Sustainable spatial and semantic-web enhanced pathfinding in dynamic domains (SPEED): A case study of grain transportation in Ukraine

by

Yinglun Zhang

B.S., Kansas State University, 2019

A THESIS

submitted in partial fulfillment of the
requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
Carl. R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2023

Approved by:

Major Professor
Dr.Hande Küçük McGinty

Copyright

Abstract

Optimal routing of goods is crucial when addressing supply chain challenges. Within this context, grain transportation stands out as a significant sub-issue. For this research,grain transportation is defined as the process of determining the best routes between grain elevators and railway stations. As in most supply chain and path finding problems, this process is also challenging due to the complexity and dynamics of the domain. In this work, we propose a novel application that uses the KNowledge Acquisition and Representation Methodology (KNARM) and the KnowWhereGraph to enhance the path-finding process during transportation of goods, in this specific case grain. KNARM is a methodology that allows for creating and maintaining modular ontologies that can represent complex domains and support reasoning. KnowWhereGraph, is a densely connected, cross-domain knowledge graph and geo-enrichment service stack that provides rich and up-to-date geospatial information. By integrating these two components, our application aims to leverage the semantic and spatial knowledge to find more accurate and efficient paths for grain transportation. To find the optimal path, we use the A* algorithm, which is a heuristic search algorithm that can find the optimal path between two locations, taking into account the criteria of cost, time, and risk. The A* algorithm can also adapt to dynamic and uncertain situations, such as changes in weather, traffic, or security conditions by updating the paths. We also discuss the challenges and limitations that we faced during the ontology development and data integration process, and how we resolved or mitigated them. Our work demonstrates the potential of using ontologies and knowledge graphs to enhance path-finding problems in complex and dynamic domains. We show that our application can find accurate, efficient, and robust paths for grain transportation, based on the feedback from domain experts and GIS experts. We also provide new insights into the modeling of the factors that may affect grain transportation in Ukraine, such as weather, traffic, and security conditions.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Grain is one of the most important agricultural products in Ukraine, accounting for about 15% of its GDP and 40% of its exports. However, the ongoing military invasion with Russia has disrupted the transportation infrastructure and threatened the security and sustainability of grain trade. In this work, we aim to answer the following research question: How can we use semantic and spatial knowledge to enhance the path-finding process for grain transportation in Ukraine under a military invasion situation accurately and dynamically?

Finding optimal paths for grain transportation is a challenging problem that requires sophisticated algorithms and data. The paths should minimize the distance travelled, maximize the grain storage capacity, and reduce the risk of disruption. Existing solutions often rely on static maps and historical data, overlooking the dynamic and uncertain nature of real-world scenarios. On the other hand, knowledge graphs are capable of integrating a wide variety of data sources in a semantically rigorous way, according to well-established W3C standards[1–3]. Due to their ability to bridge human conceptualization and machine understanding[4] and a powerful standard for recording provenance[5], results drawn from them are also inherently explainable, and interpretable.

Our research focuses on addressing this gap by leveraging semantic and spatial knowledge to provide dynamic solutions tailored to the unique challenges of grain transportation. While some similarities exist with other problems involving dynamic and uncertain situations, such as military logistics, our work distinguishes itself through its domain specificity, criteria, and data sources. In doing so, we not only offer valuable insights and methodologies for solving these transportation challenges but also tackle specific nuances that set our problem apart.

Furthermore, our research has broader implications. It extends to various applications demanding dynamic solutions for transportation optimization. For instance, finding optimal paths in transportation networks with random link travel times[6], hierarchical optimization of optimal path finding for transportation applications[7], and global optimal path to ensure a certain chance of reaching the destination on time in a network with randomness[8] all require sophisticated algorithms and data analysis. These applications share the common thread of dynamic and uncertain conditions, including factors like traffic congestion, accidents, and weather conditions, impacting path cost, time, and risk. Consequently, our research contributes valuable insights and methods that can be applied to these related problems, leveraging semantic and spatial knowledge to enhance decision-making. We hypothesize that using knowledge graphs and semantic representations of path-finding related data, we will improve performance of finding relevant paths by integrating the heuristics faster using the data entered in knowledge graphs.

In this work, we present a novel application that uses the KNowledge Acquisition and Representation Methodology (KNARM)[9] and the KnowWhereGraph[10] to enhance the path-finding process. KNARM is a methodology that allows for creating and maintaining modular ontologies that can represent complex domains and support reasoning. KnowWhereGraph is a densely connected, cross-domain knowledge graph and geo-enrichment service stack that provides rich and up-to-date geospatial information. By integrating these two components, our application can leverage the semantic and spatial knowledge to find more accurate and efficient paths for grain transportation, especially when we need to account for many different heuristics. To achieve this, we use the A* algorithm[11], which is a heuristic search algorithm that can find the optimal path between two locations, taking into account the criteria of distance, grain storage, and risk. The A* algorithm can also adapt to the rapdly changing and uncertain situations, such as changes in weather, traffic, or security conditions, by updating the path based on the latest information from the KnowWhereGraph.

The rest of this thesis is organized as follows: In chapter 2, we present a literature review on the related work in the fields of path-finding, ontology engineering, and geospatial knowledge graphs. We discuss the main challenges and opportunities in these fields, and identify the gaps and limitations that our work aims to address. In chapter 3, we describe the design and implementation of our application, including the ontology development, the data integration, and the path-finding algorithm. In chapter 4, we present the results of our application, and elaborating on the main findings of this research and implications of our research for theoretical basis of computer science. And in chapter 5, we discuss some limitations and future directions for improvement. We also present our evaluation plan for our application using real-world data of grain transportation in Ukraine, and suggest some possible extensions or applications of our work to other domains or problems that can benefit from semantic and spatial knowledge.

Chapter 2

LITERATURE REVIEW

In this chapter, we review the related work in the fields of knowledge graphs and optimization algorithms, which are the main components of our path-finding application. Knowledge graphs are structured representations of knowledge that can support semantic interoperability, reasoning, and querying. Optimization algorithms are methods that can find optimal or near-optimal solutions for complex problems that involve multiple objectives, constraints, and uncertainties.

Both fields have been widely studied and applied in various domains, such as artificial intelligence, information retrieval, natural language processing. However, there are still many challenges and opportunities for improving the performance and accuracy of knowledge graphs and optimization algorithms, especially in dynamic and uncertain environments such as war zones. Therefore, we aim to address the following research questions in this literature review:

- How are knowledge graphs constructed, maintained, and utilized in various domains, and what are the current techniques and trends?
- What are the specific challenges and limitations of applying knowledge graphs and optimization algorithms to path-finding problems in war situations?
- What are the potential opportunities and directions for enhancing knowledge graphs

and optimization algorithms for path-finding problems in war situations?

To answer these research questions, we conducted a systematic and comprehensive search for relevant and reliable sources that cover the topics of knowledge graphs generation methodology and optimization algorithms. We used online databases, search engines, bibliographies, and recommendations from domain experts to find sources that are recent, authoritative, and peer-reviewed. We synthesized and summarized the main findings and contributions of the sources, and related them to our own research problem and contribution. We organized the literature review into two main sections: one for knowledge graphs and one for optimization algorithms. In each section, we further divided the sources into subtopics based on their focus or approach.

2.1 Knowledge Graph Generation Methodology

Knowledge graphs are structured representations of knowledge that can support semantic interoperability, reasoning, and querying. They consist of entities, relations, and attributes that capture the concepts and facts of a domain of interest.[2] Knowledge graphs can be constructed from various sources, such as databases, text, images, etc., using different methods, such as manual annotation, rule-based extraction, machine learning, etc. In this section, we review some of the existing generation methodology and applications of knowledge graphs in various domains and provide insights into how knowledge graphs are constructed, maintained, and utilized across different domains, highlighting current techniques and trends.

A great source for learning more about ontologies and ontology methods is through a comprehensive review paper by Hogan et al.[12] among other review papers and books[13]. There exists many methods and approaches: DILIGENT[14] was an early example of an agile methodology that supported distributed, loosely-controlled, and evolving engineering of ontologies. It involved a fine-grained methodological approach based on Rhetorical Structure Theory, which enabled domain experts to participate in ontology engineering by argumentation; Another example is Modular Ontology Modeling (MOM)[13;15], which aimed to reduce the complexity and increase the reusability of ontologies by applying principles of modularity and abstraction. MOM involved a top-down approach that started from a high-level conceptual model and refined it into a modular ontology using patterns and rules.

2.1.1 KNARM Methodology

One of the current techniques for constructing knowledge graphs is the KNowledge Acquisition and Representation Methodology (KNARM), which was proposed by Küçük McGinty et al.[9] to systematize the process of creating comprehensive, consistent, and useful ontologies for big data integration and analysis. Ontologies are formal specifications of the concepts and relations in a domain, which can serve as the schema or backbone of a knowledge graph. KNARM consists of nine steps: sub-language analysis, unstructured interview, sub-language recycling, meta data creation and knowledge modeling, structured interview, KA validation, database creation, semi-automated ontology building, and ontology validation. In each step, the authors provide guidelines and best practices for involving domain experts and knowledge engineers, as well as tools and resources for facilitating the ontology development. The authors applied KNARM to overview several ontologies for biomedical domains, such as the Drug Target Ontology (DTO)[16], which is a semantic model for drug discovery that covers various aspects of drug targets, such as their classification, function, interaction, etc., the LINCS[17] Information FramEwork (LIFE) ontology[18], which is a semantic model for the Library of Integrated Network-based Cellular Signatures (LINCS) project that covers various types of cellular response signatures for different perturbations, such as drugs, genes, and diseases, and the BioAssay Ontology (BAO) 2.0[19], which is an extension and refinement of the original BAO[20] that describes chemical biology screening assays and their results, including high-throughput screening (HTS) data.

KNARM is an example of how knowledge graphs can be constructed from domain knowledge using a systematic and collaborative approach. It also shows how knowledge graphs can be maintained by updating and validating the ontologies with new data and feedback from domain experts. Moreover, it demonstrates how knowledge graphs can be utilized for various purposes, such as data integration, analysis, visualization, and discovery. KNARM is a technique that can be applied to other domains as well, such as war situations, where there is a need for integrating and analyzing heterogeneous and dynamic data sources.

2.1.2 Existing ontologies

In this section, we explore relevant ontologies that play a crucial role in addressing path-finding challenges within military contexts. Ontologies are formal representations of concepts and relations in a domain of interest that can support semantic interoperability, reasoning, and querying on knowledge graphs. Some examples of such ontologies are: GeoNames[21], DBpedia[22], OpenStreetMap[23], and KnowWhereGraph[10]. These ontologies serve as formal representations of concepts and relationships within a specific domain, facilitating semantic interoperability, reasoning, and querying within knowledge graphs. We looked into these existing ontologies to borrow their vocabularies that may be suitable for our application, as well as to identify the gaps and limitations that need to be addressed by our proposed ontology. In the following text, we will elaborate more on each of these ontologies and explain how they can be used for path-finding problems in war situations.

GeoNames[1] is a comprehensive geographical database that covers all countries and contains over 11 million placenames, such as cities, mountains, rivers, etc. It provides various types of information for each place, such as coordinates, population, elevation, timezone, etc, where users are able to access and query the database through web services and APIs, such as geocoding, reverse geocoding, nearby places, etc. These services and APIs can help users to find places by name, address, or coordinates, and to retrieve relevant information and data for their applications. GeoNames can be useful for path-finding problems in war situations because it can provide basic geographic information and location names for any

[1] https://www.geonames.org/

place in the world. For instance, if a user wants to find a route from kviv to lviv in Ukraine, GeoNames can provide the coordinates, population, elevation, and timezone of both cities, as well as the names and distances of other places along the way. However, GeoNames has some limitations that can affect the accuracy and efficiency of path-finding solutions in war situations. For it may not have the latest information about road closures, checkpoints, or enemy movements that may affect the safety and feasibility of a route. Hence, GeoNames may need to be complemented by other sources of geographic information that can provide more details and updates for path-finding problems in war situations.

DBpedia[2] is a popular and widely used knowledge base that extracts structured information from Wikipedia, such as infoboxes, categories, links, abstracts, etc. It covers various domains and topics, such as people, places, events, organizations, etc., and provides rich semantic annotations and links for each entity. These annotations and links can help users to explore and discover the relationships and connections between different entities, and to access more information and data from other sources. Users can access and query the knowledge base through web services and APIs, such as SPARQL endpoint, lookup service, keyword search, etc. These services and APIs can help users to find entities by name, type, or keyword, and to retrieve relevant information and data for their applications. DBpedia may be useful for path-finding problems in war situations because it might provide background knowledge and context for any entity or topic related to the war scenario, such as historical events, political actors, military operations, etc. DBpedia may also help users to understand the causes and consequences of the war situation, and to identify potential allies and enemies along the way. However, DBpedia has some limitations that may affect the accuracy and timeliness of path-finding solutions in war situations. Similarly to GeoNames, DBpedia may not have the latest information about the current status of the war situation, the changes in the political and military landscape, or the impact of the war on the environment and infrastructure due to its nature. Therefore, it may also need to be complemented by other

[2] https://www.dbpedia.org/

sources of general knowledge that can provide more up-to-date and comprehensive data for path-finding problems in war situations.

OpenStreetMap[3] is a collaborative project that creates a free editable map of the world. It provides detailed maps and road networks for any place in the world, as well as using tags and attributes that describe the properties and characteristics of these features and objects, such as names, types, addresses, etc. Users can access and query the map data through web services and APIs, and such APIs can help users to view and edit the map, to find locations and directions, and to use the map data for their applications. This might be a useful feature in our application for path-finding problems in war situations because it may provide information about obstacles and hazards that may affect the safety and feasibility of a route, such as road blockages, checkpoints, or enemy positions (if possible). However, OpenStreetMap has some major limitations in out applications that might affect the quality and timeliness of path-finding solutions in war situations,like mentioned above. For example, OpenStreetMap may not have the most accurate and up-to-date data depending on the contributors, the lack of semantic annotations and links that may provide more information and context for the map features and objects, and the difficulty of handling dynamic data that may change due to war situations.

KnowWhereGraph[4] is a geospatial knowledge graph that integrates multiple sources of geographic information with semantic annotations and links. It combines and aligns the data from different sources using semantic matching and linking techniques, and enriches the data with additional information and context from other sources, such as traffic conditions, weather, events, etc. Users can access and query the knowledge graph through a SPARQL endpoint and a web interface, such as finding places by name, type, or keyword, retrieving relevant information and data for each place, exploring the relationships and connections between different places, etc. KnowWhereGraph may be useful for path-finding problems in war situations because it provides an integration of multiple sources of geographic infor-

[3]https://www.openstreetmap.org/
[4]https://www.knowwheregraph.org/

mation. It may also help users to discover and compare alternative routes that may have different advantages and disadvantages depending on the war situation. In our application, KnowWhereGraph also has some limitations that might compromise the quality and reliability of path-finding solutions in war situations. A possible issue is that KnowWhereGraph may produce errors or conflicts in the data due to the differences and inconsistencies between different sources; another issue is that KnowWhereGraph may face scalability and efficiency issues due to the large size and complexity of the knowledge graph, and have difficulty handling dynamic data that may change due to war situations. Hence, KnowWhereGraph may require additional sources of geographic information that can enhance the accuracy and reliability of path-finding solutions in war situations.

In Table 2.1, we provide a summarized comparison of these ontologies, encompassing aspects such as their domain coverage, size, data format, data sources, accessibility, utility, and inherent limitations to our application.

2.2 Challenges and limitations of knowledge graphs and optimization algorithms for path-finding in war situations

Path-finding is a fundamental problem in many domains, such as robotics, navigation, logistics, etc. It involves finding a sequence of actions or movements that can lead an agent from a start location to a goal location, while satisfying some criteria or constraints, such as distance, time, cost, safety, etc. Path-finding can be formulated as an optimization problem, where the objective is to minimize or maximize some function of the path, such as its length, duration, risk, etc.

Our approach involves knowledge graphs with optimization algorithms. Knowledge graphs may be useful for path-finding problems, as they can provide rich and structured informa-

Table 2.1: Comparison of Existing Ontologies

Ontology	Domain	Size	Format	Source	Access	Usefulness	Limitations
GeoNames	Geography	11 million place-names	RDF	Geographical database	Web services and APIs	Basic geographic information and location names	Lack of detailed maps, road networks, and dynamic data
DBpedia	General knowledge	6.6 million entities and 9.5 billion facts	RDF and OWL	Wikipedia	Web services and APIs	Background knowledge and context for any entity or topic related to the war scenario	Dependence on Wikipedia's quality and coverage, inconsistency and incompleteness of some data, difficulty of handling temporal and spatial aspects
OpenStreetMap	Geography and mapping	2.7 billion nodes and 4.8 million ways	XML and RDF	Crowdsourcing	Web services and APIs	Detailed maps and road networks for any place in the world	Data quality and accuracy may vary depending on contributors, lack of semantic annotations and links
KnowWhereGraph	Geography and semantics	1.2 billion triples and 110 million entities	RDF and OWL	Multiple sources of geographic information	SPARQL endpoint and web interface	Integration of multiple sources of geographic information with semantic annotations and links, dynamic data such as traffic conditions and weather	Data integration and alignment may introduce errors or conflicts, scalability and efficiency issues

tion about the environment, the agent, and the goal. For example, a knowledge graph can represent the spatial layout of a terrain, the properties and capabilities of a vehicle, and the preferences and requirements of a user. Knowledge graphs can also support semantic queries and reasoning, which can help to find relevant and feasible paths, as we will discuss in the following text.

One of the main challenges of applying knowledge graphs and optimization algorithms to path-finding problems in war situations is dealing with dynamicity, uncertainty, and security issues. War situations might change rapidly and frequently, with roads being blocked, enemies moving unexpectedly, and allies changing their plans. These dynamic changes may affect the validity and optimality of the paths that are computed by the optimization algorithms. Therefore, knowledge graphs and optimization algorithms must be able to update their information and solutions in real-time or near real-time to account for these changes. Another challenge is handling incomplete or hidden information, such as obscured terrain, etc. These uncertainties may affect the accuracy and reliability of the paths computed by optimization algorithms. To handle uncertainty, knowledge graphs and optimization algorithms help incorporate probabilistic or fuzzy models and methods. For example, probabilistic models may be used to represent the uncertainty in the location of enemies or allies, while fuzzy models might be used to represent the uncertainty in the terrain or the weather. A third challenge is ensuring security and confidentiality in war situations, as the environment and the agents might be vulnerable or hostile. For example, some information about the terrain or the agents may be classified or secret, some enemies may try to intercept or sabotage the communication or computation of the paths, some allies may have conflicting or competing interests or agendas, etc. These security issues can affect the integrity and availability of the paths that are computed by the optimization algorithms. Therefore, knowledge graphs and optimization algorithms need to be able to handle security by implementing encryption or authentication mechanisms and protocols. These are some of the challenges and limitations that we face when applying knowledge graphs and optimization algorithms to path-finding problems in war situations.

We try to answer the following research question: What are the specific challenges and limitations of applying knowledge graphs and optimization algorithms to path-finding problems in war situations? To do so, we review some existing works that address these challenges and limitations in different ways. We also identify some gaps and opportunities for future research in this area.

2.2.1 Dynamicity

One of the main challenges of applying knowledge graphs and optimization algorithms to path-finding problems in war situations is handling the dynamicity of the environment and the agents. Dynamicity refers to the fact that the environment and the agents can change rapidly and frequently, which can affect the validity and optimality of the paths that are computed by the optimization algorithms. For example, roads can be blocked or damaged by explosions, enemies can move or attack unexpectedly, allies can change their plans or objectives, etc. Therefore, knowledge graphs and optimization algorithms need to be able to update their information and solutions in real-time or near real-time.

Some existing works that address this challenge are:

Koenig et al.[24] proposed an incremental heuristic search algorithm called D* Lite, which is a simplified version of D*, a classic algorithm for dynamic path-finding. D* Lite uses a reverse search strategy that starts from the goal and moves towards the start, updating the costs and values of the states along the way. D* Lite can efficiently handle changes in edge costs by only re-planning from the states that are affected by the changes, rather than re-planning from scratch. The authors showed that D* Lite is faster and simpler than D*, and can find optimal paths in dynamic environments. Wu et al.[25] proposed a dynamic knowledge graph embedding model called DKGE, which can learn low-dimensional vector representations of entities and relations in a knowledge graph that changes over time. DKGE uses a recurrent neural network to capture the temporal dependencies and evolution patterns of the knowledge graph, and a convolutional neural network to capture the spatial correlations and semantic similarities of the knowledge graph. DKGE can update the embeddings of the entities and relations in real-time or near real-time, and can handle insertions, deletions, or updates of triples in the knowledge graph. However, these works also have some limitations and gaps, such as:

Koenig et al.'s algorithm assumes that the changes in edge costs are small and infrequent, which may not be realistic for war situations where large and frequent changes can occur. Moreover, their algorithm does not consider multiple objectives or constraints for path-finding, such as distance, time, risk, etc. Wu et al.'s model assumes that the knowledge graph is complete and consistent, which may not be true for war situations where incomplete or inconsistent information can exist. Moreover, their model does not consider how to use the embeddings for path-finding or optimization tasks.

2.2.2 Uncertainty

Another challenge of applying knowledge graphs and optimization algorithms to path-finding problems in war situations is handling the uncertainty of the environment and the agents. Uncertainty refers to the fact that the environment and the agents can be partially observable or hidden, which can affect the accuracy and reliability of the paths that are computed by the optimization algorithms. For example, some data about the terrain or the agents can be missing or incomplete, some entities or relations can have ambiguous or duplicate names, some connections or transitions can be uncertain or probabilistic, etc. Therefore, knowledge graphs and optimization algorithms need to be able to handle uncertainty by incorporating probabilistic or fuzzy models and methods.

Jaillet et al.[26] applied an existing decision criterion, called the Requirements Violation (RV) Index, to various routing optimization problems under uncertainty. The RV Index quantifies the risk associated with the violation of requirements taking into account both the frequency of violations and their magnitudes whenever they occur. The RV Index can handle instances when probability distributions are known, and ambiguity when distributions are partially characterized through descriptive statistics such as moments. The authors extended the robust framework based on the RV Index to the case where the requirements (time windows) are also part of the decisions, and solved the robust vehicle routing problem with time window assignments (RVRP-TWA). They developed practically efficient algorithms involving Benders decomposition to find the exact optimal routing solution in which the RV Index criterion is minimized.

While reviewing the literature on path-finding problems under uncertainty, we also looked into internet routing problems[27], as one of our professors suggested that they might be relevant to our research. We found that internet routing problems share some similarities with path-finding problems, such as the use of graphs, optimization algorithms, and uncertainty models. However, we also found that they have some differences that make them distinct and challenging in their own ways.

Path-finding is very similar to internet routing, but they have some differences in terms of graphs, objectives, and uncertainty. Path-finding is a more general problem of finding a sequence of actions or movements that have the ability to lead an agent from a start location to certain goal location, while satisfying some criteria or constraints. On the other hand, internet routing is a special case of path-finding where the agent is a data packet, the location is a network node, and the criteria or constraints are related to network performance or security. One difference between path-finding and internet routing is that path-finding involves different types of graphs, such as spatial graphs or planar graphs, while internet routing usually involves aspatial graphs. Spatial graphs represent the physical layout of the environment, such as roads, buildings, or terrain, while aspatial graphs represent the logical connections between nodes, such as routers, switches, or servers.

Another difference is that path-finding considers different types of objectives or constraints, such as distance, time, cost, safety, etc., while internet routing usually considers only one objective or constraint, such as minimizing hop count or delay. Hop count is the number of intermediate nodes that a packet passes through before reaching its destination, while delay is the time it takes for a packet to travel from its source to its destination. A third difference is that path-finding may have different types of uncertainty sources, such as environmental factors or agent behaviors, while internet routing usually has only one type of uncertainty source, such as link failures or congestion. Environmental factors may affect the availability or quality of paths, such as weather conditions, natural disasters, or enemy attacks, while agent behaviors will affect the preferences or requirements of paths, such as user preferences, vehicle capabilities, or mission objectives. Link failures are situations where a link between two nodes becomes unavailable due to hardware malfunction, power outage, or malicious attack, while congestion is a situation where a link becomes overloaded due to high traffic demand. These are some of the differences between path-finding and internet routing that make them distinct problems.

2.2.3 Security

A third challenge of applying knowledge graphs and optimization algorithms to path-finding problems in war situations is handling the security of the information and the solutions. Security refers to the fact that the information and the solutions might be vulnerable or confidential, which might affect the integrity and availability of the paths that are computed by the optimization algorithms. For example, some information about the terrain or the agents can be classified or secret, and some enemies may try to intercept or sabotage the communication or computation of the paths, etc. Hence, knowledge graphs and optimization algorithms need to be able to handle security by implementing encryption or authentication mechanisms and protocols.

Xie and Xing[28] proposed CryptGraph, which runs graph analytics on encrypted graph data to preserve the privacy of both users' graph data and the analytic results. In CryptGraph, users encrypt their graphs before uploading them to the cloud. The cloud runs graph analysis on the encrypted graphs and obtains results which are also in encrypted form that the cloud cannot decipher. During the process of computing, the encrypted graphs are never decrypted on the cloud side. The encrypted results are sent back to users and users perform the decryption to obtain the plaintext results. In this process, users' graphs and the analytics results are both encrypted and the cloud knows neither of them. Thereby, users' privacy can be strongly protected. Meanwhile, with the help of homomorphic encryption, the results analyzed from the encrypted graphs are guaranteed to be correct. The authors presented how to encrypt a graph using homomorphic encryption and how to query the structure of an encrypted graph by computing polynomials. They also proposed hard computation outsourcing to seek help from users for certain operations that are not executable on encrypted graphs. They applied their methods to two graph algorithms: shortest path and connected components, and showed their correctness and feasibility.

2.3 Opportunities and directions for enhancing knowledge graphs and optimization algorithms for path-finding in war situations

This section explores some possible ways to enhance knowledge graphs and optimization algorithms for path-finding problems in war situations. Path-finding is a fundamental problem in many applications, such as navigation, planning, routing, etc. However, path-finding in war situations is a challenging problem that involves dynamicity, uncertainty, security, multiple objectives, constraints, etc. Therefore, our main goal in this section is to identify the potential opportunities and directions for improving knowledge graphs and optimization algorithms for path-finding problems in war situations.

We begin by reviewing some of the current limitations and gaps of the existing approaches that we discussed in the previous section, followed by suggesting some potential solutions and directions that might address or reduce these limitations and gaps, and enhance the performance and accuracy of path-finding in war situations. We also examine some of the expected benefits and challenges of implementing these solutions and directions.

2.3.1 Challenges and Solutions for Path-Finding in War Situations

In this section, we discuss some of the challenges and solutions for path-finding in war situations. Path-finding is a fundamental problem in many applications, such as navigation, planning, routing, etc. However, path-finding in war situations involves several challenges that require enhancing knowledge graphs and optimization algorithms.

One of these challenges is scalability, as most existing approaches are designed for small- or medium-scale knowledge graphs. However, path-finding in war situations requires the use of large-scale and complex knowledge graphs that represent the terrain, enemy forces, and allied forces. These knowledge graphs might be computationally expensive to process and update in real time. A possible solution to this challenge is to use distributed computing techniques to parallelize or distribute the computation or storage of knowledge graphs and optimization algorithms across multiple nodes or devices.

Another challenge is adaptability, as most existing approaches are based on fixed or predefined models or methods, which may not be able to adapt to the changing environment or agents in war situations. For example, some approaches may rely on static or deterministic models or methods that do not account for the dynamicity or uncertainty of the environment or agents. Some approaches may also depend on specific or domain-specific models or methods that do not generalize well to different scenarios or domains. A possible solution to this challenge is to use reinforcement learning techniques to learn or update the models or methods of knowledge graphs and optimization algorithms dynamically based on feedback from the environment or agents.

A final challenge is interoperability, as most existing approaches are focused on either knowledge graphs or optimization algorithms, but not both. This may limit the interoperability and integration of knowledge graphs and optimization algorithms for path-finding in war situations. For example, some approaches may not be able to exploit the full potential of knowledge graphs for enhancing optimization algorithms, or vice versa. A possible solution to this challenge is to use semantic web techniques to standardize or harmonize the representation or communication of knowledge graphs and optimization algorithms.

In conclusion, while knowledge graphs and optimization algorithms have the potential to revolutionize path-finding in war situations, they also face several challenges. By addressing these challenges, we may develop more effective and robust path-finding systems that might help grain transporting navigate the complex and dangerous battlefields of the future.

Chapter 3

APPROACH

In this chapter, we discuss how we develop the ontology component for our application, using the KNARM methodology[9] – as shown in figure 3.1– with the KnowWhereGraph[10], and the path-finding algorithm. We describe the purpose and scope of our ontology, the sources and methods of knowledge acquisition, the structure and content of our ontology, the data integration, and the evaluation and validation of our ontology. We also discuss the challenges and limitations that we faced during the ontology development process, and how we resolved or mitigated them.

3.1 Using KNARM Methodology to Model Location for Grain Transportation

We developed our ontology, which we named SPEED (Sustainable Spatial and Semantic-web Enhanced Pathfinding in Dynamic Domains), using ROBOT[29] and Protégé[30]. We used the Web Ontology Language (OWL)[31], which is a W3C standard[1] language for developing ontologies on the Semantic Web . We followed the KNARM methodology , which consists of nine steps: sub-language analysis, unstructured interview, sub-language recycling, meta data

[1]https://www.w3.org/OWL/

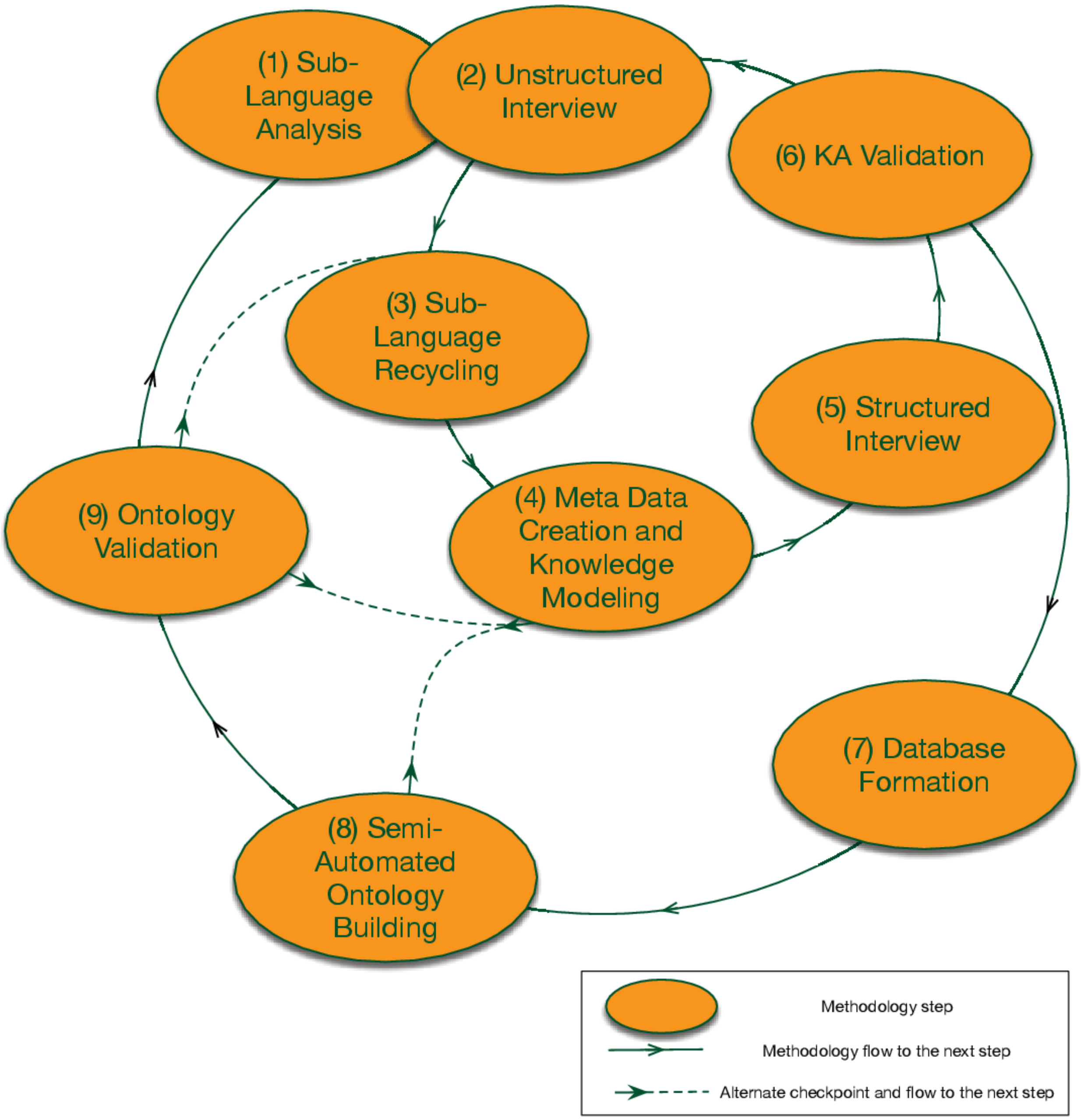


Figure 3.1: The nine-step of KNARM methodology from Mcginty, Hande Küçük. "KNowledge Acquisition and Representation Methodology (KNARM) and Its Applications." (2018).[9]

creation and knowledge modeling, structured interview, KA validation, database creation, semi-automated ontology building, and ontology validation

Table 3.1 shows the main features or benefits of the tools and languages that we used to create our ontology.

Tool/Language	Feature/Benefit
ROBOT	A command-line tool that allows for automating various ontology tasks, such as mapping, reasoning, validation, conversion, etc.
Protégé 2	A graphical user interface tool that allows for creating and editing ontologies using various formats and plugins.
OWL	A standard language for developing ontologies on the Semantic Web that supports logic-based reasoning and inference.
KNARM	A methodology that supports the creation and maintenance of modular ontologies that can integrate different types of knowledge sources and support reasoning.

Table 3.1: Tools and languages used for ontology engineering

Sub-language analysis

The first step of sub-language analysis involved discovering the units of information or knowledge, and the relationships between them within existing knowledge sources related to our domain of grain transportation in Ukraine. We used a corpus of documents and reports from various sources to find the recurring patterns and concepts in the data. We consulted with the domain expert to verify and refine our sub-language analysis.

We concluded that there are certain types of places and things that we wanted to capture in our ontology, such as elevators, railways, train stations, and motorways. Elevators are facilities that store and process grain; railways are tracks that connect different locations; train stations are places where trains stop or depart; and motorway are connections between elevators and train stations. We also decided that automobile roads were too much to handle and to address in the limited time we had, so we focused on railroads as the main mode of transportation. To illustrate the data format that we need to transform for our ontology, we show a map of railroads in Ukraine as in figure3.2. This map shows the network of rail lines and stations in Ukraine, without any semantic annotations or labels, and such data is in a format that's not operable directly by any algorithms, hence we need to transform this data into a knowledge graph that can support semantic interoperability, reasoning, and querying for path-finding problems in war situations. The map also shows the complexity of the railroads in Ukraine, which pose challenges for path-finding problems in war situations. In later chapter 4.1, we will show the resulting map in ArcGIS system that we created using our ontology and optimization algorithm as in figure4.1.

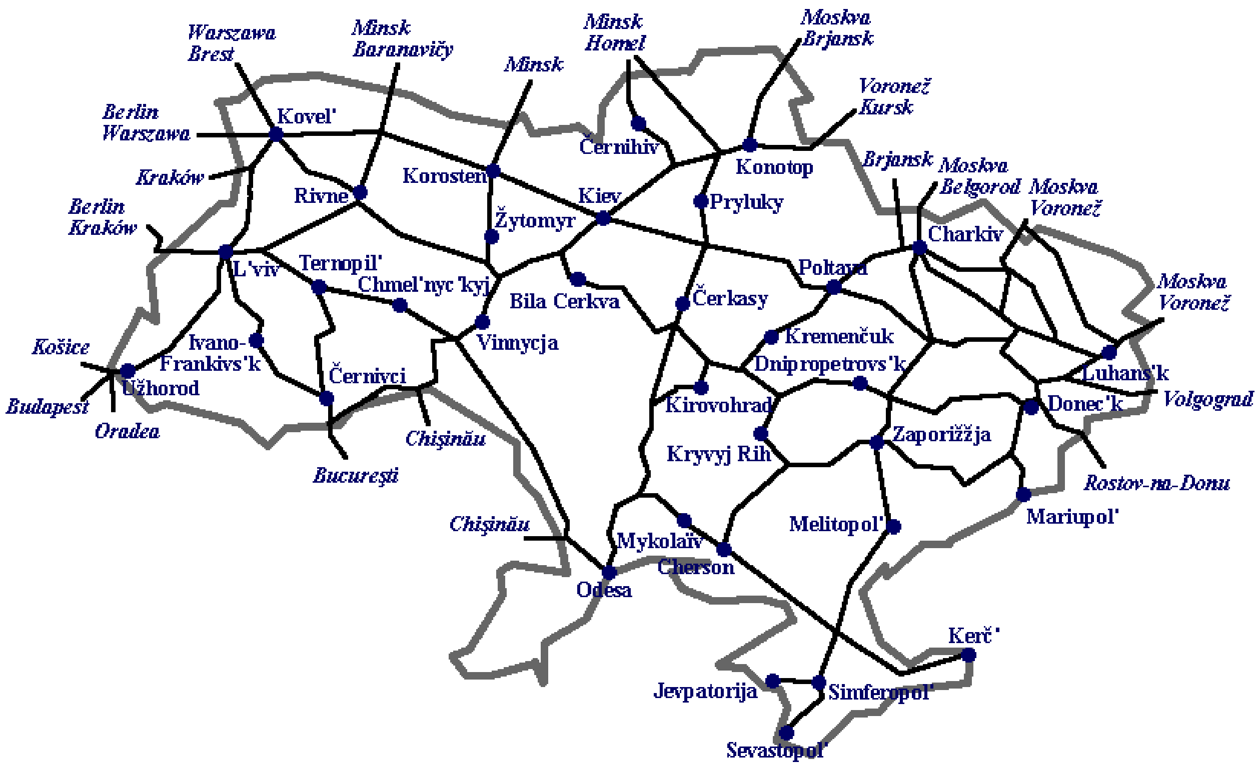


Figure 3.2: A map of Ukraine showing the network of rail lines and stations in the country, without any semantic annotations or labels. This is a version of this data in a format that's not operable directly by any algorithms. We need to transform this data into a knowledge graph that can support semantic interoperability, reasoning, and querying for path-finding problems in war situations[32]

Unstructured interview

The second step of unstructured interview involved conducting a semi-formal interview with the domain expert to elicit his tacit knowledge and experience about the domain. We asked open-ended questions about their expectations from our application. We also asked them to provide examples and scenarios of grain transportation problems and solutions. Due to the sensitives nature of the data, We did not record nor transcribe the interview.

We learned more about the details and characteristics of elevators, railways, train stations, ports, and motorway from the domain expert. For example, we learned that some elevators and train stations are the same place, where the train goes inside the elevator to load or unload grain. We also learned about the preferences and constraints that affect the choice of optimal paths for grain transportation, such as distance, capacity, risk, etc. We used an agile approach to refine our ontology and knowledge graph based on the domain expert's feedback.

We also created a graphical representation of our railway station modeling, based on the information and feedback from the domain expert as in figure 3.3. The figure shows the main components and properties of a railway station, such as its name, location, connections, etc.

Sub-language recycling

The third step of sub-language recycling involved reusing existing standards and ontologies that could provide useful information for our domain. We searched for relevant resources on the Semantic Web and other online repositories, such as GeoNames , DBpedia , OpenStreetMap, and KnowWhereGraph. These resources provide rich and up-to-date geospatial information, such as location names, coordinates, distances, types, relations, etc.

However, we did not use all the information from these resources, as some of them did not make much sense for our application. For example, we did not use the population or elevation data from GeoNames or DBpedia, as they were not relevant for our path-finding problem. We also did not use the road network data from OpenStreetMap or KnowWhereGraph, as we focused on railroads as the main mode of transportation.

We decided to use only the vocabulary for certain concepts that were very limited and essential for our domain. However, we made sure that we could have cross references or mappings to the terms in these different resources, so that we could reuse them in different applications or contexts in the future.

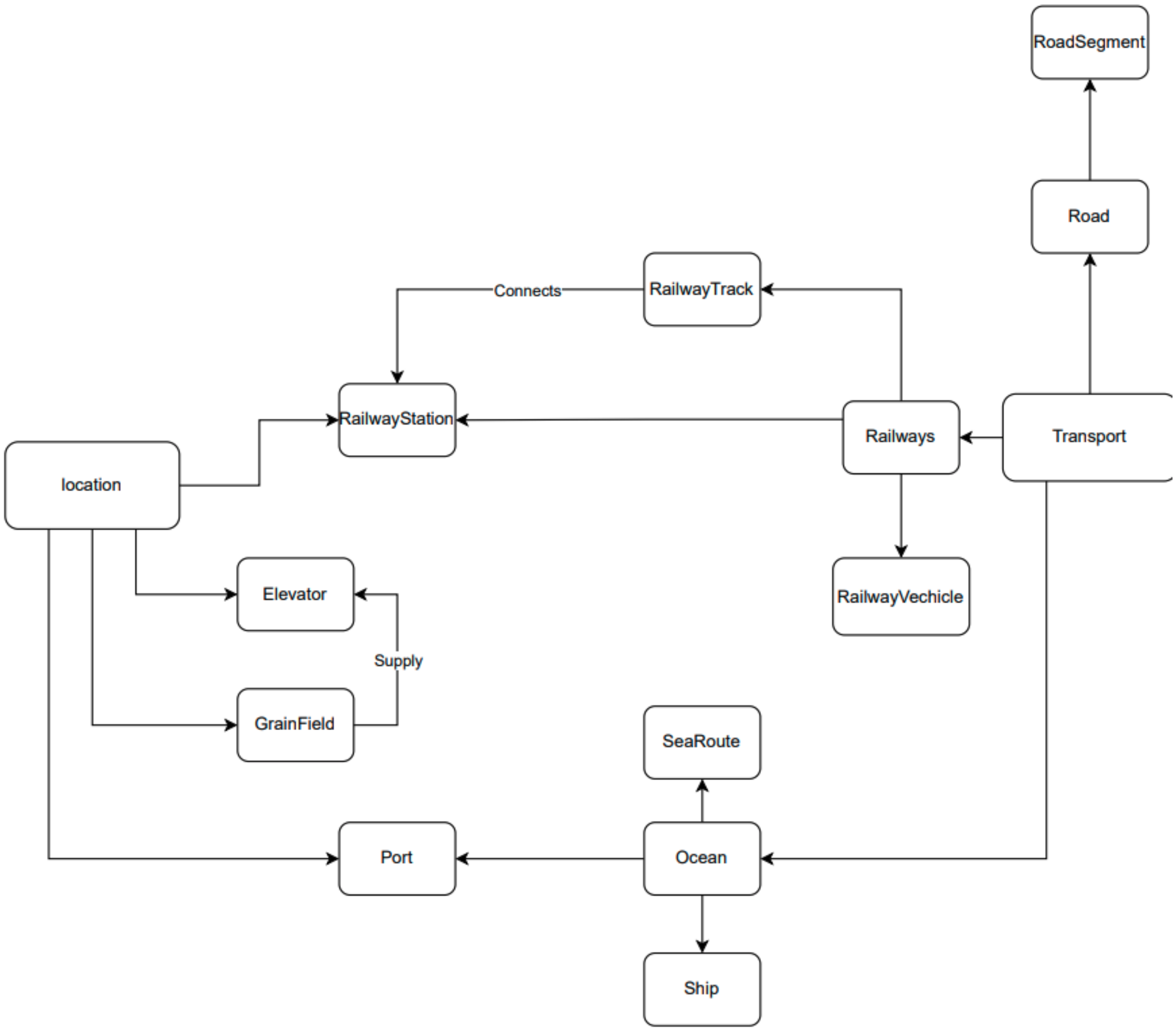
Meta data creation and knowledge modeling

The fourth step of meta data creation and knowledge modeling involved creating a conceptual model of our domain using ROBOT and Protégé. We defined the classes, properties, individuals, axioms, rules, and constraints that represent the essential aspects of grain transportation in Ukraine. We used a modular approach to organize our ontology into different regions based on geographic criteria. We also used a location ontology as a core module that defines the basic concepts and relations for any location in our domain.

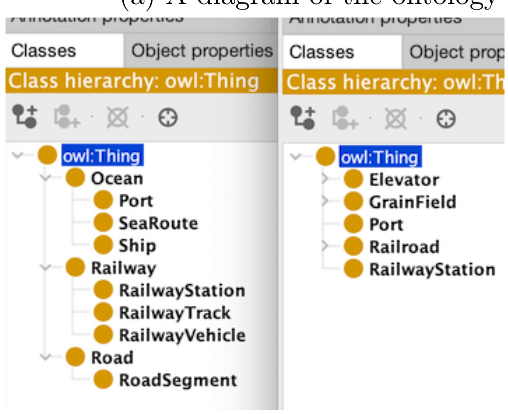
Structured interview and KA validation

The fifth and sixth steps of structured interview and KA validation involved constantly checking the consistency and completeness of our knowledge acquisition process with domain expert and GIS expert to see if what we were doing was correct and accurate. We used various methods to validate our ontology, such as logical reasoning, competency questions, scenario testing, peer review, and feedback sessions. We also revised and refined our ontology based on the results of the validation.

As we were taking an agile approach, we got more clear with the problem, as the domain experts were getting a better understanding with what we were doing. The domain expert in the very beginning did not even realize what we were asking her for, due to background information/knowledge differences. After we put some data in an Excel sheet format, we communicated with the domain expert that some of the names were in Ukrainian, and we asked for her help to clean that up. We also assigned global identifiers to each individual



(a) A diagram of the ontology modeling for grain transportation in Ukraine



(b) A screenshot of the ontology editing in Protégé

Figure 3.3: An example of ontology design and editing for grain transportation in Ukraine, using draw.io and Protégé tools. The figure shows the ontology modeling their entity relation (a), and the ontology editing in Protégé software (b). The ontology consists of several classes, properties and relations that capture the spatial, temporal, economic and risk aspects of the domain.

data entry to avoid name duplications. Then we cleaned up the data sheet using various tools and techniques. We also visualized the route in the ArcGIS system/environment using the data from our ontology and knowledge graph with the help from the GIS expert.

Database creation

The seventh step of database creation involved creating a data structure that could store the instances and their relations based on our model. We decided to use a dictionary data structure in Python to store the data of the railway system, instead of creating a database. This decision was based on the following factors:

- Time constraint: We had a limited time to complete the project, and creating a database would require extra steps and resources.

- Data sensitivity: The data we used was confidential and sensitive, and we did not want to expose it to any potential risks or breaches. By using a dictionary, we could read the data directly from the Excel file, and manipulate it within the program, without having to store it in a database.

- Data dynamics: The data we used was constantly changing and updating, and we wanted to reflect those changes in our model. By using a dictionary, we could easily modify the data and its relations, without having to update the database.

The dictionary we used was a nested dictionary, where each key-value pair represented an instance of the railway system. The key was the name of the instance, and the value was another dictionary that contained its attributes and relations. The attributes included its main transportation method (railway, auto, port), and its location type (railway station or grain elevator). The relations included the nearest railway station and/or grain elevator it connected to, with its distance in kilometers.

Semi-automated ontology building

The eighth step of semi-automated ontology building involved creating an OWL file that contains our model and our data using ROBOT. We converted our dictionary into an OWL file that contained individuals and their relations based on our location ontology. We also added some annotations and metadata to our file to make it more understandable and reusable. The following figure shows the ROBOT template we used to generate the OWL file from the dictionary:

	A	B	C	D	E	F	G	J	K
1	Ontology Id	coordiant	Region / district	Name	parent	Location	One-time storage, metric tonnes	Google map Link	
2	ID	A rdfs:coordinate	A rdfs:region	A rdfs:label	SC % SPLIT=|	AL rdfs:Location@en	A rdfs:Storage	A rdfs:Google Map Link	
3									
4	ex:10000000			Lviv Railway					
5	ex:10000809			Southern Railway					
6	ex:10001476			South West Railway					
7	ex:10002283			Prydneprovskaya Railway					
8	ex:10002906			Odesa Railway					
9	ex:10003576			Donetsk railway					

Figure 3.4: A template that contains the fields and the rules for converting excel data into an owl file using the ROBOT ontology tool. The template is based on the data collected and feedback from the domain expert.

Ontology validation

And for the ninth step of ontology validation, we continued working with our domain expert and GIS expert to cross check all the process and data are correct and accurate. We also evaluated our ontology using some quality metrics, such as consistency, completeness, correctness, and reusability.

As we got more clear with the problem, and as the domain experts were getting a better understanding of what we were doing, we went back and forth with them to refine our ontology and data. We made some adjustments and improvements based on their feedback and suggestions.

3.2 Path-finding algorithm

In this section, we discuss how we use the path-finding algorithm to find the optimal path for grain transportation, based on the semantic and spatial knowledge from our ontology and the KnowWhereGraph. We describe the A* algorithm, which is a heuristic search algorithm that can find the optimal path between two locations, taking into account of certain criteria, such as distance, grain storage, and risk. We also explain how we implemented the A* algorithm in our application, what criteria and heuristics we used, and how we handled dynamic and uncertain situations.

3.2.1 The A* algorithm

The A* algorithm is a widely used path-finding algorithm that can find the optimal path between two locations in a graph, where each node has a cost associated with it. The algorithm works by maintaining an open set of candidate nodes, which are nodes that have not been explored yet, and a closed set of visited nodes, which are nodes that have been explored already. The algorithm starts with the source node in the open set, and at each step, it selects the node with the lowest cost from the open set, moves it to the closed set, and expands its neighbors. The cost of a node is calculated by adding two values: g(n), which is the actual cost from the source to the node n; and h(n), which is an estimate of the cost from n to the destination. The estimate h(n) is called a heuristic function, and it should be optimistic, meaning that it should never overestimate the actual cost. The algorithm terminates when either the destination node is moved to the closed set, or when the open set becomes empty. If the destination node is found, then the optimal path can be traced back by following the parent pointers of each node.

The A* algorithm is an extension of Dijkstra's algorithm[33], which is a path-finding algorithm that only uses g(n) as the cost function. Dijkstra's algorithm can find the optimal path in any graph with non-negative edge weights, but it can be slow and inefficient because it explores all possible paths equally. The A* algorithm improves on Dijkstra's algorithm by using h(n) as a guide to direct the search towards the destination more quickly and effectively. However, the A* algorithm requires a good heuristic function that can provide accurate and consistent estimates of the cost. If h(n) is zero for all nodes, then A* becomes Dijkstra's algorithm; if h(n) is very high for all nodes except the destination, then A* becomes a greedy best-first search, which is a path-finding algorithm that only uses h(n) as the cost function. Greedy best-first search can be very fast but not optimal, because it can get stuck in local minima or dead ends.[34]

The A* algorithm has some advantages over other path-finding algorithms for our domain of grain transportation. It can find the optimal path in terms of distance, grain storage, and risk, which are important criteria for our application. It can also adapt to dynamic and uncertain situations, such as changes in weather, traffic, or security conditions, by updating the cost and heuristic functions based on the latest information from the KnowWhereGraph.

3.2.2 Integration of A* in our application

In this subsection, we describe how we integrated the A* algorithm in our application using Python[35].We modified the A* algorithm into a recursive algorithm, where we can add some "must-pass" nodes, either can be high priority elevators because of their huge grain storage or their location, or hand-picked by domain expert, who says such location is high value and we must have the path that pass through these nodes. We made this modification to accommodate the preferences and constraints of our domain expert, and to ensure that our paths are feasible and realistic. The algorithm works as follows: The algorithm takes as input the start node, the goal node, and a list of must-pass nodes [A,B,C,...], where [A,B,C,...] are the names of the must-pass locations in any order. The algorithm recursively explores all possible paths in the list, based on the heuristics mentioned above. If the list is empty, then it simply applies the standard A* path finding. Figure 3.5 demonstrates how the algorithm works.

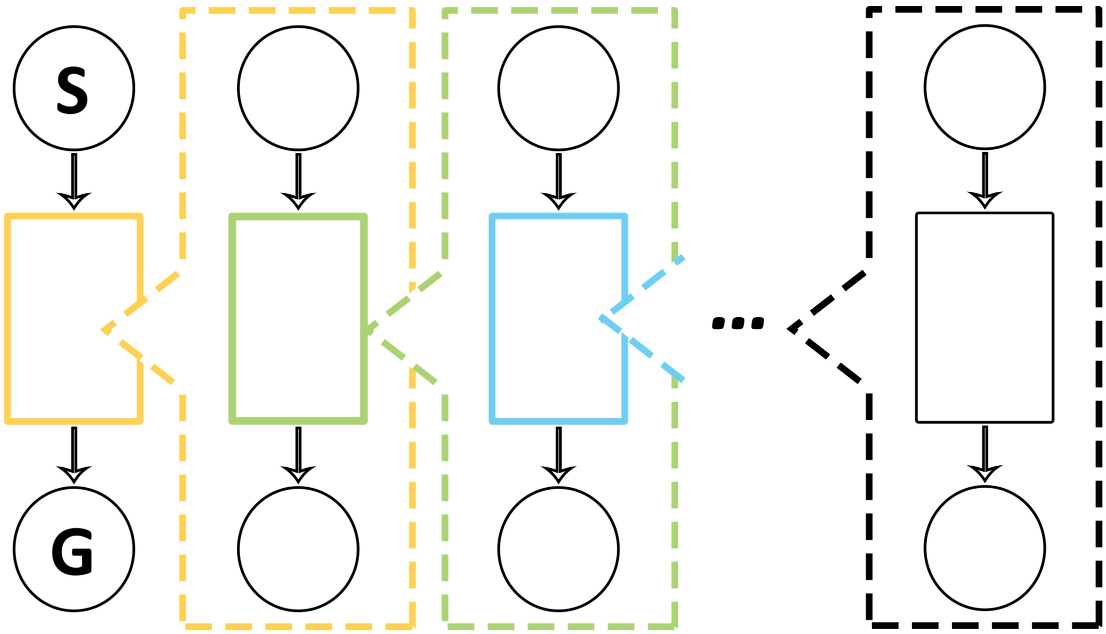


Figure 3.5: A schematic diagram of the recursive A* algorithm with must-pass nodes. The algorithm takes as input the start node S, the goal node G, and a list of must-pass nodes [A,B,C,...] in any order. The algorithm recursively explores all possible paths from S to G that pass through [A,B,C,...], using the heuristics of distance, elevation, and grain storage. The algorithm returns the optimal path that minimizes the total cost.

We integrated the A* algorithm as follows in algorithm 1:

We tested our implementation using some sample data from our domain expert, and evaluated our results using some quality metrics[12;36], such as performance, scalability, reliability, usability, and maintainability. We found that our application can find some accurate, efficient, and robust paths for grain transportation, and can handle dynamic and uncertain situations. However, we also encountered some challenges and limitations, such as missing or incorrect paths, incomplete or outdated data, and complex or ambiguous scenarios. We are working on improving our application to overcome these issues and provide better results. We also received positive feedback from our domain expert and GIS expert on the usability

Algorithm 1 A path-finding algorithm with must-pass nodes

Require: start node, goal node, list of must-pass nodes
Ensure: result path from start node to goal node passing through all must-pass nodes

```
if must-pass nodes is empty then
    resultpath ← shortestpath                                    ▷ using A-star algorithm
else
    resultpath ← []
    current ← start
    while mustpass ≠ null do
        shortest ← shortest in mustpass nodes from current node using A* algorithm
        resultpath.add(shortest ← A * (current, mustpass))
        must-pass nodes.remove(shortest node)
        current ← shortest
    end while
    resultpath.add(shortest ← A * (current, goal)
    resultpath.flatten()
end if
Show result
Handle any errors or exceptions that may occur
```

and usefulness of our application. They also suggested some possible improvements and extensions for our application, such as adding some features and options for filtering out the railway stations that is intended for passengers only, which is irrelevant in out application, and etc.

3.3 Challenges

In this section, we discuss the challenges that we faced during the data integration process. We explain how we dealt with these challenges using various techniques and tools, such as data cleaning, data transformation, data mapping, and data validation. We also describe how these challenges affected the quality and validity of our ontology component, and how we tried to overcome or mitigate them.

One of the challenges that we faced during the data integration process was the poor quality and format of the data provided by the domain expert. The data was in an Excel spreadsheet that had various issues, such as:

- Typos and spelling errors, mainly due to translation problems. For example, "ДЖАНКОЙ" in Ukrainian can be translated to either "Dzhankoy" or "Dzhankoi" in English, but the name has to be consistent for creating a proper data structure.

- Inconsistent separators for splitting the text. Sometimes it was a dash (-), sometimes it was a comma (,), and sometimes it was just an empty space ( ). This was because the domain expert obtained the information from various sources, hence the format was not unified.

- Duplicate names for different locations. Since some grain elevators in different regions were owned by the same corporation, they had the same name.

These issues made it difficult to create a data dictionary and to map the data to our ontology.

To fix these problems, we performed several steps to clean and transform the data using Python scripts. First, we converted the Excel spreadsheet into a CSV file, which is easier to process and manipulate. Second, we used regular expressions to extract the information for connections based on a pattern of location name, separator, distance, and KM. We handled different cases of separators and formats of location names. Third, we fixed the typos by generating a list of pairs with highly similar string names, and double checking with the domain expert. Fourth, we added a unique identifier to each name to avoid duplicate names that might belong to different locations in different regions. Finally, we validated the data by comparing it with other sources and checking for any errors or inconsistencies.

These steps helped us to improve the quality and format of the data, and to integrate it with our ontology. This in turn enabled us to use the data for our evaluation and to find optimal paths for grain transportation.

Another challenge that we faced during the data integration process was the complexity and diversity of the locations involved in grain transportation. We had to understand how the different stations were situated, and differentiate between passenger stops and commerce stops, small stops with no stations, etc. This required a lot of domain knowledge and consultation with the domain expert. We also had to use the Arc-GIS [2]system to quality control and identify possible problems with the locations, such as missing coordinates, incorrect names, or inaccurate connections. We also used our ontology component to help us better identify some of the clean up tasks, such as finding synonyms, resolving ambiguities, or adding annotations.

[2]ArcGIS is a geographic information system (GIS) software that allows users to create maps, analyze spatial data, and manage geographic information. https://www.arcgis.com/

Chapter 4

RESULTS AND CONCLUSIONS

In this chapter, we present the results of our application, which combines the ontology and the geospatial data and the A* algorithm to find optimal paths for grain transportation. We continue the chapter elaborating on the main findings of this research and implications of our research for theoretical basis of computer science.

4.1 Results

We applied our application to seven example locations that were provided by the domain expert, who wanted us to find the possible routes for grain transportation between them. We used the ontology and the geospatial data that we constructed in chapter 3 and the modified A* algorithm that we integrated in chapter 3.2.2 to find the optimal paths based on the criteria of distance, grain storage, and risk. We imported the resulting paths into the arc-GIS system for visualization and evaluation. We obtained four feasible and good paths for four of the example locations, which showed the potential and effectiveness of our application. However, we encountered some errors and challenges when applying our application to the other three example locations. For two of them, we could not find any path at all because some location in the must-pass node list was not reachable. For the remaining one, we got an error saying that the location was not present in the graph. We suspect that these problems were due to the missing or outdated data and/or the name duplication problems in our data set. For instance, in our data set, there was a connection between elevator A and elevator B, but B was a large-scale elevator that had different instances with the same name across the country, depending on the ownership and location. Therefore, we could not simply add a connection between A and B without specifying which B it was. We learned about this complexity by following the KNARM methodology, which involved going back and forth with the domain expert throughout the whole process. We could either ask the domain expert to manually check each individual elevator with name duplication problems, which would be too time-consuming and impractical, or we could automate this process by assuming that A connects to the closest B, since we had their coordinates and we calculated their distances using the haversine formula. The haversine formula is a mathematical equation that determines the great-circle distance between two points on a sphere given their longitudes and latitudes1. We applied it because the data that we have is geospatial coordinates of locations that are located on the Earth's surface, which is approximately spherical. By simply calculating their distance based on the coordinates using the simple distance formula: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ we would obtain inaccurate results because such formula is used to find the distance between any two points on a flat plane, while the Earth is curved. Therefore, we need to get a more accurate distance result by applying the haversine formula, which takes into account the curvature of the Earth and the angles between the points.

Due to time constraints, we have not addressed some data issues that affected our application performance. In the future, we will either ask the domain expert to verify each elevator with name duplication problems, or automate this process by connecting each elevator to the nearest one with the same name. We will elaborate on these solutions in chapter 5.2. In the following section, we will present and discuss the main findings and implications of our research. We will relate the key results of our data analysis to our research questions and objectives. We will also explore how our findings contribute to the theory, practice, and

Figure 4.1: A map of Ukraine created using the ArcGIS system for visualization, showing the locations of elevators (orange dots) and railroads (black lines). The map is blurred to protect sensitive data.

policy in our field of study.

4.2 Summary of Findings

We conducted a data analysis using the modified A* algorithm, a heuristic search algorithm that can find and update the optimal path between multiple locations, taking into account the criteria of distance, grain storage, and risk, as well as dynamic and uncertain situations. We used real-world data of grain transportation in Ukraine, which is a complex and dynamic domain facing a military invasion situation. We also integrated our data with the KnowWhereGraph, a cross-domain knowledge graph and geo-enrichment service stack that provides rich and up-to-date geospatial information. We developed our ontology using the KNARM methodology, which allows for creating and maintaining modular ontologies that can represent complex domains and support reasoning.

Our data analysis yielded the following main findings: The A* algorithm was able to find optimal paths for grain transportation in Ukraine, based on the criteria of distance, grain storage, and risk. The algorithm was also able to adapt to dynamic and uncertain situations by updating the path based on the latest information. The KnowWhereGraph provided valuable geospatial information that enhanced the path-finding process. The KnowWhereGraph included information such as population, economic indicators, geographical features, coordinates, connections, and security levels for each location. This information helped to refine the criteria of distance, grain storage, and risk for each path, and to provide more accurate and comprehensive results. The ontology we developed using the KNARM methodology represented the domain of grain transportation in Ukraine in a modular and reusable way. The ontology included concepts such as elevator/station, has grain storage size, coords, connections, etc. The ontology also reused some existing ontologies such as GeoSPARQL and FOAF to enrich our vocabulary. The ontology supported reasoning tasks such as consistency checking, validation, annotation propagation, diffing, etc. One of our goal of this research was to explore and investigate the effect of having and encoding different and many heuristics as part of the knowledge graph on the performance of the path-finding algorithm. Nonetheless, this research was limited by the lack of data for for different and many heuristics that could affect the performance of the path-finding algorithm. Therefore, we could not evaluate the impact of having and encoding such heuristics as part of the knowledge graph on the quality and efficiency of the path-finding process. However, we designed and implemented our ontology using KNARM's modular architecture, which enables us to easily integrate new knowledge graph modules for various heuristics, if they become available in the future. This way, we can leverage the existing data and reuse it for path-finding, without having to modify the ontology structure or the algorithm logic. In summary, our findings demonstrate the potential of using ontologies and knowledge graphs to enhance path-finding problems in complex and dynamic domains as a generalizable solution approach. We showed how the A* algorithm, the geo-spatial semantic web resources, and the ontology we developed using the KNARM methodology worked together to find optimal paths for grain transportation in Ukraine.

4.3 Implications for Theory

Our research contributes to the theory of ontology engineering and knowledge graph construction for complex and dynamic domains. We propose a novel application of the KNARM methodology and the KnowWhereGraph to address the problem of finding optimal paths for grain transportation in Ukraine. We show how these two components can work together to create a modular ontology that can represent the domain knowledge and support reasoning tasks. We also show how they can integrate with a heuristic search algorithm that can leverage the semantic and spatial knowledge to find accurate and efficient paths.

Our research also challenges the existing theory of path-finding algorithms for complex and dynamic domains by proposing a novel modification of the A* algorithm that can accommodate preferences and constraints from domain experts by adding some must-pass nodes to the path. Furthermore, we are using a novel combination of approaches and methods, and with that, we used semantic web, A* algorithm, optimization algorithms, and human expert in the loop, to find the optimal path in the dynamic domain. Such modification is important because it improves the feasibility and realism of the paths found by the algorithm, which are often ignored or overlooked by the existing theory in this area. The existing theory in this area does not provide adequate solutions or guidance for this modification, as most of the existing algorithms assume that the path is determined by only one criterion (such as distance), or that the path is independent of any external factors (such as preferences). Our research fills this gap by showing how this modification can improve the performance and usability of the A* algorithm, and how it can adapt to dynamic and uncertain situations by updating the path based on the latest information from the KnowWhereGraph.

Chapter 5

LIMITATIONS AND FUTURE DIRECTIONS

In this chapter, we discuss the limitations and future directions of our research. We acknowledge the challenges and shortcomings that we faced during our research process, and suggest some possible ways to overcome them or improve them in future work.

5.1 Limitations

Our research has several limitations that need to be addressed. Some of these limitations are related to the data completeness and its cleaning process, which will be discussed in the following. One of the main limitations of our research is the quality and availability of the data that we used for our ontology and knowledge graph. As mentioned in chapter 3.3, the data that we obtained from the domain expert was in an Excel spreadsheet that was inconsistent in format and had typos due to either human errors or other factors. This made it difficult to create a data dictionary and to map the data to our ontology. We had to perform several steps to clean and transform the data, such as using regular expressions, fixing typos, adding unique identifiers, etc. However, these steps were not reliable or accurate and could introduce errors or inconsistencies in the data. Moreover, the data that we obtained from the domain expert was not complete or up-to-date. There were some locations or connections that were missing or outdated in the data. This could affect the accuracy and completeness of our ontology and knowledge graph, and consequently, our path-finding algorithm. Furthermore, the data that we obtained from the domain expert was sensitive and confidential. This limited our ability to share or publish our data or our ontology publicly, which could reduce the reproducibility and reusability of our research. Another limitation of our research is the amount of time and effort that we spent on waiting for the new versions of data and data cleaning while implementing this methodology. Data cleaning is an essential step for ensuring the quality and validity of our ontology and knowledge graph, but it is also a tedious and time-consuming process that requires a lot of manual intervention and verification. We spent hours in the beginning to get the regular expression pattern right to capture what we need, and to generate some "questionable names" for the domain expert to double check. "By "questionable names", we mean a list of location names that had very similar spelling, but different properties such as coordinates. For example, some locations used the letter i or y interchangeably in their names. These locations were in the same region and very close to each other, which made it hard to distinguish them based on their names alone. After sending the file to the domain expert for the location name disambiguation, we then had to wait for weeks to receive the updated data from the domain expert, and then spend hours to clean and transform the data using various tools and techniques, such as Python scripts, regular expressions, etc. This delayed our progress and reduced our efficiency in developing and evaluating our application. Moreover, data cleaning is not a one-time task, but a continuous process that needs to be repeated whenever there is a change or an error in the data. Therefore, having a set of clean data is very important for implementing this methodology successfully and effectively.

5.2 Evaluation Plan and Future Directions

Our research has developed an application that integrates the ontology and the geospatial data and the A* algorithm to find optimal paths for grain transportation in Ukraine. However, our research also has some limitations and challenges that need to be addressed in future work. In this section, we present our evaluation plan for our application, which aims to assess its validity, reliability, and usefulness. We also discuss some possible future directions that will extend or improve our methodology and application. Some of these future directions are related to the data, the ontology, the knowledge graph, and the path-finding algorithm.

5.2.1 Evaluation Plan

In this section, we outline the evaluation plan of our application, which integrates the ontology and the geospatial data and the A* algorithm to find optimal paths for grain transportation. We explain the data sources, the metrics, and the expected outcomes of our evaluation. We also discuss the significance and implications of our evaluation plan, and how it relates to our research questions and objectives. Due to time constraints, we did not perform the evaluation during this research, but we intend to continue the research in the future. The purpose of our evaluation plan is to address the following questions:

- Q1: How accurate are the paths found by our application, compared to existing solutions, based on the criteria of distance, grain storage, and risk?
- Q2: How efficient are the paths found by our application, compared to existing solutions, based on the criteria of computational complexity and scalability?
- Q3: How robust are the paths found by our application, compared to existing solutions, based on the criteria of adaptability and resilience to dynamic and uncertain situations?

To answer these questions, we would use the following methods and criteria:

Q1: How accurate are the paths found by our application, compared to existing solutions, based on the criteria of distance, grain storage, and risk?

For question 1, which measures the accuracy of our application's paths based on distance, grain storage, and risk, we would compare the paths found by our application with the paths found by existing solutions, such as Google Maps or ArcGIS. We would use metrics such as total distance, total grain storage capacity, and total risk score to measure the accuracy of each path. We would also use statistical tests, such as t-test[37] or ANOVA[38], to determine if there are significant differences between the paths in terms of these metrics. To measure the accuracy of the distance criterion, we would calculate the mean absolute error (MAE) between the paths found by our application and Google Maps for each scenario. The MAE is a measure of how close the predictions are to the actual outcomes. It is calculated by taking the average of the absolute differences between the predicted and actual values. For example, if we have n scenarios, we can calculate the MAE as follows:

$$MAE = \frac{(|d_1 - d_2| + |d_3 - d_4| + \ldots + |d_n - d_{n+1}|)}{n} \tag{5.1}$$

where d_1, d_3, ..., d_n are the distances of the paths found by our application, and d_2, d_4, ..., d_{n+1} are the distances of the paths found by Google Maps.

The lower the MAE, the better the performance of our application compared to Google Maps. We would also use a t-test or an ANOVA to test if there is a significant difference between the MAEs of our application and Google Maps.

Q2: How efficient are the paths found by our application, compared to existing solutions, based on the criteria of computational complexity and scalability?

For question 2, which evaluates the efficiency of our application's paths based on computational complexity and scalability, we would compare the computational complexity and scalability of our application with those of existing solutions. We would use metrics such as running time, memory usage, and number of nodes expanded to measure the efficiency of each solution. We would also use statistical tests, such as t-tests or ANOVA, to determine if there are significant differences between the solutions in terms of these metrics.

Q3: How robust are the paths found by our application, compared to existing solutions, based on the criteria of adaptability and resilience to dynamic and uncertain situations?

And finally for question 3, which assesses the robustness of our application's paths based on adaptability and resilience to dynamic and uncertain situations, we would compare the adaptability and resilience of our application with those of existing solutions. We would use scenarios that simulate dynamic and uncertain situations that affect grain transportation in Ukraine, such as changes in weather, traffic, or security conditions. We would use metrics such as number of path updates, number of path failures, and number of path alternatives to measure the robustness of each solution.

Data sources

The data used in this study was provided by a world-renowned domain expert in grain transportation in Ukraine to use his data for this study. The data was sensitive and confidential, as it related to a war situation that could affect the security and economy of Ukraine. Therefore, we took several measures to protect the privacy and security of the data and the expert. These measures include:

- Anonymizing the data by removing any personal or identifiable information of the domain expert or any other parties involved in grain transportation.
- Storing the data on a local computer that is not connected to the internet or any external network, and deleting the data after the completion of the study.
- Reporting the results of the evaluation in an aggregated and generalized way, without

revealing any specific or detailed information about the locations, routes, or costs of grain transportation.

The data was an Excel spreadsheet file that had information about the locations of grain elevators and railway stations in Ukraine. We worked closely with the the domain expert and GIS expert to check and clarify the data. We also consulted with them to resolve any discrepancies or ambiguities in the data.

We expect that our evaluation plan will provide evidence for the validity, reliability, and usefulness of our application. This would demonstrate the potential of using ontologies and knowledge graphs to enhance path-finding problems in complex and dynamic domains. It would also provide new insights into the factors that affect grain transportation in Ukraine. We also expect that our evaluation plan will reveal some strengths and weaknesses of our application, as well as some opportunities and threats for its further development and we will use these findings to inform our future directions and recommendations for improving our application.

Data

One possible future direction to improve the accuracy and completeness of our ontology, knowledge graph, and path-finding algorithm is to obtain more complete and up-to-date data from domain experts or other sources, such as public databases or web scraping. This will help us to find more possible routes for all the example locations, and to avoid the missing or outdated data and/or the name duplication problems that we encountered in our data set. We will also automate or semi-automate the data cleaning and transformation process using techniques such as data integration, data quality assessment, and data wrangling. This will reduce the time and effort required for data cleaning and transformation, and increase efficiency and productivity in developing and evaluating our application. This will also help us to avoid the errors or inconsistencies that we introduced in our data set during the manual data cleaning and transformation steps.

Ontology

To enhance the expressiveness and semantics of our ontology, we will enrich or refine it with more concepts, properties, relations, axioms, and rules that are able to capture the complexity and diversity of the domain. This will enable more sophisticated reasoning and inference on our knowledge graph. For example, we will add more concepts or properties that are related to the security or risk levels of different locations or connections, such as has security level, has risk factor, etc. This will help us to find more optimal paths that minimize the risk of attacks or disruptions. Another possible direction is that we could enhance the interoperability and reusability of our ontology by aligning or linking it with other existing ontologies or vocabularies in the same or related domains, such as GeoNames and DBpedia. This will facilitate the integration or exchange of knowledge across different platforms or domains. For example, we could use owl:sameAs to link our location names with their counterparts in GeoNames or DBpedia, which will enable us to access more information or services from these resources.

Knowledge Graph

One possible future direction is to update or expand our knowledge graph with more data or information from different sources or modalities, such as text, images, and videos. This will improve the coverage and diversity of our knowledge graph, and enable more comprehensive and multi-modal analysis on our knowledge graph. Namely natural language processing techniques to extract information from text sources such as news articles or social media posts, which is able to provide us with more up-to-date information about the weather, traffic, or security conditions of different locations or connections. We could also use computer vision techniques to extract information from image or video sources such as satellite images or surveillance cameras, which could provide us with more visual information about the terrain, infrastructure, or activity of different locations or connections. Another possible direction is to apply or develop more advanced knowledge graph embedding or learning techniques, such as graph neural networks and attention mechanisms. This will improve the performance and accuracy of our knowledge graph embedding or learning models, and enable more efficient and effective query or retrieval on our knowledge graph. For example, we may use graph neural networks to capture the complex structure and semantics of our knowledge graph, which has the ability to enhance the representation and inference capabilities of our models. We could also use attention mechanisms to focus on the most relevant parts of our knowledge graph for a given query or task, which will improve the precision and recall of our models.

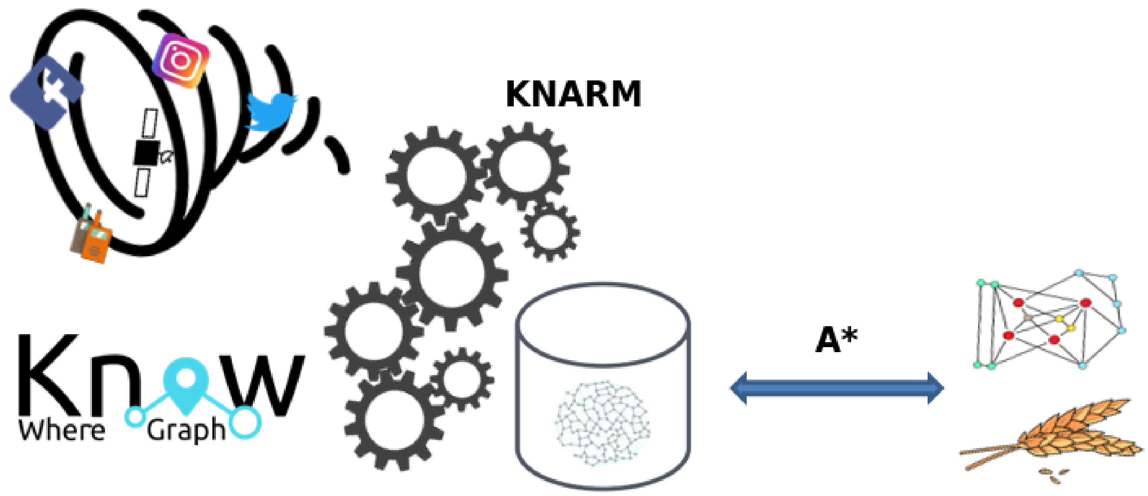


Figure 5.1: Integrating KnowWhereGraph and KNARM to enhance A* algorithm for optimal pathfinding in transportation of goods.[39]

Path-Finding Algorithm

In order to improve the flexibility and adaptability of our path-finding algorithm further, we will optimize or customize it for different scenarios or objectives in war situations, such as minimizing distance, time, risk, or cost. This will enable more optimal or near-optimal solutions for different path-finding problems. To achieve that, we will use different cost functions or heuristic functions for different scenarios or objectives, such as using Euclidean distance for minimizing distance, using travel time for minimizing time, using risk score for minimizing risk, etc. Additionally, We will apply different optimization techniques or algorithms for different scenarios or objectives, such as using linear programming for minimizing cost. Besides that, we will further improve the reliability and security of our path-finding algorithm by incorporating or developing more robust and resilient techniques to handle uncertainty or security issues in war situations, such as probabilistic models, fuzzy models, and encryption techniques. This will enable more accurate and trustworthy solutions for path-finding problems. In particular, we could use probabilistic models to represent the uncertainty of different factors that affect our path-finding problem, such as weather conditions, traffic conditions, security conditions etc., which has the ability to help us to find more realistic paths that account for uncertainty. We could also use fuzzy models to represent the vagueness of different factors that affect our path-finding problem , such as grain storage capacity, risk score, etc., which may help us to find more flexible paths that account for vagueness.

Bibliography

[1] Richard Cyganiak, David Wood, and Markus Lanthaler, editors. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation 25 February 2014, 2014. Available from http://www.w3.org/TR/rdf11-concepts/.

[2] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph, editors. *OWL 2 Web Ontology Language: Primer (Second Edition)*. W3C Recommendation 11 December 2012, 2012. Available from http://www.w3.org/TR/owl2-primer/.

[3] Holger Knublauch and Dimitris Kontokostas, editors. *Shapes Constraint Language (SHACL)*. W3C Recommendation 20 July 2017, 2017. https://www.w3.org/TR/shacl/.

[4] Hande Küçük McGinty. *KNowledge Acquisition and Representation Methodology (KNARM) and Its Applications*. PhD thesis, University of Miami, 2018.

[5] Satya Sahoo, Deborah McGuinness, and Timothy Lebo. PROV-o: The PROV ontology. W3C recommendation, W3C, April 2013. http://www.w3.org/TR/2013/REC-prov-o-20130430/.

[6] Shi-chao Sun, Zheng-yu Duan, and Dong-yuan Yang. Optimal paths planning in dynamic transportation networks with random link travel times. *Journal of Central South University*, 21(4):1616–1623, Apr 2014. ISSN 2227-5223. doi: 10.1007/s11771-014-2103-4. URL https://doi.org/10.1007/s11771-014-2103-4.

[7] Ning Jing, Yun-Wu Huang, and Elke A. Rundensteiner. Hierarchical optimization of optimal path finding for transportation applications. In *Proceedings of the Fifth International Conference on Information and Knowledge Management*, CIKM '96, page

261–268, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897918738. doi: 10.1145/238355.238550. URL https://doi.org/10.1145/238355.238550.

[8] Liang Shen, Hu Shao, Long Zhang, and Jian Zhao. The global optimal algorithm of reliable path finding problem based on backtracking method. *Mathematical Problems in Engineering*, 2017:4586471, Oct 2017. ISSN 1024-123X. doi: 10.1155/2017/4586471. URL https://doi.org/10.1155/2017/4586471.

[9] Hande Küçük Mcginty. Knowledge acquisition and representation methodology (kn-arm) and its applications. 2018. URL https://api.semanticscholar.org/CorpusID:64864929.

[10] Krzysztof Janowicz, Pascal Hitzler, Wenwen Li, Dean Rehberger, Mark Schildhauer, Rui Zhu, Cogan Shimizu, Colby K Fisher, Ling Cai, Gengchen Mai, et al. Know, know where, knowwheregraph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence. *AI Magazine*, 43(1):30–39, 2022.

[11] Peter Hart, Nils Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. doi: 10.1109/tssc.1968.300136. URL https://doi.org/10.1109/tssc.1968.300136.

[12] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37, 2021.

[13] Pascal Hitzler, Aldo Gangemi, and Krzysztof Janowicz. *Ontology engineering with ontology design patterns: foundations and applications*, volume 25. IOS Press, 2016.

[14] Steffen Staab and Rudi Studer, editors. *Handbook on Ontologies*. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-540-92673-3. URL https://doi.org/10.1007/978-3-540-92673-3.

[15] Pascal Hitzler and Adila Krisnadhi. A tutorial on modular ontology modeling with ontology design patterns: The cooking recipes ontology. *CoRR*, abs/1808.08433, 2018. URL http://arxiv.org/abs/1808.08433.

[16] Yu Lin, Saurabh Mehta, Hande Küçük-McGinty, John Paul Turner, Dusica Vidovic, Michele Forlin, Amar Koleti, Dac-Trung Nguyen, Lars Juhl Jensen, Rajarshi Guha, Stephen L Mathias, Oleg Ursu, Vasileios Stathias, Jianbin Duan, Nooshin Nabizadeh, Caty Chung, Christopher Mader, Ubbo Visser, Jeremy J Yang, Cristian G Bologa, Tudor I Oprea, and Stephan C Schürer. Drug target ontology to classify and integrate drug discovery data. *J. Biomed. Semantics*, 8(1):50, November 2017.

[17] Amar Koleti, Raymond Terryn, Vasileios Stathias, Caty Chung, Daniel J Cooper, John P Turner, Dušica Vidovic, Michele Forlin, Tanya T Kelley, Alessandro D'Urso, Bryce K Allen, Denis Torre, Kathleen M Jagodnik, Lily Wang, Sherry L Jenkins, Christopher Mader, Wen Niu, Mehdi Fazel, Naim Mahi, Marcin Pilarczyk, Nicholas Clark, Behrouz Shamsaei, Jarek Meller, Juozas Vasiliauskas, John Reichard, Mario Medvedovic, Avi Ma'ayan, Ajay Pillai, and Stephan C Schürer. Data portal for the library of integrated network-based cellular signatures (LINCS) program: integrated access to diverse large-scale cellular perturbation response data. *Nucleic Acids Res.*, 46 (D1):D558–D566, January 2018.

[18] NIH. Lincs information framework (life) to integrate and analyze diverse data set, Jul 2013. URL https://grantome.com/grant/NIH/U01-HL111561-02S1#panel-abstract.

[19] Saminda Abeyruwan, Uma D Vempati, Hande Küçük-McGinty, Ubbo Visser, Amar

Koleti, Ahsan Mir, Kunie Sakurai, Caty Chung, Joshua A Bittker, Paul A Clemons, Steve Brudz, Anosha Siripala, Arturo J Morales, Martin Romacker, David Twomey, Svetlana Bureeva, Vance Lemmon, and Stephan C Schürer. Evolving BioAssay ontology (BAO): modularization, integration and applications. *J. Biomed. Semantics*, 5(Suppl 1 Proceedings of the Bio-Ontologies Spec Interest G):S5, June 2014.

[20] Ubbo Visser, Saminda Abeyruwan, Uma Vempati, Robin P. Smith, Vance Lemmon, and Stephan C. Schürer. Bioassay ontology (bao): a semantic description of bioassays and high-throughput screening results. *BMC Bioinformatics*, 12(1):257, Jun 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-257. URL https://doi.org/10.1186/1471-2105-12-257.

[21] GeoNames. Geonames. URL https://www.geonames.org/.

[22] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 3540762973.

[23] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017.

[24] Sven Koenig and Maxim Likhachev. Fast replanning for navigation in unknown terrain. *Robotics, IEEE Transactions on*, 21:354 – 363, 07 2005. doi: 10.1109/TRO.2004.838026.

[25] Tianxing Wu, Arijit Khan, Melvin Yong, Guilin Qi, and Meng Wang. Efficiently embedding dynamic knowledge graphs. *Knowledge-Based Systems*, 250:109124, 2022. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2022.109124. URL https://www.sciencedirect.com/science/article/pii/S0950705122005548.

[26] Patrick Jaillet, Jin Qi, and Melvyn Sim. Routing Optimization Under Uncertainty. *Operations Research*, 64(1):186–200, February 2016. doi: 10.1287/opre.2015.1462. URL https://ideas.repec.org/a/inm/oropre/v64y2016i1p186-200.html.

[27] K. Ramasamy, A. Arokiasamy, and P. A. Balakrishnan. *Internet Routing—The State of the Art*, pages 37–60. Springer US, Boston, MA, 2002. ISBN 978-0-387-35600-6. doi: 10.1007/978-0-387-35600-6_2. URL https://doi.org/10.1007/978-0-387-35600-6_2.

[28] Pengtao Xie and Eric Xing. Cryptgraph: Privacy preserving graph analytics on encrypted graph, 2015.

[29] Rebecca C. Jackson, James P. Balhoff, Eric Douglass, Nomi L. Harris, Christopher J. Mungall, and James A. Overton. Robot: A tool for automating ontology workflows. *BMC Bioinformatics*, 20(1):407, Jul 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-3002-3. URL https://doi.org/10.1186/s12859-019-3002-3.

[30] Mark A. Musen. The protégé project: a look back and a look forward. *AI Matters*, 1(4): 4–12, 2015. doi: 10.1145/2757001.2757003. URL https://doi.org/10.1145/2757001.2757003.

[31] Grigoris Antoniou and Frank van Harmelen. *Web Ontology Language: OWL*, pages 67–92. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-24750-0. doi: 10.1007/978-3-540-24750-0_4. URL https://doi.org/10.1007/978-3-540-24750-0_4.

[32] Wikimedia Commons. Rail transport in ukraine, 2014. URL https://upload.wikimedia.org/wikipedia/commons/8/88/Rail_Map_Ukraine.png.

[33] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[34] Fan Xie, Martin Müller, and Robert Holte. Adding local exploration to greedy best-first search in satisficing planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), June 2014. doi: 10.1609/aaai.v28i1.9035. URL https://doi.org/10.1609/aaai.v28i1.9035.

[35] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[36] R. S. I. Wilson, J. S. Goonetillake, W. A. Indika, and Athula Ginige. Analysis of ontology quality dimensions, criteria and metrics. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, and Carmelo Maria Torre, editors, *Computational Science and Its Applications – ICCSA 2021*, pages 320–337, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86970-0.

[37] Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

[38] Ellen R Girden. *ANOVA: Repeated measures*. Number 84. Sage, 1992.

[39] Yinglun Zhang, Antonina Broyaka, Jude Kastens, Allen M. Featherstone, Cogan Shimizu, Pascal Hitzler, and Hande Küçük Mcginty. Sustainable grain transportation in ukraine amidst war utilizing knarm and knowwheregraph. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 742–745, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394192. doi: 10.1145/3543873.3587618. URL https://doi.org/10.1145/3543873.3587618.