowledge graph, we can use a unified structure to align different modal information. It can also be used to solve different problems, improve the mobility of the scene parsing model, and the knowledge graph generated by scene tasks could improve the reliability of the generated results. In order to verify the feasibility of the processing proposed, the scene information needs to be represented by a graph structure; however, the corresponding datasets and related research are missing both in the current field of scene paring and the knowledge graph. At present, large-scale multi-modal knowledge graphs often only provide data and do not provide corresponding task annotations. Taking MMKG as an example, we can find many small graphs in the dataset, but the categories of each small graph are not marked. Therefore, through the data pre-processing, the data in the scene parsing tasks could be transformed into the nodes, attributes, and edges of the knowledge graph, and then the parsing results are generated by a deep learning model.

Single-modal information is not enough for a computer to understand the real world, and multi-modal information is more important for scene perception and cognition than before. Multi-modal knowledge graphs that use information, such as text, images, audio, and video, as entities or attributes are increasingly available, so traditional knowledge representation methods have difficulty representing them in a unified semantic space.

In order to solve the above problems, unify the processing of multi-modal information on the knowledge graph, and preserve the real characteristics of entities, we proposed

a structure named Entity Descriptor to form the knowledge representation of an entity, where the representation result of an entity is related only to its own attributes. The representation preserves the independence and distinguishability of entity nodes and has the ability to express multi-modal information. At the same time, the representation can be perfectly integrated with the transformer [11] structure and can make full use of its attention mechanism to perceive global information. The new transformer-based encoder, EDET, which can aggregate the entity features of multi-modal scenes, was validated by conducting a large number of experiments, and the results show its effectiveness in scene parsing tasks. The main contributions of this work are as follows:

1. A knowledgerepresentation method named Entity Descriptors was proposed. This method focuses on the intrinsic multi-modal attributes of entities to ensure that the entities have independence in the process of forming knowledge representation. The knowledge representation result formed by this method retains the correlation and the differences between nodes and has a good clustering effect without training.
2. In order to transform the scene information into knowledge graph representation, this paper summarized three basic methods to select entity attributes according to experience. This method is oriented to specific questions, answering the questions of how to screen, add, transform, delete (ignore), and other operations of the entities and attributes of the knowledge graph when facing different scenarios, so as to retain useful information, remove redundancy, and reduce the cost of model training, deployment, and other stages.
3. A new way of handling scene parsing tasks based on Entity Descriptor was proposed, and a transformer-based network for multi-modal knowledge mapping tasks was established. The definition of Entity Descriptor perfectly fits the long sequence input and global perception of transformer structure, which can still guarantee attention to the full graph in scenes where entity nodes are independently represented.

## 2. Related Work

### 2.1. Knowledge Representation

Conventional convolution methods are only applicable to data in Euclidean space and are difficult to apply directly to the non-Euclidean graph topological structure data. Based on the parameter sharing property of convolution kernels in CNNs, GCN has been proposed by applying the convolution operation analogy to graph structures. This work used Laplacian matrix representation of graph structures, restricted the perceptual field of learnable convolution kernels using Chebyshev polynomials, and aggregated the information of the node itself and its first-order neighboring nodes to form a knowledge representation of that node. This representation performs well on a variety of graph structure tasks such as node classification and graph categorization. This method is suitable for small-scale homogeneous graphs, and the application presupposes that the node has the LP (Label Propagation) property and that the representation of each node depends on the nodes connected to it. This property is not applicable in the knowledge graph, where similar models based on GCN, such as GAT, GraphSAGE [12], etc., have similar problems.

The basic elements of the knowledge graph are entities, attributes, and relationships. When the distinction between entity nodes and attribute nodes is ignored, attributes can be expressed in the form of entity-owning attribute–attribute value, at which time the knowledge graph can be regarded as only two elements—nodes and edges—and all the information within the graph can be expressed in the form of *head-edge-tail*, i.e., ($h$,$l$,$t$). Inspired by the translation invariance of word2vec [13], TransE was proposed to represent the vectors of nodes and links as satisfying the form $h + l = t$. Most of the subsequent extensions based on the TransE method, such as TransH and TransR, expand the semantic space and replace the distance metric function on this basis. This representation method is convenient for obtaining the relationship information between entities easily at the time of application, but due to the complexity of realistic semantics, there may be multiple relationships between nodes that are difficult to describe by the distance metric function,

and this method is applicable to the semantic web that requires nodes to all be represented in the same semantic space, and is not applicable to the multi-modal knowledge graph scenario.

### 2.2. Scene Parsing and Multi-Modal Knowledge Graph

#### 2.2.1. Predicate Classification

The scene graph generation task is the task of generating the semantic graph structure of the scene according to the input scene image, which is a part of the understanding of the scene content. Predicate classification is one of the sub-tasks of scene graph generation.

In the scene graph generation task, for a given image, the target in the image can be recorded as a node set, and the relationship between the targets can be recorded as an edge set, so the scene graph of the image can be represented, that is, the scene graph containing the target and its associated relationship is generated according to the image. The predicate classification task obtains the correlation relationship between targets under the condition of knowing the image and the target in the image. The purpose of the corresponding task is to obtain the transition probability of various relations between targets in a given set of relations.

#### 2.2.2. Image Captioning

Image captioning is one of the classic tasks of scene parsing. This task is required to generate a semantic description based on the input image. The content of the image captioning task is the intersection of computer vision and natural language processing, which requires the computer to perceive the high-level semantic information according to the underlying visual information.

At present, the generative models designed for image captioning mostly use the structure of image coding—visual feature extraction—description coding—decoding, in which CNN is commonly used for image coding, RNN, and LSTM for caption generation, and the related research shows that the structure of a pure transformer is also feasible and can achieve better results.

## 3. Our Approach

### 3.1. Entity Description

In tasks related to graph topology, only two constituent elements—nodes and edges—are usually considered. The attributes are often considered as nodes and thus the fact that attributes are intrinsic properties of nodes is ignored; this blurs the definition of basic constituent elements of a knowledge graph. In the knowledge graph, entity nodes should be independent and distinguishable; if the scene does not change, the entity node itself should remain unchanged when it interacts with other entities.

Figure 2 can more conveniently help us understand the importance of entity node independence. As shown in Figure 2a, in a scene with three entities—a table, a chair, and a cup—placing the cup on the table or chair will generate different relations cup-on-table and cup-on-chair, but the cup in this scene is the same cup regardless of the relation with which object. When we represent this cup using the knowledge representation represented by GCN, the high-dimensional vector representation of the cup is affected by the entities it interacts with, resulting in multiple representations of a concrete, physically existing object in a semantic space, causing the model's ability to express knowledge to shrink and the cost of parsing it to increase.

The distinguishability of entity nodes is equally important. In the scene shown in Figure 2b, when there are many objects that can generate relations with the table, the distance-based approach in the spatial domain represented by TransE reveals the drawback that it is difficult to handle complex one-to-many and many-to-many relations, because at this time a large number of entities generates the same interaction with the central entity, such as the table, which means that the high-dimensional representations of a large number of mutually-independent entities are concentrated in a same hyperplane, the

distinguishability between entities is greatly reduced, and then the relationships between these entities, and other kinds of relationships involved in the central node, are also difficult to express.

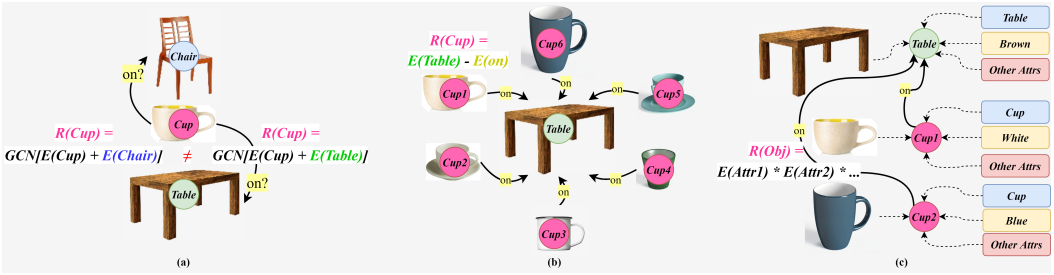



**Figure 2.** The differenceof knowledge representation methods. (**a**) Methods based on GCN; (**b**) Methods based on TransE; (**c**) Our Entity Descriptor method, the asterisk means aggregation operation.

Combining the above reasons, we want to take full advantage of the knowledge graph. As shown in Figure 2c, the representation of entity nodes focuses on describing the information of the entities themselves. Suppose entity nodes in the knowledge graph have the set of attributes $A$. For its attributes of different modalities, it uses different static feature extraction networks to represent them as a tensor with the same dimension, while attributes of the same modality are extracted using the same extraction network to ensure that a class of attributes is embedded in the same feature space. In order to avoid the attributes of different modalities from influencing each other, the processes of attribute acquisition features are all independent of each other, and finally they are stitched together to distinguish different attributes at different locations. Then, its Entity Descriptor $ED$ can be expressed as:

$$ED = \text{Aggregate}(W_1 \times (\text{Concat}(\sigma(W_2 \times A + b)))), \tag{1}$$

where $W_1$ is the weight matrix of features, $W_2$ is the attribute embedding matrix. Aggregate means aggregate function. $\sigma$ means activation function. In different scenarios, the aggregation function can choose many methods, such as weighted sum, splicing, and mapping. Splicing means to connect the obtained attribute tensors back and forth, and the result is a one-dimensional tensor. Mapping involves converting high-dimensional attribute tensors into low-dimensional representation tensors through mapping functions or convolutional layers.

Suppose the set of nodes is $X = \{x_i | i = 1, 2, \ldots, N_x\}$; every node has its attributes set $D = \{d_j | j = 1, 2, \ldots, N_d\}$, then all attributes could be expressed as $Y = \{y_i j | i = 1, \ldots, N_x, j = 1, \ldots, N_d\}$, in which $y_i j$ means that the value of the $i$th entity's $j$th attribute is $y_i j$. According to Entity Descriptor, the representation of entity $x_i$ should be:

$$x_i = \text{Aggregate}_{j=1}^{N_d} y_{ij}. \tag{2}$$

While the aggregation function chooses weighted sum, splicing and mapping methods, for $x_i \in X$, we have:

$$x_i = \begin{cases} \sum_{j=1}^{N_d} w_j y_{ij}, \sum_{j=1}^{N_d} w_j = 1 \\ \text{Concat}_{j=1}^{N_d}(y_{ij}) \\ W_0 \bar{Y}_{ij} W_1, \end{cases} \tag{3}$$

where $\bar{Y}_{ij}$ is the representation matrix of attributes and each column vector is the embedded representation of the corresponding attribute value. $W_0$ means the weight transformation matrix, the form of which is trainable hidden layers. $W_1$ aims to adjust all types of attribute vectors to the same dimension. The aggregation of weighted sum is the most direct, but this method requires manually setting the weight of each attribute and the entity representation obtained in this way is prone to confusion; splicing retains all attributes of the entity but usually causes the dimension of the entity representation to be too long to train. Using

the direct mapping method, although solving the disadvantages of the first two methods, usually requires the appropriate weight transformation hidden layer in the long-term and a large number of trainings.

In order to reduce the training time and support the dynamic of new entities, based on the GCN for graph structure information aggregation, using the triangle matrix $U$ as an attribute node and the entity link matrix, we can obtain a new aggregation function which is small and easy to train:

$$x_i = \text{ Aggregate } \left(\bar{Y}_{ij}\right) = \sigma\left(U^{-\frac{1}{2}}\left(\bar{Y}_{ij} + I\right)U^{\frac{1}{2}}\right). \tag{4}$$

To sum up, when there are few attributes, the aggregation function can choose the way of weighted sum. When the attribute is short and easy to splice, we can choose splicing as the aggregation function. When there are many, or complex, attributes, attribute features should be extracted first and then the aggregation function in Equation (4) should be selected to aggregate the attribute information.

Entity Descriptor uses each entity's own attributes to generate the corresponding representation results, which makes it possible to obtain the same entity representation for the same entity in different scenes, always maintaining the independent distinguishability of entity nodes in the process of generating the representation. The generated entity representations are also distinguishable between different entities of the same class due to the differences in attributes.

### *3.2. Attribute Selection Conditions*

In the process of scene parsing mentioned above, it is necessary to transform the data in the scene into a knowledge graph; this part of the work is not annotated in the current scene parsing dataset. While in a fixed scene, the choice of entity nodes is often uncontroversial—using the minimum level entity that is involved in the task—what needs to be discussed is how to choose attributes among the massive data. So, we summarized three attribute selection conditions to guide the construction of the knowledge graph.

#### 3.2.1. Common Condition: The Characteristics That Entity Nodes of the Same Class Have Can Be Used as Attributes

In a knowledge graph, entities are the smallest unit involved in the problem. Entity nodes of the same class should maintain homogeneity, that is, when discussing entity nodes, the characteristics they all possess can be seen as attributes. When the characteristics of an entity node are unique and cannot be expressed by other nodes, these characteristics cannot be treated as attributes and they cannot be merged for similar nodes. The purpose of this condition is to classify the entity nodes in a knowledge graph, and the processing methods of similar nodes are unified, which not only conforms to the understanding of things in the real world but also saves computing resources.

#### 3.2.2. Unique Interaction Condition: An Attribute Cannot Have Multiple Kind of Relationship Links to Entities of the Same Class

The idea of the unique interaction condition comes from the concept of the lattice representation method of Formal Concept Analysis. This method expresses things with their own connotation, and there is only a unique partial order set in the binary relationship between things and descriptors, that is, under a fixed background and attributes, the representation of things in that attribute is unique. This condition is to ensure that, in a class of attributes, there is one and only one kind of relationship between attributes and entity nodes. In the case of a fixed relationship, a single entity cannot be connected to multiple homogeneous attributes, while a single attribute can be connected to multiple homogeneous entity nodes.

#### 3.2.3. Limited Scope Conditions: Entity Nodes and Attribute Nodes Only Exist in Their Respective Scopes

When an attribute has the same name as an entity, or both are the same object in reality, the attribute node and entity node of the object cannot be represented as the same vector. Although entities and attributes are often represented as nodes in the process of building or designing knowledge graphs, they are actually completely different. This condition exists to distinguish attributes and entities. In a knowledge graph, as the size of entity nodes increases, the links between entity nodes and attributes will become more complex and some entities may exist as attributes of other entities. Using limited scope conditions to divide nodes and draw the boundary between entities and attributes can not only represent entities more clearly but can also more conveniently model entities and attributes separately to avoid confusion.

### *3.3. Entity Descriptors in Multi-Modal Tasks*

For aggregate functions in Equation (3), the weighted sum and direct splicing ways are only applicable to the knowledge graph with only a class of entities and simple attributes while, in most cases, the knowledge graph is complex and with diverse attributes. When choosing mapping, it not only faces the situation in which the dimension of the obtained representation is higher than that of the transformer architecture needed, but also faces the problem that different categories of entity nodes have different vector dimensions and do not meet the requirements of equal length sequence.

For the problems existing in the aggregation function of mapping, we can imitate the processing method of high-dimensional patches in CV and set a separate convolution channel or linear channel for each type of entity. The node representation of the same entity uses the same channel and different types of entities are processed separately. This is due to the large difference in attributes between different categories of entities, and the data of heterogeneous entity representation vectors in the same column may come from completely irrelevant information. However, using different channels to extract features of different types of heterogeneous attributes not only compresses the entity representation to two dimensions, which is easy for the model to learn, but also converts them to the same dimension while retaining the differences between different types of entity for parallel processing. This paper uniformly names the models containing Entity Descriptor as EDET (Entity Descriptor Encoder of Transformer).

In the existing multi-modal graphs, there is no corresponding processing task to verify the effectiveness of its specific application. Therefore, tasks can be converted into the form of multi-modal knowledge graphs and then Entity Descriptor can be used to solve the task.

#### 3.3.1. Predicate Classification

While facing the predicate classification task, the attribute selection conditions above were used to build a knowledge graph of the source data. Then, the entity in scene graph was noted as $N$ and all $n$ entities set as $V = \{N_1, \ldots, N_n\}$. Each entity node $N_i$ has attributes including sub-image $A^i_{image}$, label $A^i_{label}$, and location $A^i_{bbox}$. Set the shape of the feature vector as $1 \times 512$. Among them, the sub-image uses a pre-trained feature extraction convolutional model such as VGG16, which could be marked as *CNN*, and then obtains a $1 \times 512$-dimensional feature vector by flattening. The way to process label information is to establish a vocabulary and then use the word embedding method [14] to obtain the corresponding vector representation. This methods is marked as *embed*. The location information contains less information and its dimension can be expanded to 512 through a linear layer, which is marked as *Linear*. The aggregation function chooses to use a $3 \times 3$ convolutional layer and fills it with padding to maintain the dimension of the final result. At this time, the Entity Descriptor $X_i$ of the node $N_i$ is expressed as:

$$X_i = \text{Aggregate}(CNN(A^i_{image}), embed(A^i_{label}), Linear(A^i_{bbox})). \quad (5)$$

In the predicate classification, the result that needs to be output is the relationships between entity nodes, so the representation of relational elements can be ignored when input and the model structure is shown in Figure 3a. However, knowledge graphs usually have relationships, so in order to operate uniformly on the model, a set of relationship edges can be preset using the information they already have. Then, the model structure is as shown in Figure 3b. For this task, the relative location between the entity nodes can be used as the preset information of the edges. Suppose two nodes in graph with their position $(x_1, y_1)$ and $(x_2, y_2)$, their relative location $r_{1,2}$ could be expressed as:

$$\begin{aligned} l &= \arctan(y_1 - y_2) / \arctan(x_1 - x_2) \\ r_{1,2} &= \operatorname{arctanh}\left[\left(e^{l} - e^{-l}\right) / \left(e^{l} + e^{-l}\right)\right]. \end{aligned} \tag{6}$$

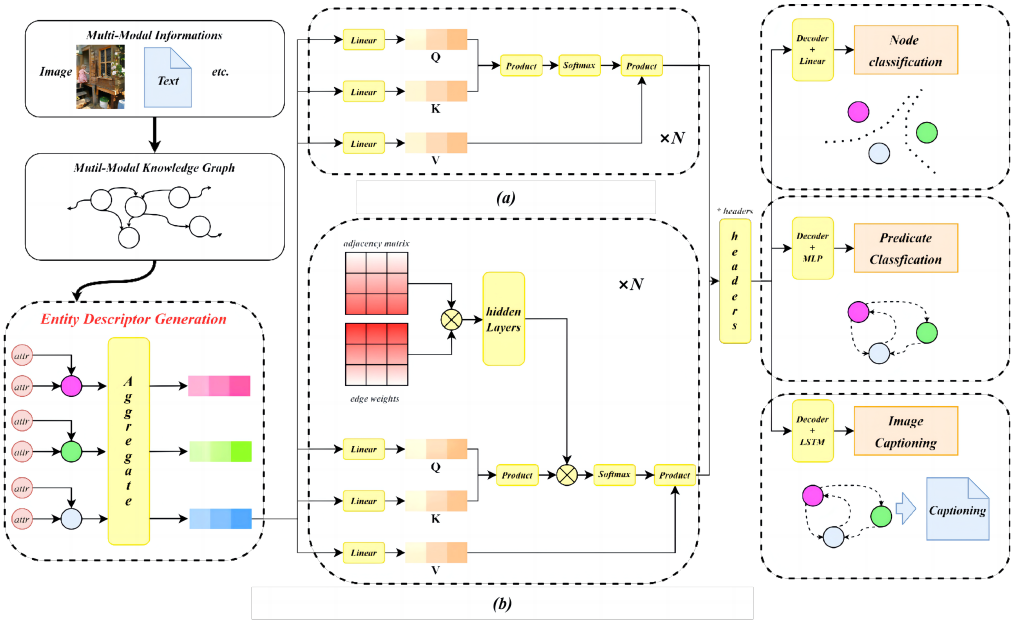



**Figure 3.** EDET: (**a**) Entity Descriptor Encoder of Transformer. (**b**) Entity Descriptor Encoder of Transformer with edge information. ×***N*** means repeat this structure for N times.

### 3.3.2. Image Captioning

Image captioning is a task for generating the captioning of the current image. Unlike in the predicate classification task, the scene image in an image captioning dataset does not mark the target and its location at first. As shown in Figure 4, for the image captioning task, the Mask R-CNN [15] model is used to obtain the entity and its attribute information. It should be noted that the entity nodes and attributes obtained here have biases and errors compared with the entities involved in the annotation sentences of the image captioning. The bias mainly comes from three aspects: one is that the entity object obtained by the pre-trained model is not necessarily the entity in the sentence; the other is that the model can usually recognize the visual target by shape but it cannot recognize backgrounds such as the sky; third, the model can recognize limited objects, and the target objects involved in the sentence may not be within the scope of recognition. The source of the error is because the model cannot achieve 100% accuracy for the target objects within the recognition range.

The entity and its attributes in the image captioning task are treated in the same way as in predicate classification. Meanwhile, in image captioning, there is another method used to generate edges for the graph—using a predicate classification model that has been trained in the previous section. It should be noted that, with this method, the relationship between entity nodes is trained by external data and has bias which exists in the predicate classification task.

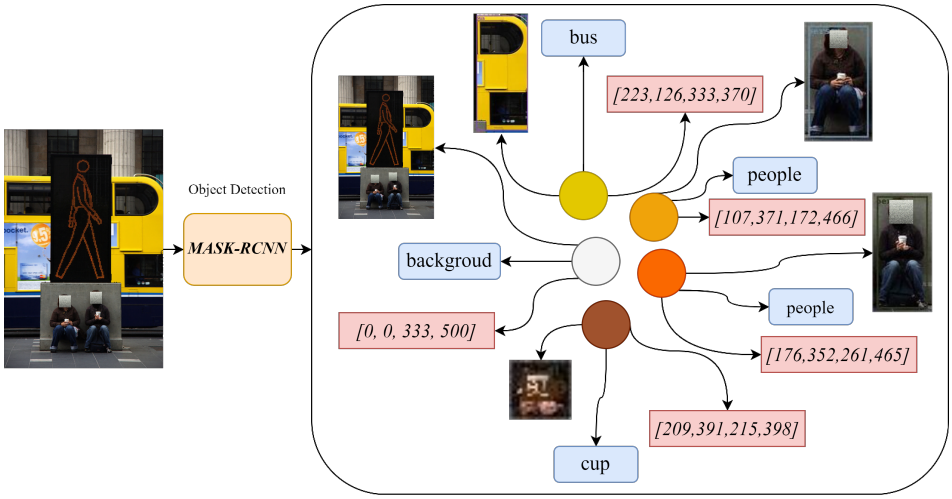



**Figure 4.** Pre-processing of image captioning task. Using Mask R-CNN to generate multi-modal knowledge graph.

## 4. Experimental Results

In this section, experiments were carried out to verify the ideas proposed above. First, the effectiveness of our proposed pipeline for scene parsing was verified using semantic segmentation experiments. This experiment was performed on the ADE20K dataset [16] and its experimental results show that feature engineering can assist deep learning models.

Secondly, the effectiveness of the Entity Descriptor was verified on the Cora dataset. Cora is a scientific publication dataset in which data can be expressed in the form of a graph structure. Then, the feasibility of our proposed attribute selection conditions was verified on industrial data from real projects.

Finally, the effectiveness of EDET was verified by predicate classification and scene description tasks on Visual Genome dataset [17] and Flickr30k dataset [18], respectively. The application of EDET in actual industrial scenes is introduced at last.

### *4.1. Validation of Feature Engineering of Deep Learning Model*

In the previous section, we presented a new solution to scene parsing tasks. It is important to note that this solution adopts a streaming solution, which is essentially to give up the end-to-end model and solve the problem as an engineering one—based on the knowledge graph structure, using the pre-trained model for feature engineering, then based on the knowledge graph of the deep learning model to solve the task. This method of using feature engineering to extract prior information has significant advantages in the interpretability of output results; however, deep learning models have a large number of parameters and a complex network structure, and are theoretically able to simulate almost all the operations in feature engineering. In order to support the idea of multi-stage process processing, feature engineering effectiveness experiments were designed to verify whether feature engineering is still necessary for the existing deep learning model in this section.

We chose to explore the above problems in the semantic segmentation task. The purpose of this task was to obtain pixel-level boundaries between different classes of objects in the image, which is a popular fine-grained classification task in CV. One of the major pain points facing this task is the accuracy of the object edges.The Laplace edge detection algorithm is a simple algorithm commonly used in image processing, which can be regarded as a fixed convolution kernel of $3 \times 3$. In order to keep the edge information data dimension consistent with the source data, we used multi layer perceptron with about 50,000 parameters to keep the data dimension.

The influence of adding a few parameters to image segmentation is shown in Table 1. The effect on mAcc is not significant (4.01–11.18%), while mIoU, which measures the accuracy at the pixel level, is significantly improved (21.59–37.32%). It can also be seen in Figure 5 that the model metrics can gain a certain amount of improvement at the beginning of training with

the inclusion of edge features and can reach higher values after the same number of training epochs. This shows that even a feature extraction operator with nine fixed parameters can improve the performance of a model with over 30 million parameters, disproving the implicit assumption that "the current model structure contains the optimal solution to the problem" and proving that feature engineering is still necessary in today's machine learning.

**Table 1.** Parameter comparison table. (Params: Number of training parameters + number of fixed parameters, mAcc: Average class accuracy, mIoU: Mean Intersection-over-Union, Score: Average of mAcc and mIoU).

| Model | Params | mAcc | mIoU | Score |
|---|---|---|---|---|
| FCN [19] | 35,322,218 | 67.3 | 22.7 | 45.00 |
| FCN + Edge | 35,372,394 + 9 | **70.0** | **27.6** | **48.80** |
| PSPNet [20] | 46,582,337 | 68.9 | 28.4 | 48.65 |
| PSPNet + Edge | 46,632,513 + 9 | **76.6** | **39.0** | **57.80** |

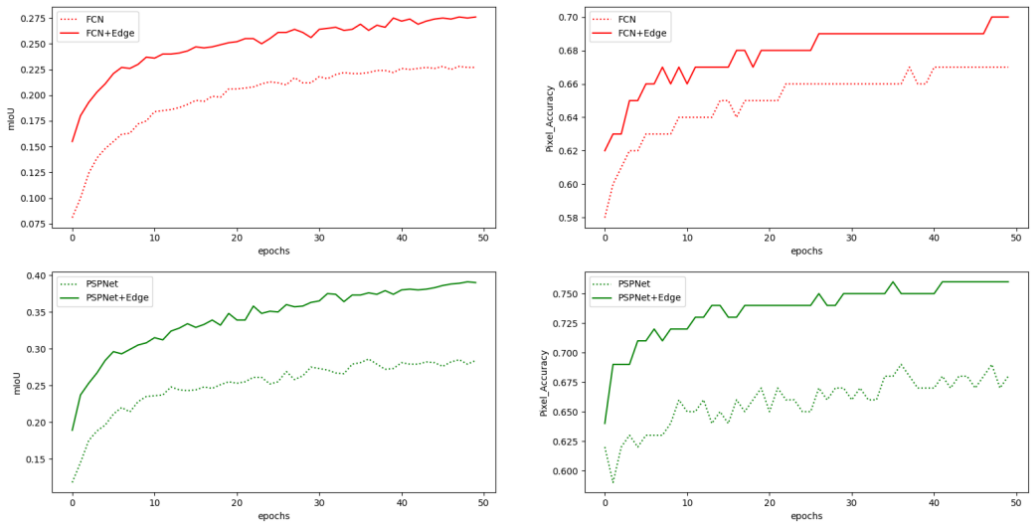



**Figure 5.** The effect of a small number of parameters on the training process.

### *4.2. Validation of Entity Descriptor Representation Methods*

After verifying the effectiveness of feature engineering, we proposed the idea of the Entity Descriptor. In order to verify that the idea of aggregating entity node information through the Entity Descriptor is feasible, preliminary experiments were performed on entity descriptors to verify its feasibility in simple tasks of a single-modal knowledge graph, and then the computer vision task and multi-modal knowledge graph were popularized in the subsequent experiments.

We chose to verify the validity on the scientific publication dataset Cora. The Cora dataset contains 2708 scientific publications, 5429 edges, and a total of seven categories. Treating publications as entity nodes, each entity node consists of 1433 attributes, each representing a keyword, taking values represented by 0/1 only, corresponding to the absence/existence of that keyword.

First, the publications were defined as entity nodes, while keywords were defined as entity properties and reference relationships were defined as relationship edges, and the reference network was converted into a single-modal knowledge graph. Secondly, the Entity Descriptor was used to encode entity nodes. The 1433 keyword attributes constitute a word list of attribute information through the embedding layer to generate 1433 one-dimensional vectors, and the dimension of this attributes vector was set to 128. Then, the initial encoding of each publication was converted to a two-dimensional vector of $1433 \times 128$, using the aggregation function in Equation (4) to encode the entity node

representation. Finally, the encoding was trained to the same GCN network as the reference network for 100 epochs.

In the traditional GCN approach, the information propagation properties of the publication entities are used to obtain a randomness representation and then the classification results are obtained by a two-layer graph convolution network. Entity Descriptors, on the other hand, first embed attributes into high dimensions to obtain semantic representations of the attributes and then fuse the attribute information into the corresponding entity nodes by trainable weights.

Experimental results can be seen in Table 2 and Figure 6. In the process of generating node representation, the node representation obtained by the traditional method is influenced by other adjacent nodes, where the trainable weight is between all nodes in the whole graph; the node representation obtained by the Entity Descriptor is only related to the properties of the entity node itself, and its trainable weight is only effective within the entity. This leads to the effect of the first epoch in Table 2 and Figure 6. The node representation obtained by the traditional method at the beginning has strong randomness and the classification accuracy is only 8%, while the Entity Descriptor already has a certain classification effect at the initial time, with a classification accuracy of 37%, far more than that of the traditional method. After the training of the same model and the same iters (100 epochs), the model classification accuracy of the traditional method was 77%, while that of the Entity Descriptor representation method was 82%, and the performance was improved by 6.49 percentage points.

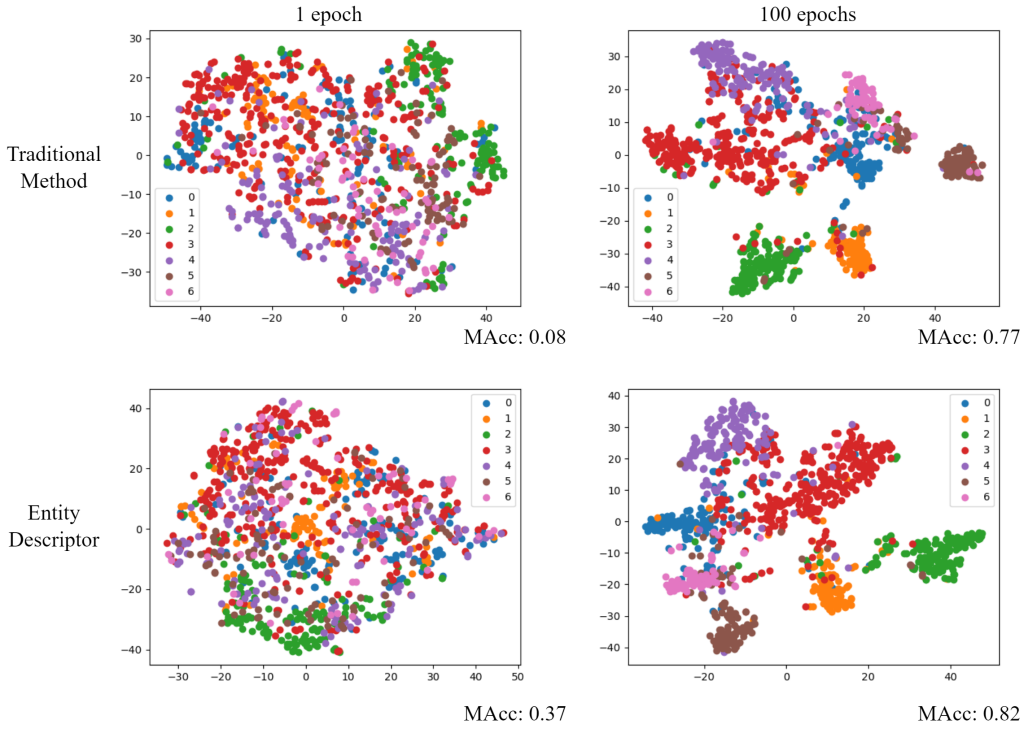



**Figure 6.** Performance of Entity Descriptor and traditional method in knowledge representation.

**Table 2.** Performanceof GCN and ED + GCN on Cora.

| Model | mAcc, Epoch = 1 | mAcc, Epoch = 100 |
|---|---|---|
| GCN | 8% | 77% |
| ED + GCN | **37%** | **82%** |

The results of the Entity Descriptor validity experiment show that the Entity Descriptor can obtain better knowledge representation results at the beginning of encoding the entity

node, retain the original clustering nature of the entity, and drive the model to learn more information so as to generate better results.

### *4.3. Validation of Attributes Selection Conditions*

To verify the effectiveness of our selection methods, we chose four entities in the industrial dataset: machine door, safety door, spindle, manipulator, and using the proposed attributes selection conditions to create a multi-modal graph. The entities and their selected attribute information are shown in Table 3, and the final multi-modal graph can be generated as shown in Figure 7.

**Table 3.** Selected attributes of entities in industrial dataset.

| Entity Attribute | Has Part | Type | Image |
|---|---|---|---|
| Machine Door | plug | Parts | ✓ |
| Spindle | bearing | Parts | ✓ |
| Safety Door | plug | Equipment | ✓ |
| Manipulator | jaws | Parts | ✓ |

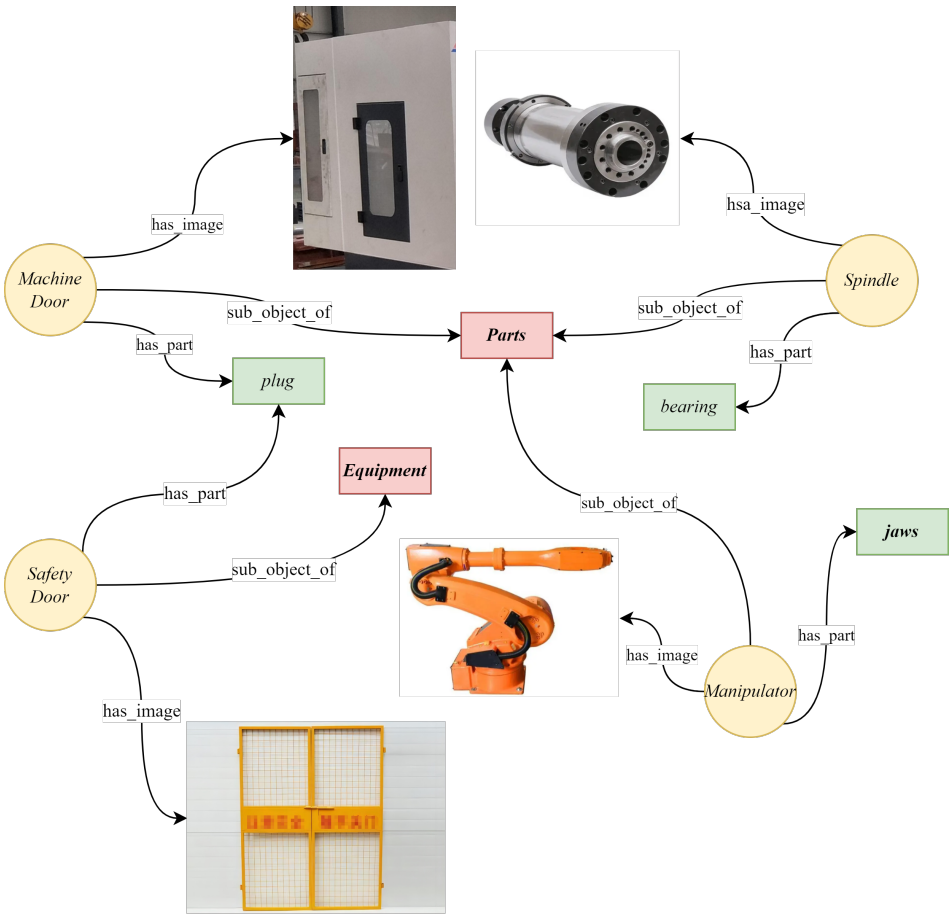



**Figure 7.** Multi-modal knowledge graph extracted from industrial data by attribute selection methods.

After obtaining node representations in the experiment, the node representation vector of each entity was compressed into the same two-dimensional space, and the distance between two representations shows the similarity of the two nodes. Four experiments were performed, including all attributes, without the *Has Part* attribute, without the *Type* attribute, and without the *Image* attribute. The experimental results, respectively, were expressed in dark red, pink, blue, and green nodes, and different entity nodes with different shape nodes. Machine door, spindle, safety door, and manipulator were marked as triangle, square,

rhombus, and circle, respectively. The results of the final attribute selection conditional validity experiment are shown in Figure 8.

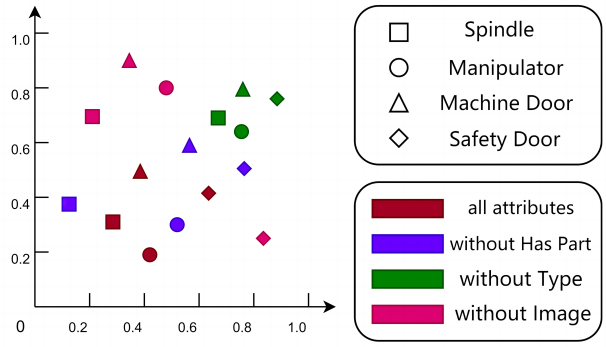



**Figure 8.** Our selection methods could help researchers filter entity attributes from massive data and build usable knowledge graphs.

In Figure 8, it can be seen that the similarity between the node representations obtained by using all attribute aggregation is not much different. When the *Type* attribute is masked, the similarity between machine door and safety door node representation increases; when the *Has Part* attribute is masked, the similarity of machine door, spindle, and manipulator node representations increases; when the sub-image information of entity nodes is masked, the similarity between all node representations is further improved.

Based on the results obtained by the aggregation of all attributes, the reason for the change of node representations was analyzed. Combining the attribute information, it can be seen that, when *Type* is ignored, the node representations of the same *Has Part* attribute become closer. Similarly, when ignoring the *Has Part* attribute, nodes of the *Parts* type become more similar. Compared with the other two attributes, the images contain too much redundant information, such as background information, and the similarity between all nodes is improved after the image attribute is ignored.

Experimental results show that attribute selection conditions can assist in judging which attributes are key attributes among entities, and the selected attributes can preserve the independence and separability of entity nodes. This has certain research significance for the large-scale development of knowledge graphs and the development of models based on knowledge graphs.

### *4.4. Performance of EDET in Scene Parsing*

#### 4.4.1. Predicate Classification

In the scene graph generation task, our model is only involved in the relationship generation part, so only the metrics in the predicate classification PredCls were compared.

The relationships in the Visual Genome dataset have the characteristics of long-tailed distribution. To overcome this difficulty, we chose to use focal loss as the loss function for EDET. The performance of EDET in predicate classification is shown in Table 4 and Figure 9.

**Table 4.** Performance of EDET in predicate classification.

| Method | R@20 | R@50 | R@100 |
|---|---|---|---|
| Graph-RCNN [21] | - | 54.2 | 59.1 |
| Neural-motifs [22] | 58.5 | 65.2 | 67.1 |
| NODIS [23] | 58.9 | 66.0 | 67.9 |
| VC-Tree [24] | 59.8 | 66.2 | 67.9 |
| GPS-Net [25] | 60.7 | 66.9 | 68.8 |
| **EDET** | **62.7** | **68.6** | **70.3** |

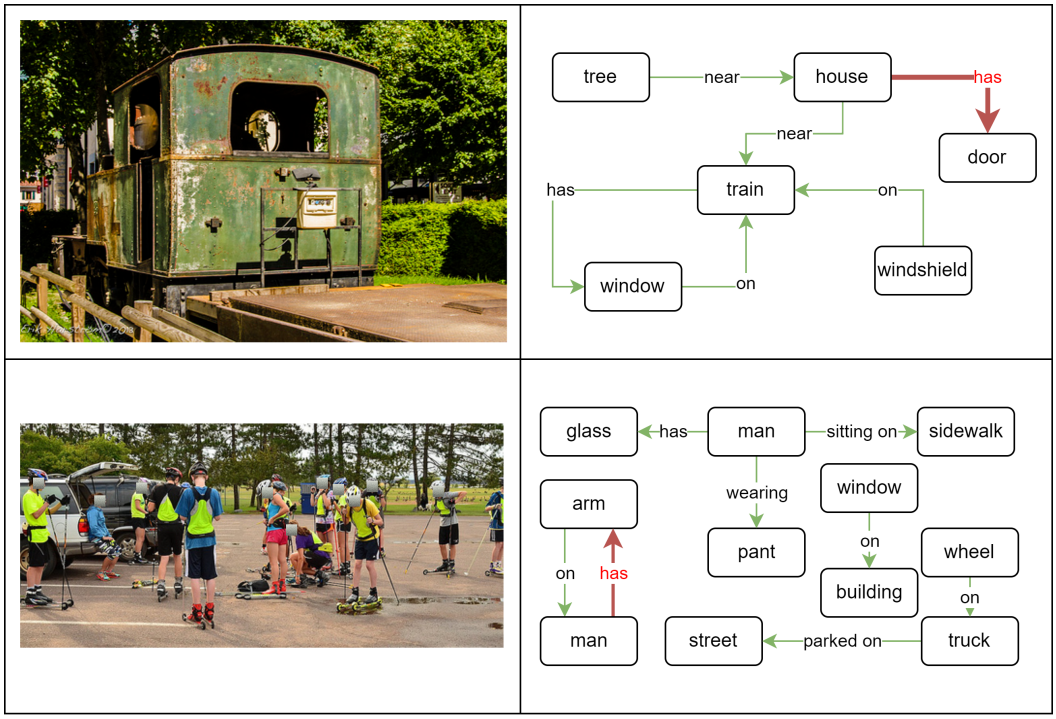



**Figure 9.** Predicate classification by EDET. Shows that EDET is able to mine deep semantic associations. The green arrow represents correct prediction results, while the red arrow represents incorrect prediction results.

The results show that EDET can generate excellent scene parsing in the scene graph predicate classification task. R@K means the recall rate of the top K prediction results.

#### 4.4.2. Image Captioning

In the section above, two methods to construct relationship edges within the knowledge graph were proposed for the image captioning task, respectively—the unbiased relative position relationship and the biased relationship obtained by previous predicate classification model. Although the latter carries the biased information in other models, the semantic information of the connotation is richer than that of the former, and it is also more logically instructive for the generated semantic description. The same model was used to train on two knowledge graphs, and the model was marked as EDET and EDET+, respectively. '+' here means that it carries the data on the other training sets (the relationship information from the Visual Genome dataset). The performance comparison of the two models with others on the Flickr30k dataset is shown in Table 5. Figure 10 shows examples of the scene image description generated by the two models.

**Table 5.** Performanceof EDET and EDET+ in image captioning. EDET+ means using extra data from the trained EDET model in the previous experiment. × means that the generation of result does not rely on extra training data, while ✓ means using extra training data.

| Method | Extra Training Data | BLEU-4 | CIDEr | METEOR | SPICE |
|---|---|---|---|---|---|
| BRNN [26] | × | 15.7 | 24.7 | 15.3 | - |
| MetaLM [27] | × | - | 43.3 | - | 11.7 |
| Cornia [28] | ✓ | 21.3 | 46.4 | 20.0 | - |
| SimNet [29] | ✓ | 25.1 | 65.0 | 22.1 | 16 |
| Unified VLP [30] | × | 30.1 | 67.4 | 23.0 | 17 |
| **EDET** | × | 30.7 | 66.7 | - | 15.3 |
| **EDET+** | ✓ | **33.2** | **71.6** | - | **21.2** |

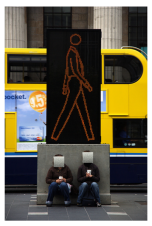
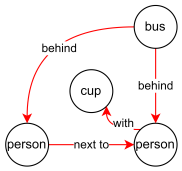
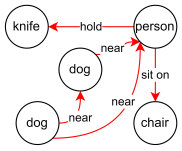
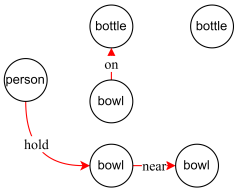
| image | graph sturcture | ground truth | image captioning |
|---|---|---|---|
|  |  | #0 Two ladies in jeans and brown sweaters and sitting under a large walk sign .<br>#1 Two woman sitting under a L.E.D. sign of a person walking .<br>#2 Two women are sitting on a stone bench drinking coffee .<br>#3 Two women sitting under a lit sign on the sidewalk .<br>#4 Two people are sitting below the walking sign . | **EDET:**<br>Two people are sitting on a road and a bus.<br>**EDET+:**<br>Two women are sitting on a road with a bus behind. |
|  |  | #0 Two dogs are waiting with their mouths open by a person sat in a dining room chair .<br>#1 Two dogs begging a partially seen person for food at a kitchen table .<br>#2 Two dogs are barking in a kitchen while a person sits at a table .<br>#3 Two dogs are waiting to be fed in the kitchen .<br>#4 Two dogs bark at a man in a chair . | **EDET:**<br>A man is sitting on a chair in the kitchen and two dogs.<br>**EDET+:**<br>A man wearing a black shirt sitting on a chair and two dogs. |
|  |  | #0 A man with dark brown hair wearing a gray shirt is making something in the kitchen with a plate of lemons next to him .<br>#1 A man in a gray T-shirt is zesting lemons into a turquoise-and-olive-green bowl .<br>#2 A man in a gray shirt with a beard prepares a recipe .<br>#3 A man staring into the camera as he cuts lemons .<br>#4 A man wearing a gray shirt grating lemon rinds . | **EDET:**<br>A man wearing a gray shirt and bowl and bottles.<br>**EDET+:**<br>A man wearing a gray shirt and bottles on a bowl. |

**Figure 10.** Image captioning by EDET and EDET+.

According to the data in Table 5, it can be seen that the EDET model also has an excellent performance in the image captioning generation task, especially in the SPICE index. The reason for the improvement of the SPICE index is that the description of EDET itself is generated based on the data of the graph structure, and the basis of the SPICE index is to use a semantic concept tree, which is similar to the graph structure. It proves that EDET generates a description that is smoother and more consistent with the grammatical structure than other models. The results shown in Figure 10 also confirm this. It can be found in Table 5 that, after using the trained model in the predicate classification task to generate biased semantic relationships, both model performance and semantic fluency were improved and it basically completely expresses the scene information contained in the multi-modal graph.

The bottleneck of EDET is not the model structure, but the object detection in the pre-processing. There are only 90 fixed classes of entities that can be detected by Mask R-CNN, which is far less than the number of entities that appear in practice, and limits the performance of our model.

### *4.5. Entity Descriptor and EDET in Industrial Application*

The previous experiment confirmed the feasibility and effectiveness of the Entity Descriptor and EDET model on the public dataset; this will appear in the section on the industrial scene parsing-related tasks with discussion of the application of EDET in the actual industrial scenarios.

#### 4.5.1. Construction of Industrial Knowledge Graph

First, it was necessary to clarify the constructed graph-oriented usage scene. In this case, the facing industrial scene was the fault analysis in the production process. Rapid warning and preliminary diagnosis of the fault information on the production line can prevent the occurrence of a large range of faults and, according to the preliminary diagnosis results, it can effectively shorten the time and labor cost of troubleshooting and improve the production efficiency.

Secondly, after clarifying the use scenarios, the massive data were analyzed and classified to find out the objects that meet the application requirements so as to avoid the

impact of excessive redundant industrial data. By analyzing the database and comparing the structure of the data table, combined with the project requirements, the final selected entities and relationships are shown in Tables 6 and 7, respectively.

**Table 6.** Selected entities and their attributes on industrial knowledge graph by attribute selection conditions.

| Entity Number | Entity Type | Attributes |
|---|---|---|
| 0 | Equipment | Parts, sub-images, produced by, etc. |
| 1 | Order | Standardized work order content |
| 2 | Production Line | Workflow |
| 3 | Work Center | Location |
| 4 | Workflow | Workflow code |
| 5 | Parts | Parts and components |
| 6 | Product | Parts, subordinate categories, sub-images, etc. |
| 7 | Parameters | Category |
| 8 | Fault | Fault code |
| 9 | Fault Phenomenon | Context description |
| 10 | Check Steps | Step description |
| 11 | Solution | Content of solution |

**Table 7.** Relationships on industrial knowledge graph.

| Relation Number | Relation Definition | Relation Label | Relation Constraint ([A:B], Entity Number) |
|---|---|---|---|
| 0 | A has part B | Has Part | [0 or 5:5] |
| 1 | A is order of B | Order of | [1:6] |
| 2 | A is produced on B | Produced On | [1:2] |
| 3 | A has process B | Has Process | [2:4] |
| 4 | A work on B | Work On | [4:3] |
| 5 | A has fault B | Has Fault | [4:8] |
| 6 | A has parameters B | Params | [0 or 4 or 6:7] |
| 7 | A has phenomena B | Occur | [0 or 4 or 6:9] |
| 8 | A is the performance of B | Has Phenomena | [9:8] |
| 9 | A has solution B | Solution Is | [8:11] |
| 10 | B may be the solution of A | May Solution | [9:11] |
| 11 | A has step B | Start Step | [9:10] |
| 12 | B is the next step of A | Nest Step | [10:10] |
| 13 | B is the final result of A | Final Solution | [10:11] |

After determining the entity and the entity relationship, the attribute selection condition was used to filter the attribute of the entity. For example, each equipment entity has a standard value in the production process, which can be regarded as an attribute of the equipment entity according to the common condition. However, some devices have more than one standard value because they measure multiple locations or types of parameters during operation. According to the unique interaction condition, multiple standard value nodes owned by the device entity need to be fused; a standard value attribute node was used to represent it uniformly and the representation method can use strings, key-value pairs, etc. At the same time, in the scene, there are situations where parts and products have the same name or the product is even a part of a device and parts are components of the device entity. According to the limited scope condition, in the process of constructing and applying the map, the node needs to be split; one exists as a component attribute and the other exists as a product entity node.

Through the analysis of more than tens of millions of data points, a large-scale knowledge graph with expansion, and in line with the practical application needs, was established. Parts of the knowledge graph are shown in Figure 11.

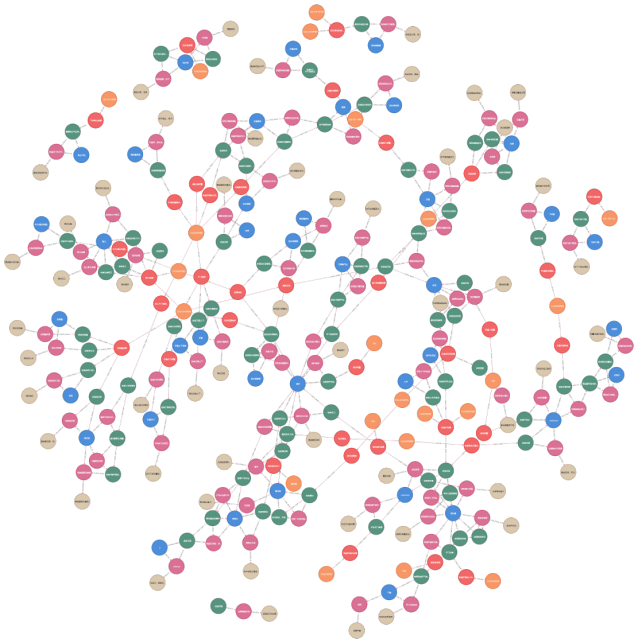


**Figure 11.** Industrial knowledge graph constructed by attributes selection conditions. Different colors represent different types of entities or attributes.

#### 4.5.2. Application of EDET on the Fault Analysis

In order to monitor the production line, a month of data were obtained before the fault parameters as a dataset, and each fault was divided into three stages: "fault", "the fault will happen", "normal". After marking the dataset, the fault analysis based on the deep learning model was transformed into a graph classification task based on a knowledge graph.

Even if the same equipment is available between different production lines, their functions and monitoring parameters in the manufacturing process of different production lines may be different, so separate modeling and monitoring should be conducted for each production line. In the implementation of the monitoring of each production line, the operation process is fixed, namely, the following monitoring of different production lines is implemented by the system automation: obtain production line equipment data, obtain the attribute information of equipment models, components, and parameters using the entity node vector representation, input the whole production line serialized vector into the EDET model of native transformer architecture, and obtain the classification results through MLP classification. After one month of historical data training, the model is put online and, when a certain amount of real-time monitoring results and feedback information are accumulated after the launch, the new data are collected to continue the EDET model so as to achieve the effect of lasting training, continuous optimization, and man–machine collaboration.

In order to achieve the purpose of rapid response of the system, the system also needs to build an offline knowledge base. As shown in Figure 12, the parameters of the Entity Description substructure on each scene are stored separately as a model.

When the system listens to the production line, the monitoring data of the same device are also two different entities with independence and differentiation at different times, while the Entity Descriptor can retain the independent separability of the entity. This means that when the scene needs to realize the function of rapid response, we only need to input graph data to the Entity Descriptor stored in an offline knowledge base to generate a representation vector, and then the similarity between different scenes can be calculated by their representation. However, it should be noted that the parameters in the Entity Descriptor are also constantly updated under the action of continuous training. After each update, the parameters stored in the offline knowledge base also need to be changed accordingly.

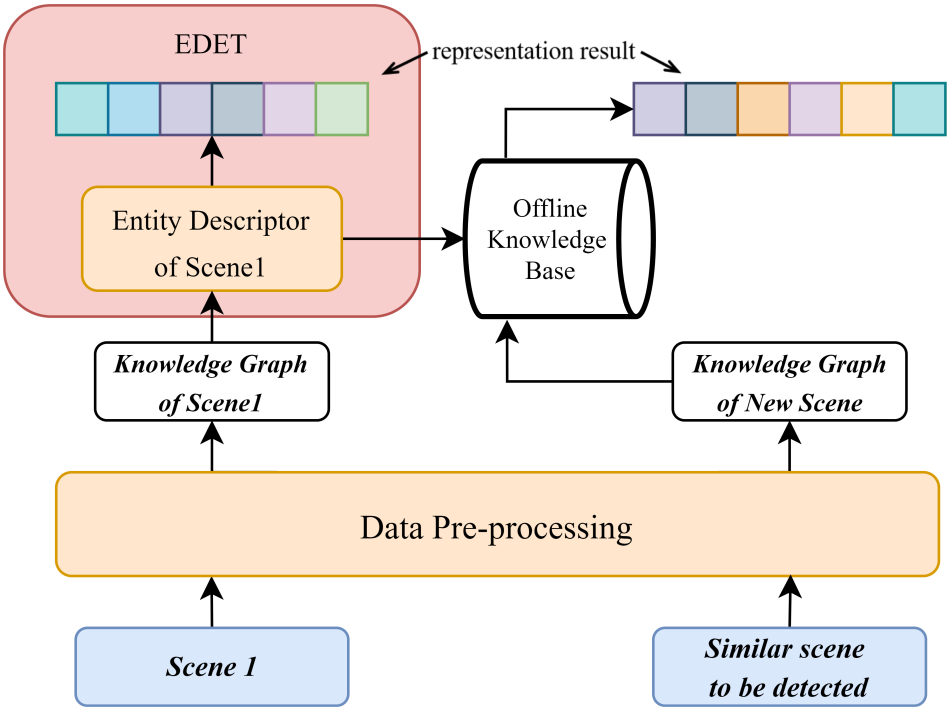



**Figure 12.** Realization of offline knowledge base.

## 5. Conclusions

This paper was oriented to the scene parsing task. In view of the powerful representation ability of knowledge graphs, we proposed a new multi-stage process of scene parsing based on a multi-modal knowledge graph and a deep learning model. By using the mentioned process to solve the scene parsing task and in the process of repeatedly building the knowledge graph, we found that, in the specific scene, the selection of graph entities is not controversial, while the selection of graph attributes is not only faced with massive data but also profoundly affects the independent separability of the entity and the quality of the knowledge graph. Therefore, through observation and summarizing the experience, this paper obtained three attribute selection conditions for the construction of the knowledge graph, which provide guidance for multiple operations of attribute nodes.

After obtaining a high-quality knowledge graph through attribute selection conditions, we set out to preserve the independence and distinguishability of entities in the real world when representing entity nodes. For this reason, the Entity Descriptor representation method based on a knowledge graph structure was proposed. This method is simple to implement and can be plug-and-play with only a few changes to various models. The model embedded with the Entity Descriptor is called EDET. In the follow-up experiments, the effectiveness of the EDET model in solving problems was proved in terms of predicate classification and image captioning.

**Author Contributions:** Conceptualization, S.M., W.W. and Y.Z.; methodology, S.M. and W.W.; software, S.M. and Z.Y.; validation, S.M., W.W. and Z.Y.; formal analysis, Y.Z.; investigation, S.M. and Z.Y.; resources, W.W. and Y.Z.; data curation, S.M. and Z.Y.; writing—original draft preparation, S.M.; writing—review and editing, W.W. and Y.Z.; visualization, S.M.; supervision, W.W.; project administration, W.W.; funding acquisition, W.W. and Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Science and Technology Innovation 2030—Major Project of "New Generation Artificial Intelligence" granted by Ministry of Science and Technology, China, grant number 2020AAA0109300. and 2022 Major RD Special 03 and 5G Projects of Jiangxi Provincial Department of Science and Technology, China, grant number 20224ABC03A15.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References


1. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1–9.
2. Chang, D.; Chen, M.; Liu, C.; Liu, L.; Li, D.; Li, W.; Kong, F.; Liu, B.; Luo, X.; Qi, J.; et al. Diakg: An annotated diabetes dataset for medical knowledge graph construction. In Proceedings of the Knowledge Graph and Semantic Computing: Knowledge Graph Empowers New Infrastructure Construction: 6th China Conference, CCKS 2021, Guangzhou, China, 4–7 November 2021; Proceedings 6; Springer: Berlin/Heidelberg, Germany, 2021; pp. 308–314.
3. Wang, L.L.; Lo, K.; Chandrasekhar, Y.; Reas, R.; Yang, J.; Eide, D.; Funk, K.; Kinney, R.; Liu, Z.; Merrill, W.; et al. Cord-19: The COVID-19 open research dataset. *arXiv* **2020**, arXiv:2004.10706v4.
4. Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; Bronstein, M.M. Geometric deep learning on graphs and manifolds using mixture model cnns. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5115–5124.
5. Giles, C.L.; Bollacker, K.D.; Lawrence, S. CiteSeer: An automatic citation indexing system. In Proceedings of the Third ACM Conference on Digital Libraries, Pittsburgh, PA, USA, 23–26 June 1998; pp. 89–98.
6. Liu, Y.; Li, H.; Garcia-Duran, A.; Niepert, M.; Onoro-Rubio, D.; Rosenblum, D.S. MMKG: Multi-modal knowledge graphs. In Proceedings of the European Semantic Web Conference, Portorož, Slovenia, 2–6 June 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 459–474.
7. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
8. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
9. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
10. 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.
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
12. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
13. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
14. Almeida, F.; Xexéo, G. Word embeddings: A survey. *arXiv* **2019**, arXiv:1901.09069.
15. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
16. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ade20k dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 633–641.
17. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.J.; Shamma, D.A.; et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.* **2017**, *123*, 32–73. [CrossRef]
18. Young, P.; Lai, A.; Hodosh, M.; Hockenmaier, J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Trans. Assoc. Comput. Linguist.* **2014**, *2*, 67–78. [CrossRef]
19. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
20. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
21. Yang, J.; Lu, J.; Lee, S.; Batra, D.; Parikh, D. Graph r-cnn for scene graph generation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 670–685.
22. Zellers, R.; Yatskar, M.; Thomson, S.; Choi, Y. Neural motifs: Scene graph parsing with global context. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5831–5840.
23. Cong, Y.; Ackermann, H.; Liao, W.; Yang, M.Y.; Rosenhahn, B. Nodis: Neural ordinary differential scene understanding. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XX 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 636–653.
24. Tang, K.; Zhang, H.; Wu, B.; Luo, W.; Liu, W. Learning to compose dynamic tree structures for visual contexts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6619–6628.
25. Lin, X.; Ding, C.; Zeng, J.; Tao, D. Gps-net: Graph property sensing network for scene graph generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3746–3753.

26. Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.
27. Hao, Y.; Song, H.; Dong, L.; Huang, S.; Chi, Z.; Wang, W.; Ma, S.; Wei, F. Language models are general-purpose interfaces. *arXiv* **2022**, arXiv:2206.06336.
28. Cornia, M.; Baraldi, L.; Serra, G.; Cucchiara, R. Paying more attention to saliency: Image captioning with saliency and context attention. *ACM Trans. Multimed. Comput. Commun. Appl. TOMM* **2018**, *14*, 1–21. [CrossRef]
29. Liu, F.; Ren, X.; Liu, Y.; Wang, H.; Sun, X. simnet: Stepwise image-topic merging network for generating detailed and comprehensive image captions. *arXiv* **2018**, arXiv:1808.08732.
30. Zhou, L.; Palangi, H.; Zhang, L.; Hu, H.; Corso, J.; Gao, J. Unified vision-language pre-training for image captioning and vqa. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 13041–13049.