

Article

Research on Driving Scenario Knowledge Graphs

Ce Zhang, Liang Hong, Dan Wang, Xinchao Liu, Jinzhe Yang and Yier Lin *

College of Mechanical Engineering, Tianjin University of Science and Technology, Tianjin 300222, China; zhangce@mail.tust.edu.cn (C.Z.)

* Correspondence: yier.lin@tust.edu.cn; Tel.: +86-185-2213-3115

Abstract: Despite the partial disclosure of driving scenario knowledge graphs, they still fail to meet the comprehensive needs of intelligent connected vehicles for driving knowledge. Current issues include the high complexity of pattern layer construction, insufficient accuracy of information extraction and fusion, and limited performance of knowledge reasoning models. To address these challenges, a hybrid knowledge graph method was adopted in the construction of a driving scenario knowledge graph (DSKG). Firstly, core concepts in the field were systematically sorted and classified, laying the foundation for the construction of a multi-level classified knowledge graph top-level ontology. Subsequently, by deeply exploring and analyzing the Traffic Genome data, 34 entities and 51 relations were extracted and integrated with the ontology layer, achieving the expansion and updating of the knowledge graph. Then, in terms of knowledge reasoning models, an analysis of the training results of the TransE, Complex, Distmult, and Rotate models in the entity linking prediction task of DSKG revealed that the Distmult model performed the best in metrics such as hit rate, making it more suitable for inference in DSKG. Finally, a standardized and widely applicable driving scenario knowledge graph was proposed. The DSKG and related materials have been publicly released for use by industry and academia.

Keywords: intelligent traffic; knowledge graph; hybrid methods; driving scenarios; ontology



Citation: Zhang, C.; Hong, L.; Wang, D.; Liu, X.; Yang, J.; Lin, Y. Research on Driving Scenario Knowledge Graphs. *Appl. Sci.* **2024**, *14*, 3804. <https://doi.org/10.3390/app14093804>

Academic Editor: Francesco Liberato Cappiello

Received: 28 March 2024

Revised: 17 April 2024

Accepted: 22 April 2024

Published: 29 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, significant progress has been made in the field of autonomous driving, largely attributed to the development and application of artificial intelligence technologies, particularly algorithms such as deep learning and reinforcement learning. These algorithms play crucial roles in perception, decision-making, and control, enabling vehicles to accurately perceive and adapt to their surrounding environments.

Despite remarkable achievements, the comprehensive application of advanced autonomous driving technologies still faces numerous challenges. This is mainly due to the limitations of artificial intelligence algorithms in interpretability, understanding complex scenarios, and handling unknown situations [1]. When faced with real-world situations that differ from the training data, current artificial intelligence algorithms may encounter significant setbacks, potentially leading to accidents involving autonomous vehicles. For instance, several fatal accidents involving Tesla's autonomous driving vehicles can largely be attributed to failures in their perception systems [2,3]. Similarly, the fatal collision between an autonomous Uber vehicle and a pedestrian also exposes shortcomings in autonomous driving in terms of perception and prediction [4]. Additionally, artificial intelligence algorithms rely on vast training datasets, requiring them to process large volumes of example data to achieve high accuracy, which sharply contrasts with the efficiency of humans in integrating new information through single instances of learning. In this context, understanding context becomes particularly important for agents to handle unknown situations, provide feedback to users, and understand their own functionality.

Scene understanding requires systems to possess a profound semantic understanding of entities and their relationships in complex physical and social environments. To meet this

requirement, a feasible approach is to represent entities and their relationships in scenes as knowledge graphs (KGs) [5]. By constructing scene knowledge graphs, systems can better predict entity behaviors, thereby enhancing scene understanding capabilities [6,7]. For example, consider a scenario where an autonomous vehicle is navigating through a complex intersection. Traditional visual perception systems may identify traffic signals, pedestrians, and other vehicles. However, relying solely on this information may not accurately predict the next actions in certain situations. If the system can leverage knowledge graphs to understand the structure and rules of the intersection, it can improve predictions. It can identify relationships between traffic signals and pedestrians, as well as the traffic rules among different vehicles. This enables more accurate predictions of when to stop, when to turn, or when to accelerate through. This knowledge-injection-learning-based approach can compensate for the limitations of traditional visual perception systems. It improves the performance and safety of autonomous driving systems in complex scenes. Therefore, knowledge graph construction based on knowledge injection learning holds tremendous potential in addressing the complex technical challenges of scene understanding in autonomous driving.

However, the application of knowledge graphs also faces a series of challenges and issues, such as the cost of knowledge acquisition, the uncertainty of knowledge representation, and the timeliness of knowledge updates. Among these, the cost of knowledge acquisition is a significant consideration factor because building and maintaining a complete and accurate knowledge graph require substantial human and time investments. Additionally, the uncertainty of knowledge representation poses another challenge because real-world knowledge is often fuzzy, uncertain, and may evolve over time and in different environments. There is also a high demand for timely knowledge updates because autonomous driving systems need to promptly acquire the latest information to adapt to continuously changing road conditions and traffic rules.

The main contributions of this paper include:

- Proposing a more efficient method for knowledge acquisition and updating, including automated extraction and updating of knowledge from various sources to reduce manpower and time costs.
- Constructing a standardized and widely applicable knowledge graph for driving scenarios, and making it open-source.
- Validating the inference effects of different knowledge embedding models in DSKG to discover new knowledge and confirm the most effective embedding models for scene understanding.

The subsequent sections of this paper will unfold in the following order: Section 2 will review the construction methods of knowledge graphs. Section 3 will detail the construction process of DSKG and provide in-depth analysis and discussion of all evaluation results. Finally, Section 4 will summarize the main conclusions of this paper and outline future research directions.

2. Knowledge Graph

The concept of a knowledge graph [8] was formally introduced by Google in 2012, aiming to depict various concepts, entities, and their relationships in the real world using graph theory. A knowledge graph is composed of entities, relationships, and facts, formalized as $G = (E, R, T)$. Here, E represents the set of entities, R represents the set of relationships, and T represents the set of triples $(h, r, t) \in T$, where $h \in E$ denotes the head entity, $t \in E$ denotes the tail entity, and $r \in R$ denotes the relationship. In this structure, entities are interconnected through relationships, forming a networked knowledge system. Compared to traditional text and tabular representations, knowledge graphs, with their intuitive and efficient characteristics, can clearly display entity attributes and their connections, thereby enabling tasks such as intelligent search, question reasoning, and recommendations [9–11]. Furthermore, knowledge graphs often utilize a resource description framework (RDF) and web ontology language (OWL) to represent and define the structure of the data, thereby

enhancing interoperability and reasoning capabilities by facilitating data exchange and specifying concepts, relationships, and constraints within a domain.

The construction of a knowledge graph involves two main layers: the schema layer and the data layer [12]. The ontology, as the schema layer of the knowledge graph, constrains the data layer through rules and axioms defined by the ontology. The data layer, situated below the ontology layer, stores information in the form of triple tables, attribute tables, etc., in a graph database. For the schematic diagram of a domain knowledge graph structure, please refer to Figure 1.

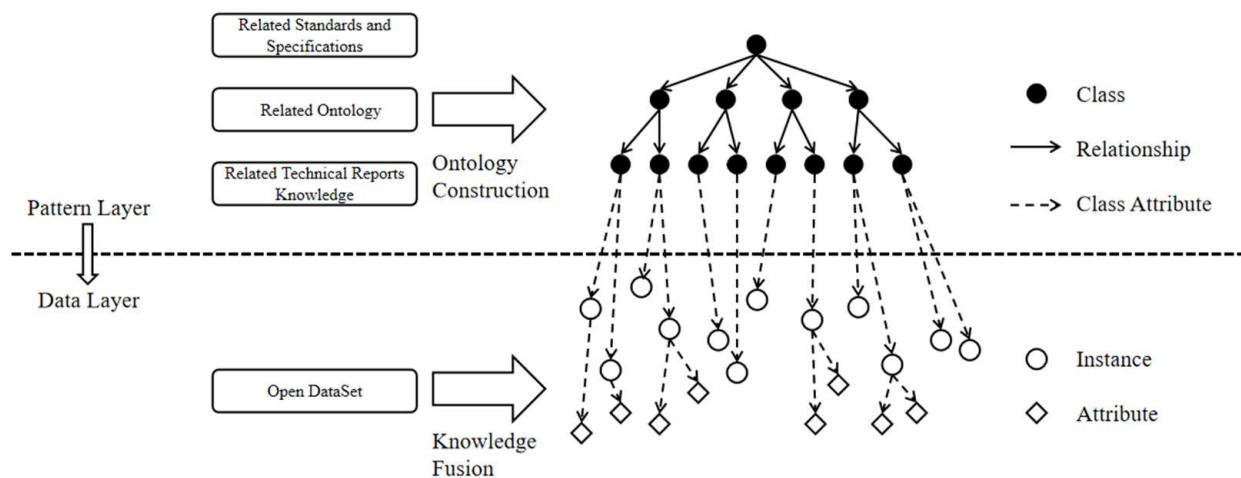


Figure 1. Conceptual representation of knowledge graph structure.

The construction of a knowledge graph can be mainly divided into two categories: top-down and bottom-up approaches [13]. The top-down approach establishes the schema layer first, guiding the construction of the data layer, directly extracting ontology and schemas from structured data sources, and incorporating them into the knowledge base. This method performs well in displaying hierarchical relationships between concepts but requires a high degree of human involvement and has limited schema layer updates, making it more suitable for constructing knowledge graphs with smaller scales and clear hierarchical structures. If a domain knowledge graph has a small scale and a clear knowledge system, the top-down approach can be considered. The bottom-up approach, on the other hand, first constructs the data layer, integrates the acquired knowledge, extracts and filters ontology patterns from the data, and then manually selects new patterns with high confidence to add to the knowledge base. This method updates rapidly and is suitable for large-scale knowledge graph construction but may contain more noise. Typically, the bottom-up approach is chosen for constructing general-purpose knowledge graphs due to the richness and complexity of general knowledge, which makes defining patterns manually challenging.

To balance the advantages of both approaches, we adopted a hybrid knowledge graph construction approach. The overall process is illustrated in Figure 2. Firstly, core knowledge is collected from the public domain of driving scenarios, and the domain concept system is summarized to construct the top-level ontology of the multidimensional hierarchical knowledge graph. Secondly, knowledge extraction is performed using massive structured data to extract entities, attributes, and relationships, forming a small-scale graph structure relevant to new knowledge. Next, the schema layer is integrated to construct the knowledge graph for intelligent connected vehicle driving scenarios. Subsequently, based on existing data in the knowledge graph, knowledge embedding reasoning is conducted, and the quality of the reasoning results is evaluated to expand and enrich the knowledge graph.

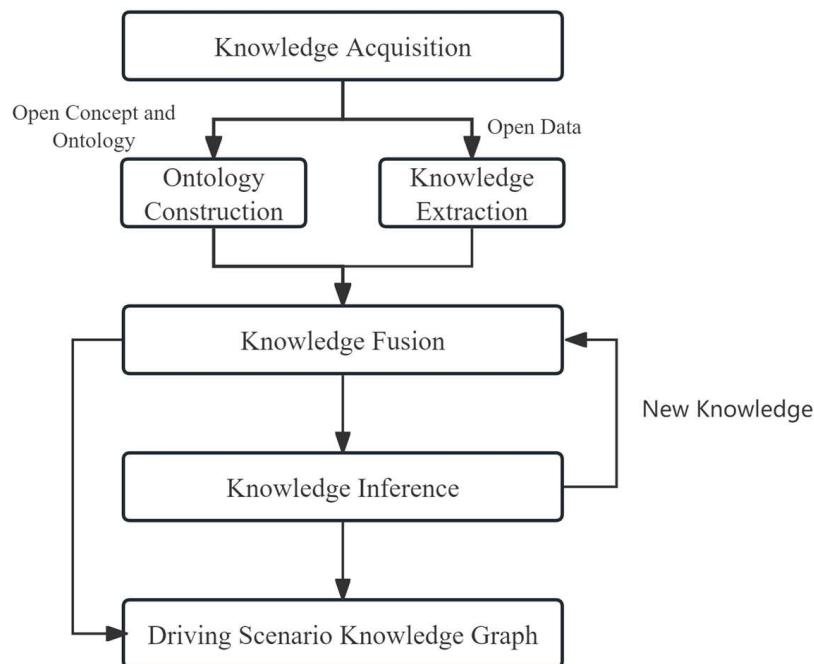


Figure 2. Construction framework of the driving scenario knowledge graph.

3. Methodology: DSKG Construction

3.1. Knowledge Acquisition

The goal of knowledge acquisition is to gather, integrate, and analyze core knowledge in the public domain, thereby constructing a comprehensive and reliable knowledge system. The following will delve into the analysis and discussion of the core knowledge of intelligent connected vehicle driving scenarios collected from the aspects of concepts, ontology, and data.

3.1.1. Domain Concepts

A driving scenario [14] refers to the comprehensive and dynamic description of the interactive process between intelligent connected vehicles and elements in the driving environment (such as other vehicles, roads, traffic facilities, and weather conditions) within a specific time and space range. Schuld's research [15,16] introduces the concept of using a layered model to construct scene and environment descriptions, divided into four layers: base road network, situation-specific adaptations of the base road network, actors and their control, and environment conditions. In the PEGASUS project [17], the layered scene concept [16] is applied to highway scenarios. Bagschik et al. [18] and Bock et al. [19] subsequently introduced the fifth and sixth layers, forming the six-layer model for highway scenarios. The road layer defines road layout and topology, while the traffic infrastructure layer includes structural boundaries, traffic signs, and other elements. In subsequent work [20,21], a new third layer was separated from the second layer to describe temporary changes, the fourth layer was named “objects”, the fifth layer represented weather conditions, and the sixth layer represented digital information. Given the emphasis of the PEGASUS project [17] on highway applications, Scholtes et al. [22] proposed a six-layer model for urban scenarios. Building upon the six-layer external scene construction performed by the CAICV-SOTIF Working Group, a seventh layer was added representing the vehicle state [23], to more accurately describe in-car information. Figure 3 illustrates the iterative process of the scene hierarchical concept, from the initial four layers to the latter seven layers, reflecting the ongoing deepening and refinement of driving scenario descriptions.

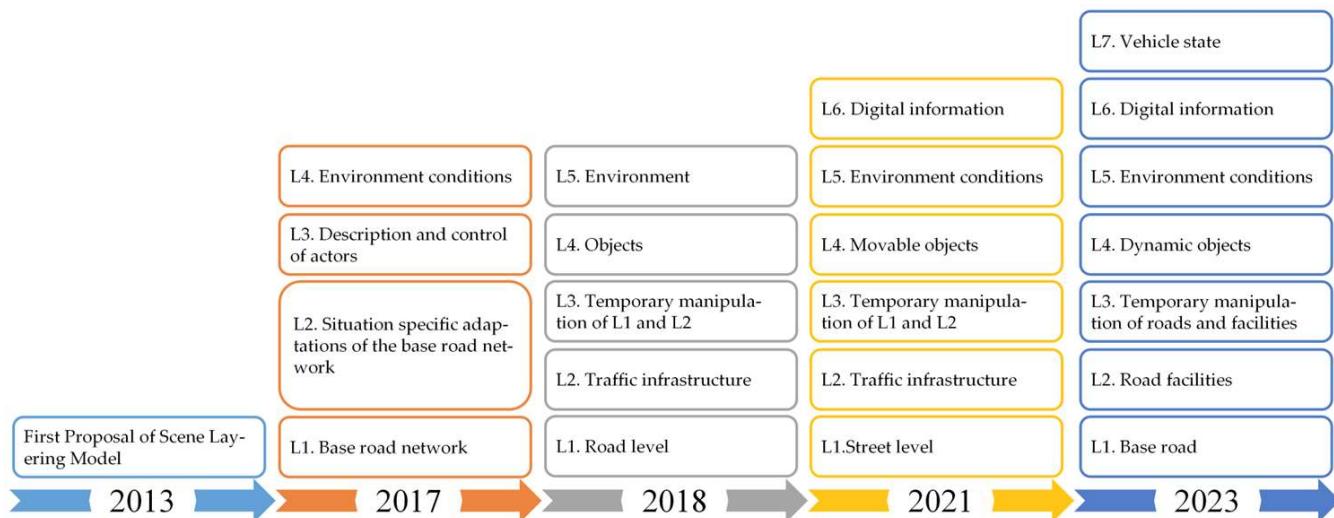


Figure 3. Iterative process of scene hierarchical concept evolution.

3.1.2. Domain Ontology

An ontology is a formalized model for describing domain knowledge, defining semantic associations between entities, properties, and relationships within the domain. In the work of Bagschik et al. [18], an ontology describing simple highway scenes based on a set of predefined keywords was proposed. In subsequent work, Menzel et al. [24] extended this concept to generate OpenSCENARIO and OpenDRIVE scenes. Tahir et al. [25] proposed an ontology focusing on urban intersection scenarios, while also addressing evolving weather conditions. Hermann et al. [26] proposed an ontology for dataset creation, focusing on pedestrian detection, including occlusions of pedestrians. Their ontology, inspired by the PEGASUS model [17], consists of 22 sub-ontologies. It can describe various scenarios and convert them into simulations. However, due to the lack of detailed description and unavailability, it is unclear whether a separate ontology is needed for each frame or if the ontology itself can describe temporal scenes. The Association for Standardisation of Automation and Measuring Systems (ASAM) [27] developed the OpenX Ontology, aiming to provide common definitions, attributes, and relationships for central concepts in the ASAM OpenX standard in the field of road traffic. The ontology is mainly divided into three modules: core ontology, domain ontology, and application ontology. The domain ontology defines the central concepts of road traffic and includes three layers: environmental conditions, road topology and traffic infrastructure, traffic participants and behaviors. Bogdolt et al. [28] proposed an extensive framework for developing edge hazardous scenarios. Westhofen et al. [29] also proposed an ontology for urban automotive city traffic (A.U.T.O.), to discover more trigger events. Table 1 summarizes the scope and attributes of the previously introduced ontologies, and whether the ontology itself is openly published.

Table 1. Ontologies related to driving scenarios.

Authors	Year	Scene Category	Open Status
Bagschik et al. [24]	2018	Highway	-
Menzel et al. [25]	2019	Highway	-
Tahir et al. [26]	2022	Urban	-
Hermann et al. [27]	2022	Urban	-
ASAM [28]	2022	Full scene	✓
Bogdolt et al. [29]	2022	Full scene	✓
Westhofen et al. [30]	2022	Urban	✓

3.1.3. Domain Data

In the field of autonomous driving, there are numerous high-quality structured datasets [30–36]. These datasets contain abundant autonomous driving scene data and annotation information, offering vital support for research and development in autonomous driving technology. Upon thorough analysis of these annotations, we observed that newer datasets often represent advanced technological progress and more comprehensive data collection strategies. For instance, the A2D2 and BDD100K datasets excel in diverse concept categories, encompassing a wide range of scenes and traffic conditions. Conversely, the NuScenes and Traffic Genome datasets excel in providing detailed attribute or relationship information, enhancing our ability to accurately describe and understand scenes. Notably, due to the significantly higher number of relationship labels in the Traffic Genome dataset compared to others, we chose it as the primary target for knowledge extraction. Detailed statistics on the quantities of various elements are provided in Table 2.

Table 2. Datasets related to driving scenarios.

Dataset	Year	Number of Concept Classes	Number of Attributes or Relationships
KITTI [30]	2013	5	-
BDD100K [31]	2018	40	-
NuScenes [32]	2019	23	5
A2D2 [33]	2019	52	-
Waymo [34]	2020	4	-
PandaSet [35]	2020	37	13
Traffic Genome [36]	2021	34	51

3.2. Construction of Ontology for Driving Scenarios

Constructing the domain ontology for driving scenarios involves considering the standardization of domain terms, the broad applicability of conceptual categories, the hierarchical structure of abstract concepts in the domain, and the definitions of relevant attributes of each concept and the relationships between concepts [37]. First, existing domain ontologies were examined. Among them, the terminology of the A.U.T.O. ontology, open-sourced by Westhofen et al. [29], was more standardized. Therefore, based on the reuse of the A.U.T.O. ontology, further extensions and adjustments were made for a full-scenario perspective based on the domain knowledge obtained in this section. Additionally, during the subsequent knowledge extraction from structured data, the ontology was updated based on the fusion results of the small-scale knowledge graph, making it not only conform to the domain consensus but also adaptable for describing and representing full-scenario knowledge.

3.2.1. Definition of Classes and Their Hierarchical Structure

Referring to the widely applicable six-layer scene hierarchical concept, this section also divides the knowledge system of intelligent connected vehicle driving scenarios into six concept classes: Base Road, Roadside Facility, Temporary Change, Dynamic Objects, Environmental Conditions, and Digital Information. Among them, Base Road provides the foundational support for all scenario elements; Roadside Facility is built upon the road structure; Temporary Change describes transient adjustments within Basic Road and Roadside Facility; Dynamic Object acts as an intermediate layer, connecting static and dynamic scenario elements; Environmental Condition characterizes the environmental elements of the scenario and their impacts on the aforementioned concept and vehicle functions; and Digital Information describes all digital data-based information related to vehicles, infrastructure, or both. These concept classes are constructed hierarchically from the bottom layer to the top layer, forming a complete architecture. Each concept class contains specific subclasses, forming a parent-child inheritance relationship among them.

We utilize the Protégé (Version 5.5.0) ontology management software developed by Stanford University for ontology editing and management [38]. During the construction of the domain ontology, we manually input the relevant ontology of driving scenarios. Additionally, we employ the OWLViz module within the software to visually display the hierarchical structure of the domain ontology, as depicted in Figure 4.

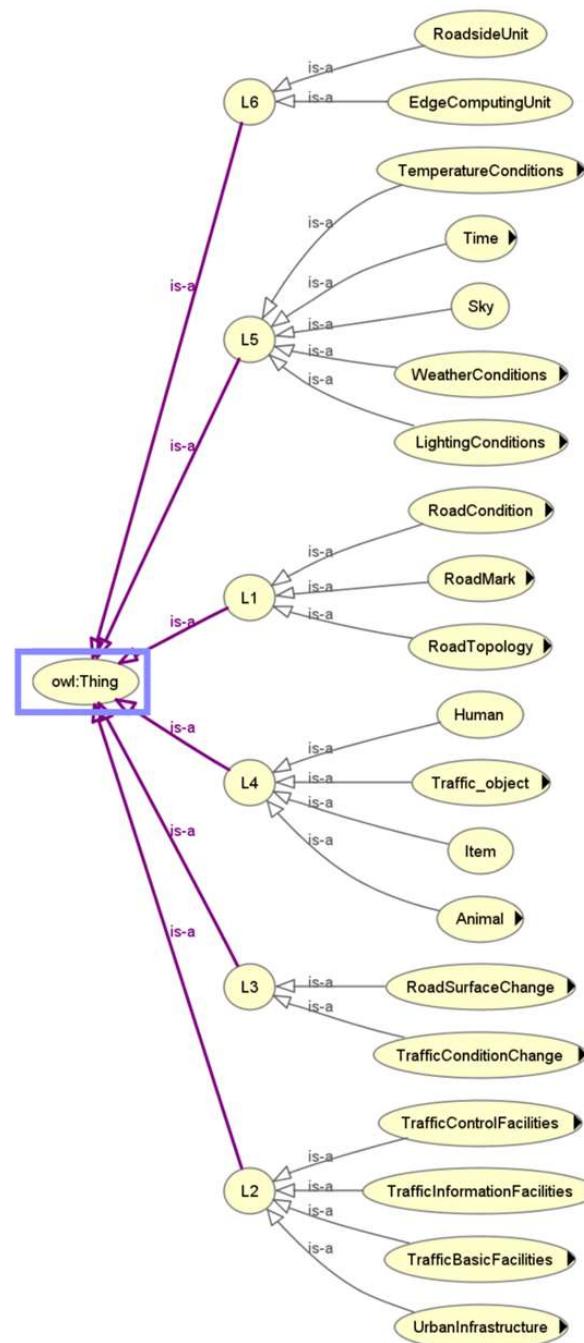


Figure 4. The hierarchical structure driving scenario.

The following briefly explains the core concept classes of the domain ontology of driving scenarios and their subclasses:

- (1) Base Road: This describes the road network and all permanent objects required for road traffic guidance. It can be divided into Road Topology, Road Surface Condition, and Traffic Markings.

Road Topology can be mapped to the road network and lanes. For complex road topologies such as intersections and roundabouts, they are recorded using a “road network + lanes” approach to avoid redundant descriptions caused by the direct classification of road structures. The road network consists of segments and connections, while lanes contain records of actual roads within each segment.

Road Surface Condition is divided into Road Surface Material and Road Surface Irregularity. The material of the road surface describes its physical characteristics, including features of materials such as asphalt, concrete, gravel, etc. Road surface irregularity describes whether the road surface has manholes, potholes, cracks, faults, depressions, speed bumps, etc., that affect driving conditions.

Traffic Markings are traffic indications drawn on the road surface, which can be divided into directional markings, prohibition markings, and warning markings based on their functions. Directional markings indicate markings on facilities such as roadways, directions of travel, road edges, and sidewalks. Prohibition markings indicate special regulations such as adherence to, prohibition, and restriction of road traffic, which drivers and pedestrians must strictly adhere to. Warning markings prompt drivers and pedestrians to understand special situations on the road, increase vigilance, and prepare for contingency measures.

(2) Roadside Facility: This describes all static objects usually placed near the road space rather than on the road. These static objects can be further decomposed into Urban Infrastructure, Traffic Infrastructure, Traffic Control Facilities, and Traffic Information Facilities. Urban Infrastructure includes buildings, vegetation, streetlights, fire hydrants, etc.; Traffic Infrastructure includes fences, tunnels, bridges, etc.; Traffic Control Facilities include traffic lights and roadblocks; Traffic Information facilities include traffic signs and information display screens.

(3) Temporary Change: This describes the non-persistent temporary changes of entities within the Base Road and Roadside Facility layers in specific scenarios. This layer does not introduce any new entity classes defined in the preceding layers but consists of temporary modifications of elements from Layers 1 and 2. This layer categorizes temporary road events into Road Condition Changes and Road Surface Changes. For each type of temporary event, information such as the lane position, starting point, and end point of the event is specified. Based on the description of the first layer, changes in road conditions can be divided into changes in road curvature, slope, coverage, lane width, number, centerline, etc. In addition to these temporary events closely related to the first layer of road structure, it also includes road surface changes caused by changes in weather conditions, such as dryness, moisture, icing, or reflective road surfaces.

(4) Dynamic Objects: This describes dynamic objects in the scenario that affect the occurrence of events. Dynamic objects can be classified from the perspective of vehicles, people, animals, and other objects. The movement of these objects evolves over time and can be described through trajectories.

(5) Environmental Conditions: This describes the natural environment in which the traffic scene is located. Weather Conditions include weather conditions, temperature, humidity, wind speed, wind direction, visibility, etc. Lighting Conditions include the position of light sources, the type of light sources, the intensity of light, the direction of light, and whether there is reflected light, etc.

(6) Digital Information: This describes all information based on digital data between vehicles, infrastructure, or both, including digital signals from information devices such as roadside units and edge computing units. The description of roadside units includes the change status of traffic signs and traffic lights. It is noteworthy that this layer describes the change status of traffic signs and traffic lights but is limited to the description of changeable information, while the respective objects themselves have already been placed in Layer 1. The description of edge computing units includes the change information of perception and control.

3.2.2. Definition of Class Attributes

After the completion of class definitions, the attributes of each class need to be defined. Since these classes have inheritance, subclasses can inherit attributes from parent classes. Therefore, placing these attributes in the most widely applicable application class, close to the top level, facilitates the efficiency of inheritance. Table 3 shows examples of attributes of some classes involved in this chapter.

Table 3. Example of concept class properties in the driving scenario domain.

Classes	Attributes
Lane	Length, width, direction, type, speed limit, etc.
Traffic markings	Type, color, width, length, shape, maintenance status, etc.
Vehicle	Type, brand, color, driving status, etc.
Person	Age group, gender, activity status, etc.
Object	Type, size, color, material, shape, mobility, etc.

3.2.3. Definition of Class Relationships

Based on the framework of the driving scenario domain ontology, the parent-child relationships of the hierarchical concepts of driving scenarios can be clearly defined. These relationships have transitivity, where subclasses can inherit parent class relationships and may also possess new specific relationships. Therefore, parent class relationships can be defined as common attributes, while subclass relationships are defined as specific relationships. The construction of conceptual relationships between classes in the driving scenario domain adopts a top-down approach. The ontology relationships describe the spatial, temporal, and semantic relationships between entities in traffic scenarios, as shown in Table 4.

Table 4. Example of relationships in the driving scenario domain.

Relationship Category	Meaning	Examples
Spatial Relationship	Describes the topological, directional, and metric relationships between concept classes.	The position dependency between lanes, road irregularities, and roadside facilities. The positional relationship between the driving direction of the vehicle and dynamic objects such as vehicles and pedestrians. The relative distance between the vehicle and roadside facilities, dynamic objects, and other elements.
Temporal Relationship	Describes the geometric topology information of time points or timelines between concept classes.	Whether the vehicle passes through the intersection during the green light phase of the traffic signal. Changes in environmental conditions over time during vehicle travel. The duration of vehicle parking in temporary parking areas.
Semantic Relationship	Describes the traffic connections between concept classes to express accessibility or restrictions, subject to constraints of time and space.	The restricted access rules for the tidal lane at different times. The relationship between people and vehicles in terms of driving or being driven.

To illustrate the hierarchical structure and associative relationships of the domain ontology further, we utilized the OntoGraf module of the Protégé for visualization. The complete ontology is shown in Figure 5. In the figure, solid lines represent the hierarchical relationships of the ontology, while dashed lines illustrate the associations between concept classes.

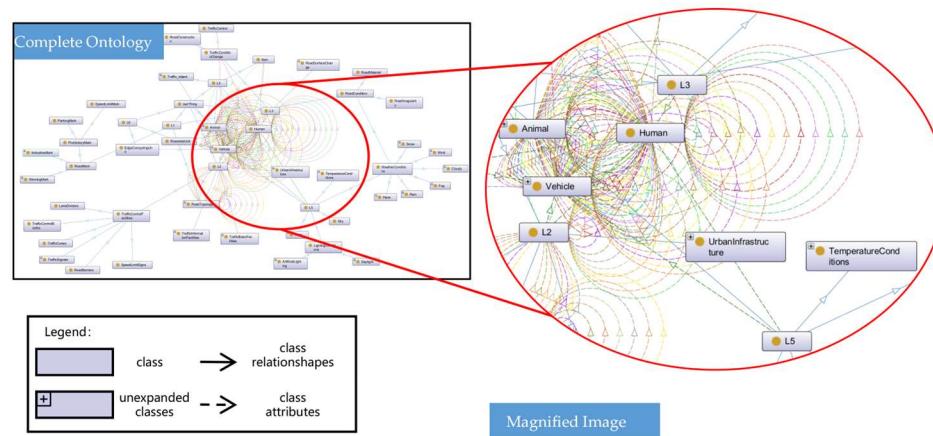


Figure 5. Ontology of driving scenario.

3.3. Knowledge Extraction and Fusion

In the construction of the domain knowledge graph generated in this chapter, the ontology layer mainly relies on manual summarization of human experience. Although it has good organization and theoretical foundation, its scale is limited and difficult to expand in bulk. The data layer, on the other hand, has a large scale and high information richness. However, due to the insufficient organizational structure of the network, the information density of the data layer is relatively low, and its utilization is limited. To address this issue, this section leverages the guiding role of the ontology layer to extract entity data from a large amount of standardized autonomous driving data, serving as the data layer of the driving scenario domain knowledge graph. By integrating the data layer with the ontology layer, the expansion and correction of the ontology layer are achieved, thereby realizing a "bottom-up" semi-automatic update of the ontology layer.

3.3.1. Data Extraction

Knowledge extraction refers to the process of accurately extracting and acquiring key information from massive data. According to the different characteristics of the data's intrinsic structure, knowledge extraction can be subdivided into three types: structured data extraction, semi-structured data extraction, and unstructured data extraction. The Traffic Genome dataset, as a typical structured dataset, has representative processing methods. When dealing with such data, we need to convert the original labeled HDF5 format scene graphs into a more standardized RDF format data. Figure 6 details how to convert the scene graph labels from the Traffic Genome dataset, thereby generating a small-scale but well-structured knowledge graph.

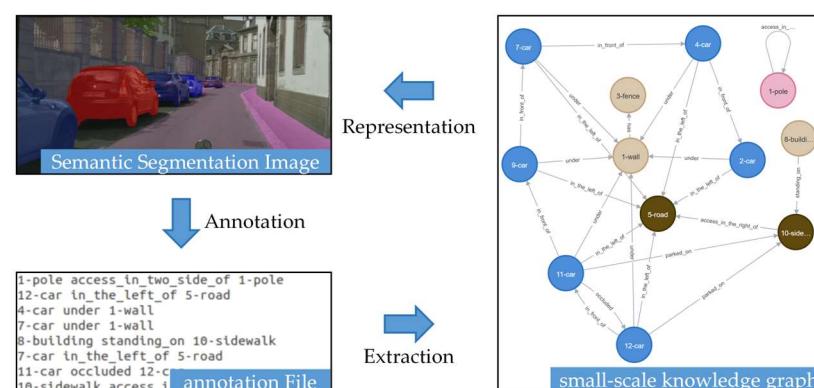


Figure 6. Extraction process of small-scale knowledge graph in the Traffic Genome dataset.

3.3.2. Knowledge Fusion

Knowledge fusion is a process that integrates, disambiguates, processes, verifies, and updates heterogeneous data, information, methods, experiences, and human thoughts into a high-quality knowledge base through high-level knowledge organization under the same framework. Considering that this project involves the fusion between the top-level ontology and a small-scale knowledge graph, a text-based approach can be used for entity matching.

A text-based approach refers to matching entities based on their textual description information. Descriptive information of concepts from two graphs can be extracted, and the similarity between two concepts can be measured by calculating the similarity. For example, in different graphs, the concepts “vehicle” and “car”, despite having different names, have the same meaning and description, requiring the establishment of a matching relationship between these two concepts. Figure 7 depicts the graphical representation of classes defined in the Traffic Genome dataset, mapped to their equivalent DSKG classes. The equivalence between elements is defined by the “owl:sameAs” relation, indicating that one element can be replaced by another without changing the meaning, and vice versa.

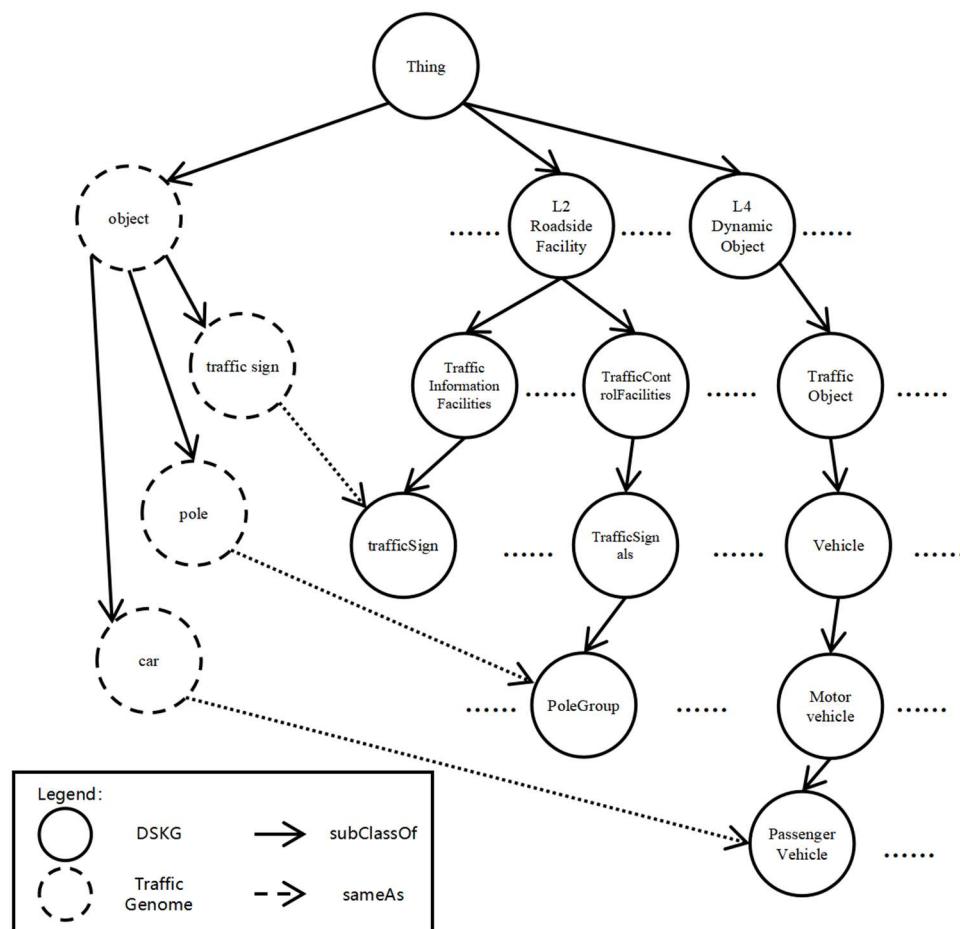


Figure 7. Entity matching diagram.

Through extraction and fusion, we constructed the initial knowledge graph of driving scenarios. The final merged DSKG contains a total of 178 classes, 34 object properties, and 43 data properties. For convenient association, it has been converted and stored in the Neo4j graph database. Neo4j supports large-scale data storage, effectively addressing issues such as low data density, large volumes, and rapid updates in the traffic domain. Additionally, the Cypher graph query language supports relevant queries and graph algorithms, facilitating data querying and value mining. Therefore, Neo4j graph database

was chosen for knowledge storage in this study. Figure 8 presents the final driving scenario domain knowledge graph (partial).

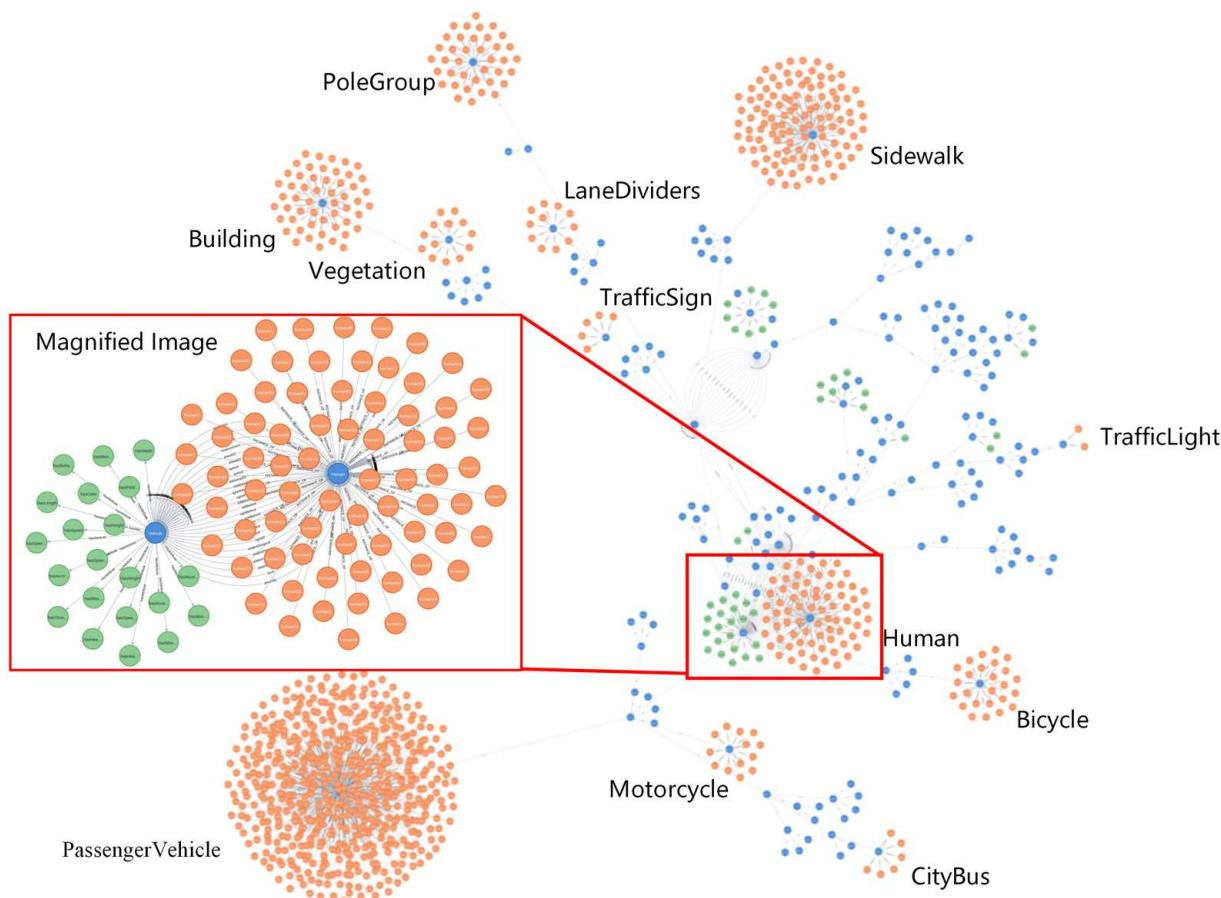


Figure 8. Integrated updated knowledge graph of the driving scenario domain (partial). Blue circles represent classes, orange circles represent instances, and green circles denote properties.

3.4. Knowledge Inference Models Based on Representation Learning

The main idea of knowledge inference based on representation learning is to learn the semantic correlations between entities and relations by mapping them into a low-dimensional continuous vector space. Specifically, knowledge embedding models first project the knowledge graph into a low-dimensional vector space, transforming entities and relations in the graph into low-dimensional vectors. Then, a scoring function is designed to compute the scores of all knowledge triples, and the backpropagation algorithm is used to maximize the scores of the triples actually existing in the knowledge graph, thereby learning the vector embeddings of entities and relations in the knowledge graph.

3.4.1. Model Definition

Knowledge graph representation learning models are a class of methods used to learn vector representations of entities and relations in knowledge graphs. These models play a crucial role in entity-linking prediction tasks in knowledge graphs. This section introduces several commonly used knowledge graph representation learning models, including the TransE [39], Complex [40], Distmult [41], and RotatE [42] models.

The TransE model is a distance-based knowledge representation learning model. Its core idea is to learn the embedding representation of knowledge by translating the relation

vector as the translation operation from the head entity vector to the tail entity vector. For a given triple (h, r, t) , its scoring function is defined as follows:

$$f_r(h, t) = \|h + r - t\|_2 \quad (1)$$

where p is the norm (usually 1 or 2), and $\|\cdot\|_p$ is the norm. This function calculates the distance between the head entity after relation translation and the tail entity, where a smaller distance indicates a better match.

The Complex model is a complex-based knowledge representation learning model. Its core idea is to represent entities and relations in the knowledge graph as complex vectors and use complex inner product to compute the similarity between entities and relations. Its scoring function is defined as follows:

$$f_r(h, t) = \operatorname{Re}(h^T r \bar{t}) \quad (2)$$

where $\operatorname{Re}(\cdot)$ represents the real part, and \bar{t} represents the conjugate of the tail entity vector.

The Distmult model is a dot-product-based knowledge representation learning model. Its core idea is to represent entities and relations in the knowledge graph as vectors and use dot product to measure the similarity between entities and relations. Its scoring function is defined as follows:

$$f_r(h, t) = h^T r \odot t \quad (3)$$

where \odot represents the element-wise product.

The RotatE model is a rotation-based knowledge representation learning model. Its core idea is to use rotation operations to capture semantic correlations between entities and relations. Its scoring function is defined as follows:

$$f_r(h, t) = \operatorname{dist}(h \circ r, t) \quad (4)$$

where \circ represents element-wise multiplication of complex numbers, and $\operatorname{dist}(\cdot)$ represents a metric function, usually 1 or 2-norm.

3.4.2. Model Training and Analysis

Multiple evaluation metrics were employed in this study to assess the performance of different models in link prediction tasks, including Mean Rank (MR), Mean Reciprocal Rank (MRR), Hits@1, Hits@3, and Hits@10. MR represents the average rank of correct triples in the test set. Hits@10 indicates the ratio of correct triples among the top 10 rankings to the total number of triples in the test set. When the average rank is lower, the MRR, Hits@1, Hits@3, and Hits@10 are higher, indicating better inference performance. The optimal hyperparameters for different models are shown in Table 5.

Table 5. Hyperparameter values of the different models.

Model	Learning Rate	Hidden Layer	Margin	Batch Size
TransE	0.001	50/256/512	0.9	20,000
Complex	0.001	50/256/512	None	20,000
Distmult	0.001	50/256/512	0.9	20,000
Rotate	0.001	50/256/512	1.0	20,000

For a deeper understanding of the model training process, we provide Figure 9, which details the training process. Additionally, Table 6 showcases the entity prediction results in the Traffic Genome dataset. These results offer insights into the accuracy, precision, recall, and other metrics of each model in capturing entities within driving scenarios.

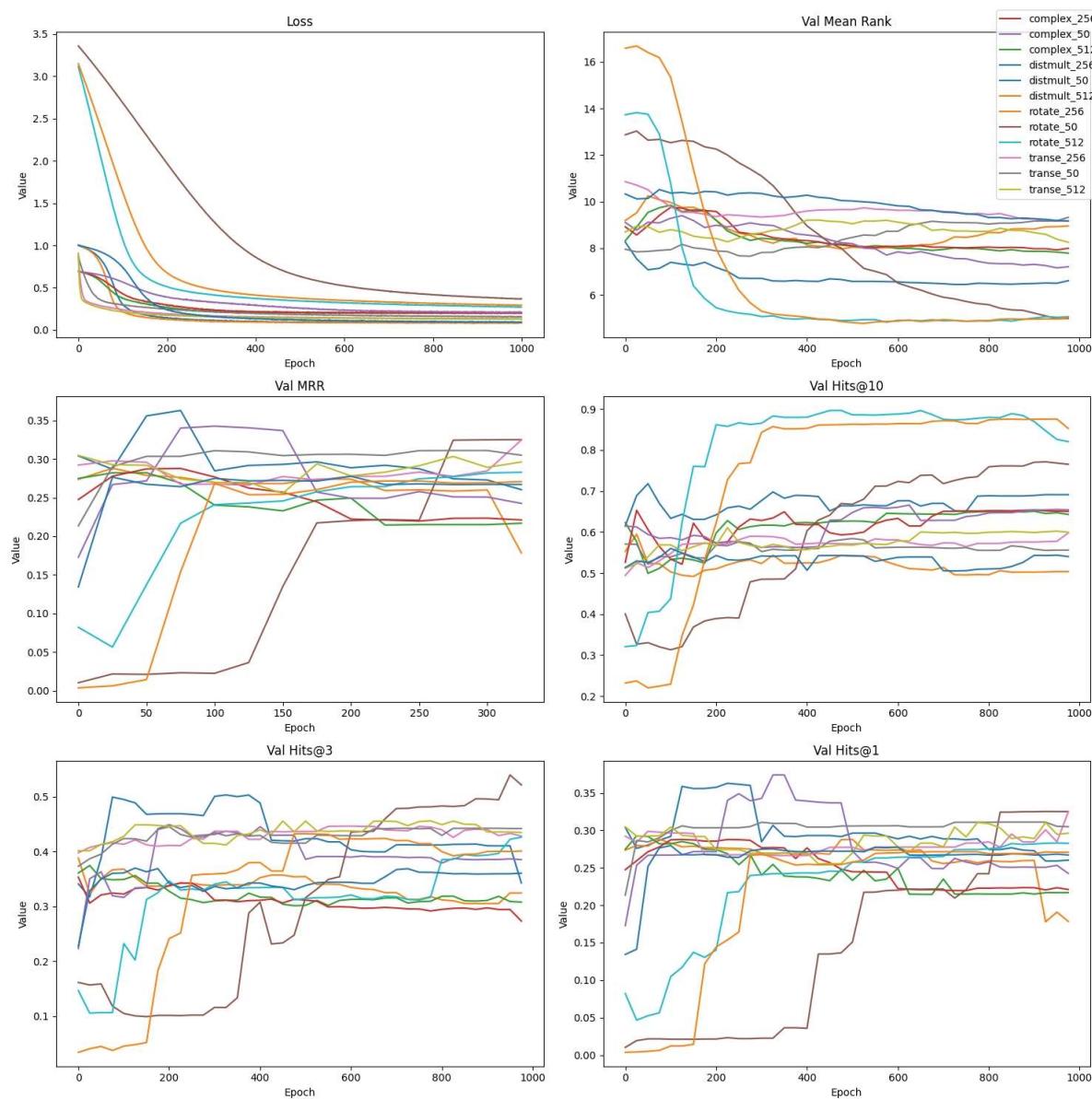


Figure 9. Details of the model training process.

Table 6. Results of predicting the entities.

Model (Optimal Hidden Layer)	MR	MRR	Hits@10	Hits@3	Hits@1
Transe (256)	9.18	41.67%	59.79%	42.82%	32.47%
Complex (50)	8.51	42.01%	56.43%	42.98%	33.77%
Distmult (256)	7.01	46.09%	65.91%	46.92%	36.29%
Rotate (50)	4.99	45.68%	76.54%	52.17%	32.51%

Based on Figure 8 and Table 6, we conducted a comprehensive analysis of four different models. The Rotate model performs best in MR, with an average rank of 4.99, significantly lower than other models. The Complex model excels in MRR, reaching 42.01%, slightly higher than the Rotate model's 45.68%. However, under stricter evaluation metrics such as Hits@10, Hits@3, and Hits@1, the Rotate model demonstrates outstanding performance, ranking first in Hits@10, Hits@3, and Hits@1 with 76.54%, 52.17%, and 32.51%, respectively. The Distmult model also performs well, ranking second in Hits@10 with 65.91%, and second in Hits@3 and Hits@1 with 46.92% and 36.29%, respectively. In contrast, the Transe and Complex models show relatively average performance across all metrics. Therefore,

considering the overall performance, the Rotate model emerges as a preferred choice among these evaluation metrics and could serve as one of the primary models in research.

4. Conclusions

This study investigates the modeling methods of knowledge graphs in the domain of driving scenarios. It includes ontology construction, knowledge extraction and fusion, and analysis of knowledge reasoning models based on representation learning. Ontology construction standardizes domain knowledge. It ensures the wide applicability of concept categories. This process lays the foundation for subsequent knowledge extraction and fusion. In the data extraction and fusion phase, entity data were extracted from the Traffic Genome dataset. It was then integrated with the ontology layer. This facilitated the expansion and updating of the knowledge graph. Finally, we will publicly release the generated DSKG and related materials for use by both industry and academia.

Furthermore, we conducted in-depth research on knowledge reasoning models based on representation learning, introducing several commonly used models and evaluating their performance in link prediction tasks. The analysis revealed that the Distmult model performed the best in metrics such as average ranking and hit rate. This finding validates the effectiveness of the Distmult model in knowledge embedding and supports its application in real-time scene understanding and driving decision-making. Additionally, it can also be applied in scenario-based testing and verification. Knowledge embedding models accurately detect scene similarity, reducing non-safety-critical scenarios and improving testing efficiency.

In the discussion, we acknowledged the scalability and applicability of our method. However, we recognize several limitations. Firstly, our method still requires human intervention in the knowledge graph construction process, lacking full automation. Secondly, the performance of our method is subject to the quality and quantity of data, which may limit its adaptability to modeling complex scenarios. Therefore, in future work, we need to explore more automated and intelligent approaches to enhance the efficiency and scope of our method. Additionally, our method has not fully considered the correlations and transitions between different scenarios, suggesting future research into knowledge associations and transfer methods between different scenes.

In conclusion, our proposed method provides a systematic framework for constructing and enriching a knowledge graph of driving scenarios, advancing research and development in intelligent connected vehicle technology. Future work will focus on addressing the current limitations of the method and exploring more advanced technologies and methods to further advance driving scenario analysis and decision-making.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data for this study are publicly available. Below is the link to access the dataset: dskg: <https://github.com/ZhangCeGitHub/dskg/tree/main> (accessed on 28 April 2024).

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

References

- ISO 21448:2022; Road Vehicles-Safety of the Intended Functionality. ISO: Geneva, Switzerland, 2022. Available online: <https://www.iso.org/obp/ui#iso:std:iso:21448:ed-1:v1:en> (accessed on 28 March 2024).
- National Transportation Safety Board. Collision between a Sport Utility Vehicle Operating with Partial Driving Automation and a Crash Attenuator: Mountain View, California, 23 March 2018. Available online: <https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR2001.pdf> (accessed on 11 April 2024).
- National Transportation Safety Board. Collision between Car Operating with Partial Driving Automation and Truck-Tractor Semitrailer. Available online: <https://www.ntsb.gov/investigations/AccidentReports/Reports/HAB2001.pdf> (accessed on 11 April 2024).
- National Transportation Safety Board. Collision between Vehicle Controlled by Devel Opmental Automated Driving System and Pedestrian. Available online: <https://www.ntsb.gov/investigations/accidentreports/reports/har1903.pdf> (accessed on 11 April 2024).
- Wickramarachchi, R.; Henson, C.; Sheth, A. CLUE-AD: A context-based method for labeling unobserved entities in autonomous driving data. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 16491–16493. [CrossRef]
- Wickramarachchi, R.; Henson, C.; Sheth, A. An evaluation of knowledge graph embeddings for autonomous driving data: Experience and practice. *arXiv* **2020**, arXiv:2003.00344.
- Wickramarachchi, R.; Henson, C.; Sheth, A. Knowledge-based entity prediction for improved machine perception in autonomous systems. *IEEE Intell. Syst.* **2022**, *37*, 42–49. [CrossRef]
- Singhal, A. Introducing the Knowledge Graph: Things, Not Strings. Available online: <https://blog.google/products/search/introducing-knowledge-graph-things-not/> (accessed on 28 March 2024).
- Cai, S.; Ma, Q.; Hou, Y.; Zeng, G. Knowledge Graph Multi-Hop Question Answering Based on Dependent Syntactic Semantic Augmented Graph Networks. *Electronics* **2024**, *13*, 1436. [CrossRef]
- Li, D.; Lu, Y.; Wu, J.; Zhou, W.; Zeng, G. Causal Reinforcement Learning for Knowledge Graph Reasoning. *Appl. Sci.* **2024**, *14*, 2498. [CrossRef]
- Meng, X.; Jing, B.; Wang, S.; Pan, J.; Huang, Y.; Jiao, X. Fault Knowledge Graph Construction and Platform Development for Aircraft PHM. *Sensors* **2024**, *24*, 231. [CrossRef] [PubMed]
- Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
- Paulheim, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semant. Web* **2016**, *8*, 489–508. [CrossRef]
- Geyer, S.; Baltzer, M.; Franz, B.; Hakuli, S.; Kauer, M.; Kienle, M.; Meier, S.; Weißgerber, T.; Bengler, K.; Bruder, R.; et al. Concept and development of a unified ontology for generating test and use—Case catalogues for assisted and automated vehicle guidance. *IET Intell. Transp. Syst.* **2014**, *8*, 183–189. [CrossRef]
- Schuldt, F.; Saust, F.; Lichte, B.; Maurer, M.; Scholz, S. Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen. *Autom. Assist. Eingebettete Syst. Transp.* **2013**, *4*, 1–7.
- Schuldt, F. Ein Beitrag für den Methodischen Test von Automatisierten Fahrfunktionen mit Hilfe von Virtuellen Umgebungen. Ph.D. Thesis, Technische Universität Braunschweig, Braunschweig, Germany, 2017.
- PEGASUS. The PEGASUS Method. Available online: <https://www.pegasusprojekt.de/en/pegasus-method> (accessed on 28 March 2024).
- Bagschik, G.; Menzel, T.; Maurer, M. Ontology based scene creation for the development of automated vehicles. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1813–1820.
- Bock, J.; Krajewski, R.; Eckstein, L.; Klimke, J.; Sauerbier, J.; Zlocki, A. Data basis for scenario-based validation of HAD on highways. In Proceedings of the 27th Aachen Colloquium Automobile and Engine Technology, Aachen, Germany, 8–10 October 2018; pp. 8–10.
- Bagschik, G.; Menzel, T.; Körner, C.; Maurer, M. Wissensbasierte szenariengenerierung für betriebsszenarien auf deutschen autbahnen. *Workshop Fahrerassistenzsyst. Autom. Fahr.* **2018**, *12*, 12.
- Weber, H.; Bock, J.; Klimke, J.; Roesener, C.; Hiller, J.; Krajewski, R.; Zlocki, A.; Eckstein, L. A framework for definition of logical scenarios for safety assurance of automated driving. *Traffic Inj. Prev.* **2019**, *20*, 65–70. [CrossRef] [PubMed]
- Scholtes, M.; Westhofen, L.; Turner, L.R.; Lotto, K.; Schuldes, M.; Weber, H.; Wagener, N.; Neurohr, C.; Bollmann, M.H.; Kortke, F.; et al. 6-layer model for a structured description and categorization of urban traffic and environment. *IEEE Access* **2021**, *9*, 59131–59147. [CrossRef]
- Report on Advanced Technology Research of Functional Safety for Intelligent Connected Vehicles. Available online: <http://www.caicv.org.cn/index.php/material?cid=38> (accessed on 28 March 2024).
- Menzel, T.; Bagschik, G.; Isensee, L.; Schomburg, A.; Maurer, M. From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 2383–2390.
- Tahir, Z.; Alexander, R. Intersection focused situation coverage-based verification and validation framework for autonomous vehicles implemented in Carla. In *International Conference on Modelling and Simulation for Autonomous Systems, Proceedings of the 8th International Conference, MESAS 2021, Virtual, 13–14 October 2021*; Springer: Cham, Switzerland, 2021; pp. 191–212.

26. Herrmann, M.; Witt, C.; Lake, L.; Guneshka, S.; Heinze, C.; Bonarens, F.; Feifel, P.; Funke, S. Using ontologies for dataset engineering in automotive AI applications. In Proceedings of the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 14–23 March 2022; pp. 526–531.
27. ASAM. OpenXOntology. Available online: <https://www.asam.net/standards/asam-openxontology/> (accessed on 28 March 2024).
28. Bogdoli, D.; Guneshka, S.; Zöllner, J.M. One ontology to rule them all: Corner case scenarios for autonomous driving. In *European Conference on Computer Vision, Proceedings of the Computer Vision-ECCV 2022 Workshops, Tel Aviv, Israel, 23–27 October 2022*; Springer: Cham, Switzerland, 2022; pp. 409–425.
29. Westhofen, L.; Neurohr, C.; Butz, M.; Scholtes, M.; Schuldes, M. Using ontologies for the formalization and recognition of criticality for automated driving. *IEEE Open J. Intell. Transp. Syst.* **2022**, *3*, 519–538. [[CrossRef](#)]
30. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
31. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2636–2645.
32. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liang, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. Nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
33. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
34. Geyer, J.; Kassahun, Y.; Mahmudi, M.; Ricou, X.; Durgesh, R.; Chung, A.S.; Hauswald, L.; Pham, V.H.; Mühlegg, M.; Dorn, S.; et al. A2d2: Audi autonomous driving dataset. *arXiv* **2020**, arXiv:2004.06320.
35. Xiao, P.; Shao, Z.; Hao, S.; Zhang, Z.; Chai, X.; Jiao, J.; Li, Z.; Wu, J.; Sun, K.; Jiang, K.; et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 3095–3101.
36. Zhang, Z.; Zhang, C.; Niu, Z.; Wang, L.; Liu, Y. Geneannotator: A semi-automatic annotation tool for visual scene graph. *arXiv* **2021**, arXiv:2109.02226.
37. Jia, M.; Zhang, Y.; Pan, T.; Wu, W.; Su, F. Ontology Modeling of Marine Environmental Disaster Chain for Internet Information Extraction: A Case Study on Typhoon Disaster. *J. Geo-Inf. Sci.* **2020**, *22*, 2289–2303.
38. Ghosh, S.S.; Chatterjee, S.K. A knowledge organization framework for influencing tourism-centered place-making. *J. Doc.* **2022**, *78*, 157–176. [[CrossRef](#)]
39. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 5–8.
40. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; Volume 48, pp. 2071–2080.
41. Yang, B.; Yih, W.; He, X.; Gao, J.; Deng, L. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. *arXiv* **2014**, arXiv:1412.6575.
42. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. *arXiv* **2018**, arXiv:1902.10197.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.