A Knowledge Graph for UAV Mission Planning Systems

Xiaofang Duan(✉), Rong Chai, Siya Zhang, and Chengchao Liang

School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, People's Republic of China
s220132035@stu.cqupt.edu.cn

Abstract. Unmanned aerial vehicles (UAVs) are increasingly applied in various mission scenarios due to their versatility, scalability and cost-effectiveness. In practical UAV application scenarios, UAV mission planning systems (UMPSs) highly rely on efficient mission planning and resource scheduling strategies in order to meet the requirement of UAV missions. However, the dynamic and complex mission environment poses challenges to mission planning in UMPSs. To tackle this problem, knowledge graph technology can be utilized which plays a critical role in managing intricate relationships and constraints among UAVs, missions and environments, ensuring the efficient information update of the mission environment as well as the intelligent perception of the environment knowledge. In this paper, we first summarize the process of constructing knowledge graphs for UMPS, and discuss the concepts and procedures of knowledge modeling, extraction, fusion and representation in detail. Then, we investigate the knowledge reasoning problem in UMPSs which is of particular importance for achieving information update in the mission execution environment, and propose a graph neural network (GNN)-based knowledge reasoning method that leverages the advantages of both graph structure and the path inferring technique to predict the missing entities or relations in the knowledge graph. Finally, the effectiveness and applicability of the proposed knowledge reasoning method are verified via simulations.

Keywords: UAV · Mission planning · Knowledge graph · Knowledge reasoning · GNN

1 Introduction

Benefited from the advantages of high mobility, low cost, strong concealment and easy deployment, unmanned aerial vehicles (UAVs) have been widely used for mission execution in both military and civil fields [1]. The accurate and reliable mission execution of UAVs depends on reasonable and efficient mission planning strategies. In particular, efficient task assignment and resource scheduling strategies, as well as flight trajectory planning for UAVs, should be designed based on perceived environment

This work is supported in part by the National Natural Science Foundation of China (No. 62271097, 62071078).

F. Gao et al. (Eds.): Chinacom 2023, LNICST 590, pp. 97–111, 2024.
https://doi.org/10.1007/978-3-031-67162-3_7

information, mission requirements and the characteristics of UAV mission planning systems (UMPS).

Due to the complex and dynamic environment and mission characteristics of UAVs, the missions should be planned and executed in a flexible and adaptive manner. Traditional mission planning approaches, however, are based on predefined strategy and manual task flow design, which cannot be modified and updated in real time. Knowledge graph technology can be applied to tackle this problem. A knowledge graph is a structured way of representing facts, using entities, relationships and semantic descriptions. Applying knowledge graph technology in UMPS, the entities existing in UMPS as well as their relationships can be depicted in a clear and reasonable manner. Furthermore, the dynamically occurred entities and their relationships can be inferred by using knowledge reasoning technology. As a consequence, the dynamic characteristics of UPMS can be captured and the corresponding mission planning strategy can be designed, which are highly desired.

Some recent research work has studied knowledge representation and reasoning problem in UMPS. Reference [6] creates an ontology for UAV flight control and a geo-ontology for geographic information. Reference [7] proposes a modeling framework for context-aware unmanned systems that use semantic web technology and a fuzzy cognitive map model for decision-making. In [8], an adversarial learning domain adaptation method is proposed to deal with inconsistent data received from multiple sensing sources in remote sensing, and the metric of min-max entropy is introduced to align data distributions. Reference [9] presents an effective method to construct a military domain-specific knowledge graph from textual data. Reference [10] proposes a UAV-based visual tracking framework which integrates tracking methods with semantic modeling, so as to improve object labeling and enhance scene understanding. Although there have been studies on designing knowledge representation or knowledge processing methods for specific task scenarios, the knowledge graph construction and knowledge reasoning approaches have not been extensively studied for autonomous UMPS.

In this paper, we investigate the problem of knowledge graph construction and knowledge reasoning for UMPS. The knowledge related to UMPS is captured, and the corresponding knowledge model is created. Then, the knowledge extraction approaches for different types of data are discussed, the knowledge fusion and representation methods are summarized and the knowledge graph is construed for UMPS. To conduct knowledge reasoning of the constructed knowledge graph, we propose a graph neural network(GNN)-based method that leverages both the graph structure and the path information to infer the missing entities or relations in the knowledge graph.

2 The Procedures of Building Knowledge Graphs

A knowledge graph is a graphical representation of knowledge that establishes a structured model of entities, concepts, and their relationships. The building of a knowledge graph can be achieved through automated methods that integrate knowledge from various sources into a visualized graph, utilizing technologies such as information extraction, natural language processing and machine learning. Knowledge graphs can be classified into general knowledge and domain-specific graphs based on the scope of entities and relationships they involve. Specifically, general knowledge graphs cover multiple domains and include various types of entities and their relationships, while domain-specific knowledge graphs are built for a particular domain and contain entities and relationships that are relevant and applicable to the specific domain.

Building a knowledge graph mainly includes five stages, i.e., knowledge modeling, knowledge extraction, knowledge fusion, knowledge representation, and knowledge reasoning. Knowledge modeling is an important base for building a knowledge graph for UMPS. By modeling the relevant knowledge in a structured and standardized way, a foundation for knowledge extraction, integration and reasoning in UMPS can be created. Knowledge extraction refers to automatically extracting structured knowledge or information from unstructured or semi-structured text or data. Through applying techniques such as entity recognition and relationship extraction, raw data can be transformed into computer-readable formats. Knowledge fusion involves integrating knowledge from various sources into a unified knowledge base through entity disambiguation and semantic integration. Consequently, unified and efficient knowledge representation can be achieved and possible repetitive descriptions and confusing representations can be avoided.

Knowledge representation is to describe knowledge using a certain formal language and data structure. In knowledge graphs, the triplet is a basic knowledge representation format that is used to represent structured knowledge. A triplet consists of three elements, i.e., (subject, predicate, object) or (head, relation, tail), and can be used to represent facts or semantic relationships. As an example, in Fig. 1, we show a knowledge graph about the movie "Inception". In the graph, nodes represent the relevant entities such as "Inception", "Christopher Nolan" and "Science faction", and directed edges represent the relationships between entities, such as "DirectedBy" and "BelongTo". As can be seen from the figure, multiple triplets are defined to describe the knowledge related to the movie, for instance, the triplet (Inception, BelongTo, Science fiction) represents the fact that "Inception" belongs to science fiction.

Knowledge reasoning refers to the process of deducing new knowledge based on known knowledge and rules. By using logical reasoning and related methods, knowledge reasoning is capable of discovering new knowledge hidden behind existing knowledge and updating knowledge graphs in time.

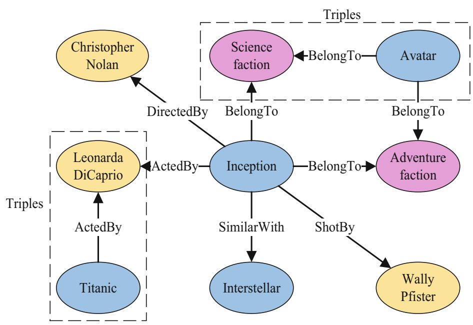

Fig. 1. An example of a knowledge graph.

3 The Knowledge Modeling of UAV Mission Planing System

To build a knowledge graph for UMPS, we first model the knowledge, which involves identifying and defining entities, relationships and attributes. The entities are mainly the physical objects such as UAVs, environment information and obstacles, as well as abstract concepts such as the requirement, constraints, and objectives of mission execution, etc. The features and characteristics of the entities are described by their attributes, e.g., the attributes of the UAVs can be described by maximum flight speed and turning radius, etc., and the missions can be described by types and execution time, etc. The relationships between the entities and that between the entities and their attributes should also be specified. For instance, to execute certain missions, UAVs may consume some resources. The relationships between the UAVs and the resources can simply be defined as “Consuming”.

The diversity and complexity of mission data and environmental information pose challenges to the knowledge modeling for UAV systems. To tackle this problem, we classify the relevant knowledge into different categories based on the domains of the entities and attributes. In particular, the knowledge involved in UMPS is categorized into scenario knowledge, mission knowledge, resource knowledge and performance knowledge, etc.

Scenario knowledge refers to the knowledge depicting the characteristics and status of different task scenarios and application domains in the process of UMPS. In general, scenario knowledge consists of environment characteristics of task scenarios, and the constraints and requirements for task execution. In light of the different types and objectives of tasks involved in the UAV mission execution process, the knowledge of UMPS can be categorized into task perception knowledge, task scheduling knowledge and task execution knowledge. By understanding, mastering, and utilizing mission-related knowledge, UAVs perform tasks more efficiently and accurately.

Resource knowledge pertains to the knowledge about various resources required for task execution, as well as the relationships and constraints among these resources. The typical resource knowledge can be payload resources, communication resources and compute resources, etc. An accurate and comprehensive description of resource knowledge is mandatory for designing efficient mission strategies. Performance knowledge is composed of relevant knowledge depicting the performance of equipment and missions in UMPS. For example, to depict the features and performance of UAVs, knowledge such as maximum flight speed can be modeled. Similarly, to depict the features and performance of task-aware, knowledge such as the detection distance and detection range can be modeled.

The connection between different categories of knowledge in UMPS can be created based on their relationship. For instance, a specific scenario poses certain requirements for resources. Similarly, sufficient resources can guarantee the performance of the system, on the contrary, ensuring the performance of the system requires a certain amount of resources. The knowledge system of UMPS is shown in Fig. 2.

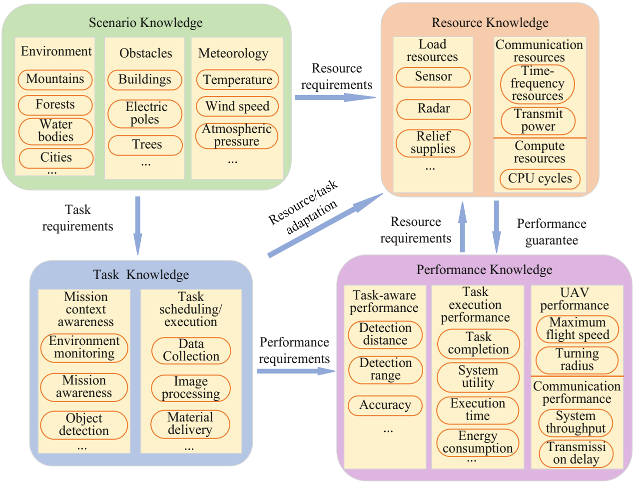

Fig. 2. The knowledge system for UMPS.

4 The Knowledge Exaction of Mission Planing System

Based on the obtained well-defined knowledge during the modeling process, we may conduct knowledge exaction, which distills useful knowledge from a large amount of data. The process of knowledge exaction mainly consists of entity recognition, relation extraction, events extraction and concept recognition. The entity recognition in UMPS aims to extract predefined specific entities, such as mission objectives, flight areas, obstacles, etc., from the collected system/mission data and identify their types accurately. A typical relation extraction approach is to determine the relationship between entities from particular text descriptions, e.g., occurrences of UAV and radar in a sentence may be linked through relations such as equipped with. Similarly, the relationship between UAV and the target can be "monitored" in general environment monitoring scenarios. Automatically identifying the relations between entities is essential for achieving efficient and autonomous task execution in UMPS.

To extract knowledge from the data of different types in UMPS, we may utilize various methods. In particular, to distill knowledge from structured data, such as environmental monitoring data, UAV resource status, task status information, etc., we may employ a database storage method. For instance, database to resource description framework (D2R) mapping technology can be utilized which maps the structured data into a resource description framework. Regarding the semi-structured data in UMPS, e.g., the mission scheduling strategies and the log files generated during the task execution process, specialized wrappers can be designed for targeted knowledge extraction. In Fig. 3, we show different knowledge extraction methods for processing various types of data in UMPS.

The characteristics of the unstructured data such as images and videos in UMPS include large data volume, numerous analysis parameters, different conceptual data models, flexible and diverse knowledge expressions, etc., which pose challenges and

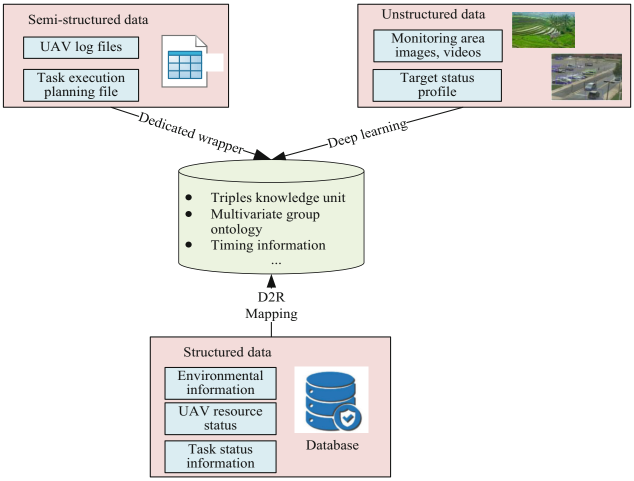

Fig. 3. Knowledge extraction methods for UMPS.

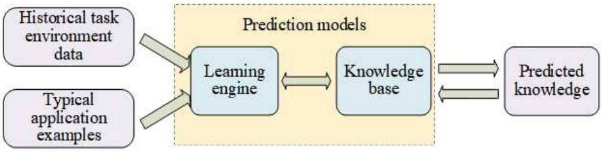

Fig. 4. Schematic diagram of knowledge extraction method based on the deep learning model.

difficulties for conducting efficient and accurate knowledge extraction. To tackle this problem, deep learning methods can be applied which are capable of solving classification and regression problems and performing accurate analysis and recognition of unstructured data in UMPS.

By leveraging the strong correlation of data in time, space and other dimensions, a learning engine can be constructed wherein the received task environment information and typical application instances are applied as inputs, and a knowledge base is established through training on a large amount of data. A prediction model can then be created which can generate the final predicted knowledge given actual input unstructured data in UMPS. Finally, the actual data is input into the prediction model to obtain the final prediction result. Figure 4 shows a schematic diagram of the method for extracting unstructured task knowledge based on deep learning models.

5 Knowledge Fusion, Representation and Storage

Due to the wide range of knowledge sources in UMPS, there may exist knowledge duplication and semantics ambiguity among multi-source heterogeneous knowledge, resulting in difficulties in building large-scale, high-performance knowledge graphs.

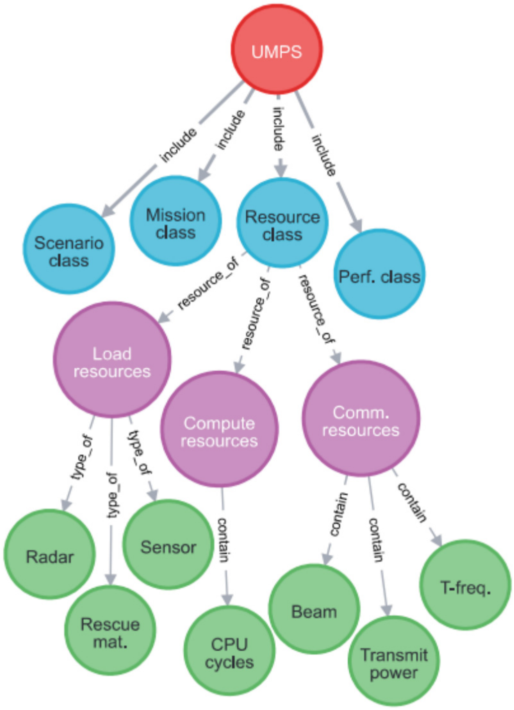

Fig. 5. Part of the knowledge graph of UMPS.

To resolve this problem, knowledge fusion operations such as conflict detection, entity disambiguation, and entity alignment can be conducted. Specifically, machine learning methods and the statistical characteristics of the knowledge can be jointly utilized to perform entity discrimination and analysis. To achieve knowledge entity alignment, classification regression decision tree algorithms and vector quantization methods can be applied. For example, "flight control" and "aircraft control" may refer to the same entity, and ambiguity can be eliminated through knowledge alignment. To fuse the knowledge collected from various fields, a multi-source knowledge fusion algorithm can be applied, which uses a semantic element reconstruction scheme and abstracts the semantic connotations of multi-domain data.

Once various types of knowledge have been extracted and integrated into UMPS, knowledge representation is an essential phase for subsequent reasoning and application. Generally, knowledge is represented in the form of triplets. For example, suppose UAV 1 is required to perform the environmental monitoring task, the relationship between the UAV and the mission can be represented by a triplet (UAV 1, to perform, environment monitoring). Likewise, to depict the fact that target A is sensed by sensor 1, we use a triplet (Target A, sensed by, sensor 1). Knowledge fusion and representation enable complex UMPS knowledge to be represented and stored in a concise and standardized format, facilitating subsequent knowledge reasoning and application.

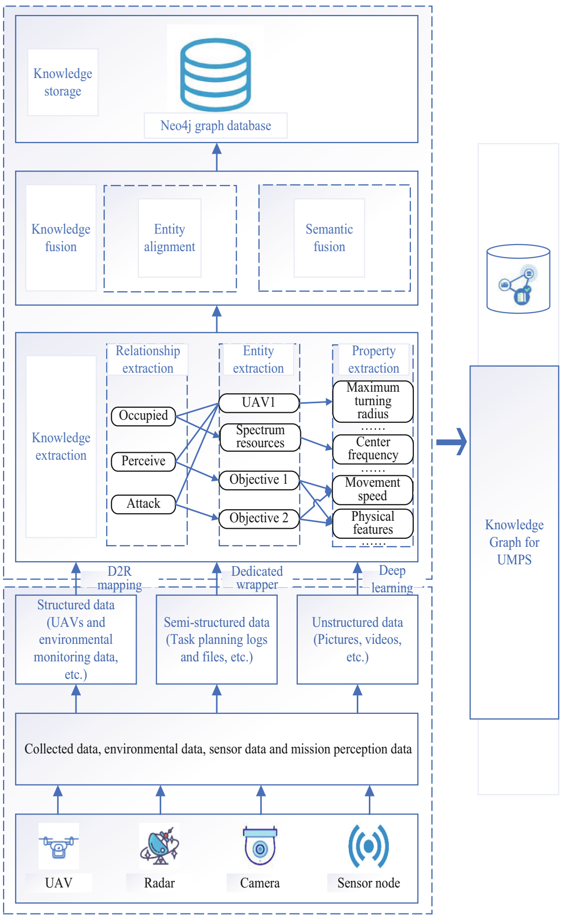

Fig. 6. The Knowledge graph building technology for UMPS.

The knowledge obtained can be stored in the database for easy query and visualization, among which the common databases used to store knowledge graphs are graph databases, relational databases, etc. In this paper, we used the Neo4j graph database to store the knowledge of UMPS, as shown in Fig. 5. Neo4j is a graph database that uses nodes and edges to represent data and Cypher query language for efficient data retrieval. This database provides a highly efficient query performance, especially when queries involve multiple nodes and relationships. In Neo4j, Cypher language is used to complete complex instructions for knowledge processing, such as calling the “CREAT” statement to create nodes or relationships between nodes and calling the “SET” statement to add new attributes to nodes. By using Neo4j, efficient and rapid knowledge addition, deletion, modification and checking can be achieved for UMPS. In Fig. 6, we present the technical framework for the construction of a knowledge graph for UMPS.

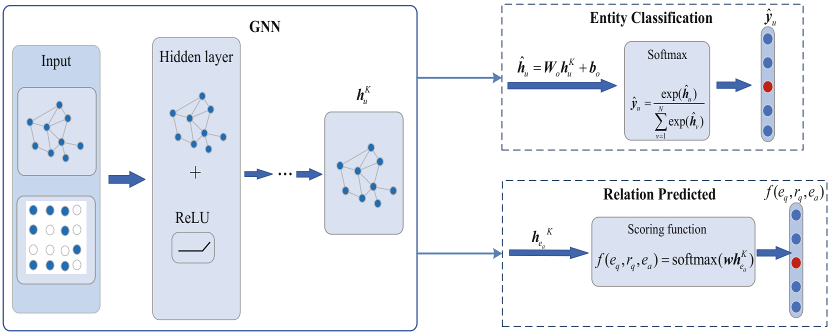

Fig. 7. GNN-based knowledge reasoning.

6 Proposed GNN-Based Knowledge Reasoning Method

In this section, we propose a graph neural network (GNN)-based knowledge reasoning method to predict the missing entities or relations in knowledge graph.

6.1 Problem Statement

Based on the procedures discussed in previous sections, an initial knowledge graph can be constructed. Let $G = (V, R, F)$ denote the obtained knowledge graph, where $V = \{e_u\}$ denotes the set of entities, e_u represents the entity, $R = \{r\}$ denotes the relation set, r represents the relation between two entities, $F = \{(e_u, r, e_v)|e_u, e_v \in V, r \in R\}$ is the set of triple and (e_u, r, e_v) is the triple consisting of two entities and their relation. In practical applications, the dynamic changes in the environment may cause the variation of entities and relations in the knowledge graph, resulting in incompleted or missing triples, e.g., (e_u, r, x), (e_u, y, e_v), where x and y are mission entity and relation, respectively. By utilizing knowledge reasoning techniques, the missing entities and relations of the triples can be predicted.

6.2 An Overview of GNN

GNNs are a type of artificial neural networks designed to handle data represented in the form of graphs. A GNN model consists of a number of layers, and each layer is composed of a number of nodes and edges connecting the adjacent nodes. Both the nodes and edges are characterized by their features. Specifically, node features capture attributes such as node labels, node degrees, and node positions, while edge features may include edge weights, edge types, and edge orientation, etc.

The fundamental principle underlying GNNs is the utilization of pairwise message passing. This mechanism enables an iterative process where graph nodes continually refine their representations by exchanging information with their neighboring nodes. Through this iterative exchange of messages, nodes gather and integrate knowledge from their local context, allowing for improved representations and enhanced reasoning capabilities within the graph.

6.3 GNN-Based Entity Classification

Since the entities in knowledge graphs are in general categorized into different classes, the entity prediction can be considered equivalent to entity classification, that is, determining the class of missing entities. When employing GNN for entity classification, the entities and relations in knowledge graph are respectively set as the nodes and edges of the GNN. In the GNN, the features of the nodes and edges are updated by applying the message passing mechanism and the output of the GNN is the updated entity and relation features.

Specifically, let $\boldsymbol{h}_u^k$ denote the features for entity e_u at the k-th GNN layer, $1 \leq k \leq K$, where the K is the number of layers in GNN. The update formula of $\boldsymbol{h}_u^k$ is given by:

$$\boldsymbol{h}_u^{k+1} = \boldsymbol{h}_u^k + \sum_{r \in R} \sigma \left(\sum_{v \in N_r(u)} \boldsymbol{W}_r^k \boldsymbol{h}_v^k + \boldsymbol{B}_r^k \right), \tag{1}$$

where $N_r(u)$ is the set of neighbors of e_u that are connected by relation r, $\boldsymbol{W}_r^k$ and $\boldsymbol{B}_r^k$ respectively represent the weight matrix and bias vector of the relation r at the k-th layer of GNN, σ is the activation function, modeled as the ReLU function.

The node features from the final layer are passed through a fully connected layer. Let $\hat{\boldsymbol{h}}_u$ denoted as the final output feature vector of node e_u, which can be computed as follows:

$$\hat{\boldsymbol{h}}_u = \boldsymbol{W}_o \boldsymbol{h}_u^K + \boldsymbol{b}_o, \tag{2}$$

where $\boldsymbol{h}_u^K$ is the final output feature vector of node e_u, $\boldsymbol{W}_o$ and $\boldsymbol{b}_o$ represent the weight matrix and bias vector of the output layer, respectively.

To conduct entity classification, we evaluate the probability distribution of various entity classes by using softmax function. Let $\hat{\boldsymbol{y}}_u = [\hat{y}_{uj}]$ denote the output classification result, where $\hat{y}_{uj}$ represents the probability that entity e_u belongs to class j, we obtain

$$\hat{\boldsymbol{y}}_u = \frac{\exp(\hat{\boldsymbol{h}}_u)}{\sum_{v=1}^{N} \exp(\hat{\boldsymbol{h}}_v)}, \tag{3}$$

where $\exp(x)$ is a natural exponential function, N indicates the number of nodes.

In order to improve the accuracy of entity classification, we use the cross entropy loss function to evaluate the error between the predicted result and the true label. Let $L(\boldsymbol{W}, \boldsymbol{B})$ denote the cross entropy loss function, where $\boldsymbol{W} = \{\boldsymbol{W}_r^k, \boldsymbol{W}_o\}$ and $\boldsymbol{B} = \{\boldsymbol{B}_r^k, \boldsymbol{b}_o\}$ represent the parameters of the GNN model, we define loss function as follows:

$$L(\boldsymbol{W}, \boldsymbol{B}) = -\frac{1}{N} \sum_{u=1}^{N} \sum_{j=1}^{C} y_{uj} \log(\hat{y}_{uj}), \tag{4}$$

where C is the number of classes, y_{uj} is the distribution of real labels, representing the probability that sample e_u belongs to class j. The model parameters $\boldsymbol{W}$ and $\boldsymbol{B}$ can be obtained by computing the derivative of the loss function and using the gradient descent method.

6.4 GNN-Based Relation Prediction

Since the entities in the knowledge graph are connected by two or more relations, accurately reasoning the relations between entities can not only enhance the information richness of the knowledge graph, but also improve the quality and accuracy of the knowledge. Moreover, by deriving new relations between entities, a more complete knowledge system can be established.

In order to capture the sequential connection of the entities, path-based reasoning method can be applied whereby one path generally refers to a series of connections that pass through from one entity to another. A path with length N is characterized by a set of N triples (e_q, r_1, e_2), (e_2, r_2, e_3), ..., (e_N, r_N, e_a), that are connected head-to-tail from e_q to e_a sequentially.

In this section, we define $G_{e_q,e_a|N}$ as a subgraph from e_q to e_a, containing all paths from e_q to e_a, where e_q is the query entity and e_a is the target entity. Applying GNN model for relation reasoning, we embed the relation r in the features of entities e_o, i.e., $\boldsymbol{h}_{e_o}$. The message passing function is specified as:

$$\boldsymbol{h}_{e_o}^{k+1}(e_q, r_q) = \boldsymbol{h}_{e_o}^{k} + \sigma\left(\sum_{(e_s,r,e_o)\in L_{e_q}^{n}} \boldsymbol{W}^k(\boldsymbol{h}_{e_s}^{k}(e_q, r_q) + \boldsymbol{h}_{r}^{k+1})\right), \tag{5}$$

where $L_{e_q}^{n}$ represents the path from e_q to e_a, r_q is query relation and $\boldsymbol{h}_r^{k+1}$ is an embedding of the relation r.

After aggregating information across K layers based on (5), the obtained $\boldsymbol{h}_{e_a}^{K}(e_q, r_q)$ encapsulate the crucial information of (e_q, r_q, e_a). Consequently, we define a scoring function of (6) as:

$$f(e_q, r_q, e_a) = \mathrm{softmax}(\boldsymbol{w}\boldsymbol{h}_{e_a}^{K}), \tag{6}$$

where w is a weight matrix. The loss function of relation prediction is defined as follows:

$$\begin{aligned} L(\theta) = & \sum_{(e_q,r_q,e_a)\in G} y \log f(e_q, r_q, e_a) \\ & + \sum_{(e_q,r_q,e_a)\notin G} (1-y)\log(1 - f(e_q, r_q, e_a)), \end{aligned} \tag{7}$$

where y is an indicator, and we set $y = 1$ for training triples with relation and $y = 0$ for training triples with no relation. We randomly initialize the model parameters $\theta = \{\boldsymbol{W}^k, \boldsymbol{w}\}$ and optimize them by applying stochastic gradient descent to minimize the loss faction. The GNN-based knowledge reasoning method is illustrated in Fig. 7.

7 Simulations

In this section, we consider an example of UMPS where a UAV is employed for package delivery and the flight trajectory of the UAV is determined. The knowledge graph of the considered UMPS is constructed and knowledge reasoning is conducted based on the proposed GNN-based algorithm.

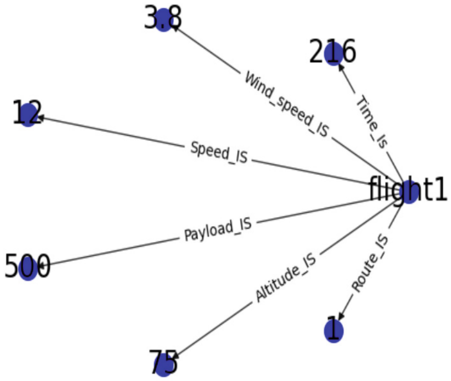

Fig. 8. Examples of triples.

7.1 Pre-process of Data Set

An online dataset is utilized which describes various flight parameters of the UAV such as wind speed, speed, payload and altitude, etc. In particular, a flight refers to the data recorded from the take-off to landing of the UAV along a predefined route. We treat each flight of the UAV as an entity and consider attributes such as flight time, flight speed, payload, altitude, wind speed, and flight route, etc. Based on the data set, we first define a number of triplets, such as (Flight1, TimeIs, 216), (Flight1, WindSpeedIs, 3.8), (Flight1, SpeedIs, 12), (Flight1, PayloadIs, 500), and (Flight1, AltitudeIs, 75), etc. In Fig. 8, we plot several triples of Flight1.

Based on the created knowledge graph of the UMPS, we apply the proposed GNN-based knowledge reasoning algorithm to predict missing entities in the triples. First, we split the data set into an 80% training set and a 20% test set. Secondly, the attribute values of the flights of the UAV are set as the features of the input nodes and the connection relationship between the entity and the attribute is used as the edge features. For example, wind speed, load, speed and height are modeled as the attributes of nodes (flight), and the connection relationship between the nodes and their attributes are set as the characteristics of edges.

7.2 Knowledge Reasoning Results of the Considered UMPS

To conduct entity reasoning, we create a GNN model consisting of an input layer, an output layer, and 16 hidden layers. The number of iterations for the GNN model is set to 2000. We divide the dataset into three subsets, which are denoted as Dataset 1, 2, and 3, respectively.

In Table 1, we show the numbers of entities and edges in each dataset and the simulation results, including the loss function, accuracy and the macro-averaged F1 score. It can be observed that the proposed GNN-based entity reasoning method achieves low cross-entropy losses and high accuracies on various datasets, including a smaller training dataset, which indicates that the constructed GNN model is capable of capturing the patterns and features in the training data effectively, demonstrating its robustness and generalization ability. Even with larger datasets, the model maintains its validity and accuracy.

Table 1. Entities reasoning. Best performance is indicated by the bold face numbers.

Datasets	The number of entities	The number of edges	Loss Function	Accuracy	F1
Dataset 1	3500	14166	0.112	0.995	0.748
Dataset 2	7000	28526	0.207	0.988	0.744
Dataset 3	14000	56964	0.372	0.993	0.747

Regarding the relation reasoning experiment, we utilize a portion of the dataset containing 12209 entities and 6 relationships. We employ the widely used ranking metrics, i.e., mean reciprocal rank (MRR), Hit@1 and Hit@10 [14], to evaluate the performance of the proposed GNN-based relation prediction method. The higher values indicate better performance. Specifically, MRR is the average of the reciprocal ranks of the correct entities over all queries. Hit@1 and Hit@10 are the percentages of queries where the correct entity is ranked within the top 1 and top 10 positions, respectively. Apparently, these metrics are able to reflect both the accuracy and the diversity of the model predictions.

Table 2. Relation reasoning. Best performance is indicated by the bold face numbers.

	MRR(%)	Hit@1(%)	Hit@10(%)
TransE [12]	44.5	37.7	53.8
ComplEX [13]	50.6	51.6	56.7
SimpIE [14]	62.5	71.5	79.7
Proposed GNN-based method	90.3	81.8	93.6

In Table 2, we show the simulation results of the relation prediction method. For comparison, we also evaluate the performance of the reference relation reasoning methods, i.e., TransE model proposed in [12], ComplEX model proposed in [13] and SimpIE model proposed in [14]. By comparing our proposed GNN-based method with other baseline models, we can observe that our GNN model outperforms other models on most of the datasets and metrics, especially on MRR, Hit@1 and Hit@10, which are more sensitive to the ranking quality. This indicates that our GNN-based method can assign a higher rank to the correct relations compared to other models, thereby offering more reliable and relevant predictions. The reason that our proposed approach offers better performance than other baseline models is because the proposed GNN-based method can leverage both the graph structure and the path information when inferring the missing entities, thus leading to a more effective and efficient result.

8 Conclusions

In this paper, we have delved into the challenges of knowledge graph construction and knowledge reasoning within the domain of UMPS. The knowledge related to UMPS has been discussed, which lays a foundation for the construction of domain knowledge graph. Then, we have thoroughly explored various knowledge extraction approaches tailored to different data types. The key methods for knowledge fusion and representation are summarized, culminating in the successful construction of a comprehensive knowledge graph for UMPS. In the pursuit of effective knowledge reasoning within the constructed knowledge graph, we have proposed a GNN-based approach which harnesses both the inherent graph structure and essential path information, empowering accurate inference of missing entities and relations in the knowledge graph. The simulation results have shown that our GNN-based knowledge reasoning method can accurately predict the missing entities and relationships.

References

1. Hassija, V., et al.: Fast, reliable, and secure drone sommunication: a comprehensive survey. IEEE Commun. Surv. Tut. **19**(4), 2802–2832 (2021)
2. Gu, J., Su, T., Wang, Q., Du, X., Guizani, M.: Multiple moving targets surveillance based on a cooperative network for multi-UAV. IEEE Commun. Mag. **56**(4), 82–89 (2018)
3. Xiong, F., Li, A., Wang, H., Tang, L.: An SDN-MQTT based communication system for battlefield UAV swarms. IEEE Commun. Mag. **57**(8), 41–47 (2019)
4. Shakhatreh, H., et al.: Unmanned aerial vehicles (UAVs): a survey on civil applications and key research challenges. IEEE Access **7**, 48572–48634 (2019)
5. Fotouhi, A., et al.: Survey on UAV cellular communications: practical aspects, standardization advancements, regulation, and security challenges. EURASIP J. Wirel. Commun. IEEE Commun. Surv. Tut. **21**(4), 3417–3442 (2019)
6. Hu, X., Liu, J.: Ontology construction and evaluation of UAV FCMS software requirement elicitation considering geographic environment factors. IEEE Access **8**, 106165–106182 (2020)
7. Cavaliere, D., Senatore, S., Loia, V.: Proactive UAVs for cognitive contextual awareness. IEEE Syst. J. **13**(3), 3568–3579 (2019)
8. Rahhal, M.M.A., Bazi, Y., Al-Hwiti, H., Alhichri, H., Alajlan, N.: Adversarial learning for knowledge adaptation from multiple remote sensing sources. IEEE Geosci. Remote Sens. Lett. **18**(8), 1451–1455 (2021)
9. Cavaliere, D., Loia, V., Saggese, A., Senatore, S., Vento, M.: Semantically enhanced UAVs to increase the aerial scene understanding. IEEE Trans. Syst. Man Cybern. Syst. **49**(3), 555–567 (2019)
10. Wu, Y., et al.: Unmanned aerial vehicle knowledge graph construction with SpERT. In: Qin, B., Wang, H., Liu, M., Zhang, J. (eds.) CCKS 2021. CCIS, vol. 1553, pp. 151–159. Springer, Singapore (2022). https://doi.org/10.1007/978-981-19-0713-5_17
11. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 [cs.CL] (2015)
12. Bordes, A., Usunier, N., Garcia-Duran, A.: Translating embeddings for modeling multi-relational data. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 2787–2795 (2013)

13. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33nd International Conference on Machine Learning, pp. 2071–2080 (2016)
14. Kazemi, S.M., Poole, D.: SimplE embedding for link prediction in knowledge graphs. In: Proceedings of the NeurIPS, pp. 4284–4295 (2018)