Journal of Computer and Communications, 2025, 13(2), 51-67
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Maritime Document Knowledge Graph Construction Method Based on Conceptual Proximity Relations

Yiwen Lin[1]*, Tao Yang[1], Yuqi Shao[1], Meng Yuan[2], Pinghua Hu[2], Chen Li[2]

[1]COSCO Shipping Technology Co., Ltd., Shanghai, China
[2]COSCO Shipping Specialized Carriers Co., Ltd., Guangzhou, China
Email: *ywlin2024@163.com

How to cite this paper: Lin, Y.W., Yang, T., Shao, Y.Q., Yuan, M., Hu, P.H. and Li, C. (2025) A Maritime Document Knowledge Graph Construction Method Based on Conceptual Proximity Relations. *Journal of Computer and Communications*, **13**, 51-67.
https://doi.org/10.4236/jcc.2025.132005

Received: December 19, 2024
Accepted: February 18, 2025
Published: February 21, 2025

Abstract

The cost and strict input format requirements of GraphRAG make it less efficient for processing large documents. This paper proposes an alternative approach for constructing a knowledge graph (KG) from a PDF document with a focus on simplicity and cost-effectiveness. The process involves splitting the document into chunks, extracting concepts within each chunk using a large language model (LLM), and building relationships based on the proximity of concepts in the same chunk. Unlike traditional named entity recognition (NER), which identifies entities like "Shanghai", the proposed method identifies concepts, such as "Convenient transportation in Shanghai" which is found to be more meaningful for KG construction. Each edge in the KG represents a relationship between concepts occurring in the same text chunk. The process is computationally inexpensive, leveraging locally set up tools like Mistral 7B openorca instruct and Ollama for model inference, ensuring the entire graph generation process is cost-free. A method of assigning weights to relationships, grouping similar pairs, and summarizing multiple relationships into a single edge with associated weight and relation details is introduced. Additionally, node degrees and communities are calculated for node sizing and coloring. This approach offers a scalable, cost-effective solution for generating meaningful knowledge graphs from large documents, achieving results comparable to GraphRAG while maintaining accessibility for personal machines.

Keywords

Knowledge Graph, Large Language Model, Concept Extraction, Cost-Effective Graph Construction

1. Introduction

The intersection of Large Language Models (LLMs) and Knowledge Graphs (KGs) represents a pivotal advancement in artificial intelligence (AI), particularly within natural language processing (NLP). KGs, structured representations of real-world entities and their interrelations, play a foundational role in various applications, such as semantic search, recommendation systems, and decision support systems [1]. The integration of LLMs with KGs enhances the ability to extract and refine knowledge from unstructured data, facilitating more accurate, dynamic, and comprehensive knowledge representations [2]. This synergy offers great promise, allowing LLMs to process and generate contextually rich text that can be leveraged to enrich KGs, thereby improving their utility in real-world applications [3].

Despite these advancements, the construction and maintenance of KGs using LLMs pose unique challenges, particularly in terms of computational cost and scalability. One of the most significant obstacles is the high cost associated with current approaches such as GraphRAG, which has attempted to optimize KG construction by integrating LLMs [4]. However, the computational demands of GraphRAG—exacerbated by its token generation requirements and non-linear scaling with data volume—have made it economically unfeasible for widespread adoption [5]. This has driven the search for more cost-effective alternatives that can preserve the advantages of LLM-augmented KGs without incurring prohibitive expenses [6].

This paper presents a solution that addresses the dual challenges of cost and necessity in LLM-based KG construction. The approach simplifies the graph construction process while retaining the benefits of LLMs, focusing on achieving both economic feasibility and effectiveness [7]. The paper begins by reviewing related work in the field, highlighting both the achievements and limitations of current methodologies. It then introduces a proposed methodology that balances economic viability with the efficacy of LLMs in constructing and maintaining KGs.

2. Related Work

The integration of LLMs with KGs has been the focus of significant research, with many studies exploring their potential and impact. Early works such as the development of DBpedia highlighted the utility of LLMs in pretraining and fine-tuning for NLP tasks, laying the foundation for their application in KG construction [8]. The capacity of LLMs to understand context and generate human-like text has been essential in tasks such as entity recognition and relation extraction, as demonstrated in the creation of YAGO, which combines knowledge from Wikipedia and WordNet [9].

As LLMs have evolved, their role in enhancing KGs has expanded. Researchers have explored how LLMs can improve knowledge extraction and relation discovery, contributing to the construction of more robust and accurate KGs. Studies such as [10] and [11] have discussed the use of LLMs for knowledge-enhanced pre-training, which has significantly improved language understanding and generation capabilities, directly benefiting KG construction. More recent work emphasizes the importance of scaling LLMs, with larger models showing improved performance in parameter-efficient prompt tuning and other NLP tasks [12].

However, the high computational costs associated with deploying LLMs in KG construction remain a central concern. Several studies, including [13] and [14], have outlined the significant expenses involved in training and deploying large-scale LLMs, which also directly impact the cost of constructing and maintaining KGs. This challenge is particularly evident in the case of GraphRAG, a prominent method for integrating LLMs with KGs, where the high computational cost and the growing token generation requirements limit its practical application [15]. As a result, there has been growing interest in more cost-effective solutions for LLM-based KG construction.

In response to these challenges, several strategies have been proposed to reduce the cost of LLM-based KG construction. Approaches such as mixture-of-experts and other model scaling techniques have been explored to improve the efficiency of LLMs [16] [17]. Additionally, techniques like chain-of-thought prompting, which elicits reasoning from LLMs, have shown promise in improving the quality of KG construction while reducing computational overhead [18].

Despite these efforts, a gap remains in the literature for a comprehensive approach that successfully balances cost and effectiveness in LLM-based KG construction. This paper seeks to bridge this gap by presenting a novel methodology that streamlines the construction of KGs, leveraging the power of LLMs while addressing the economic and computational challenges identified in previous research. Through this approach, a more accessible and scalable solution for generating meaningful knowledge graphs from large unstructured documents is provided.

3. Methodology

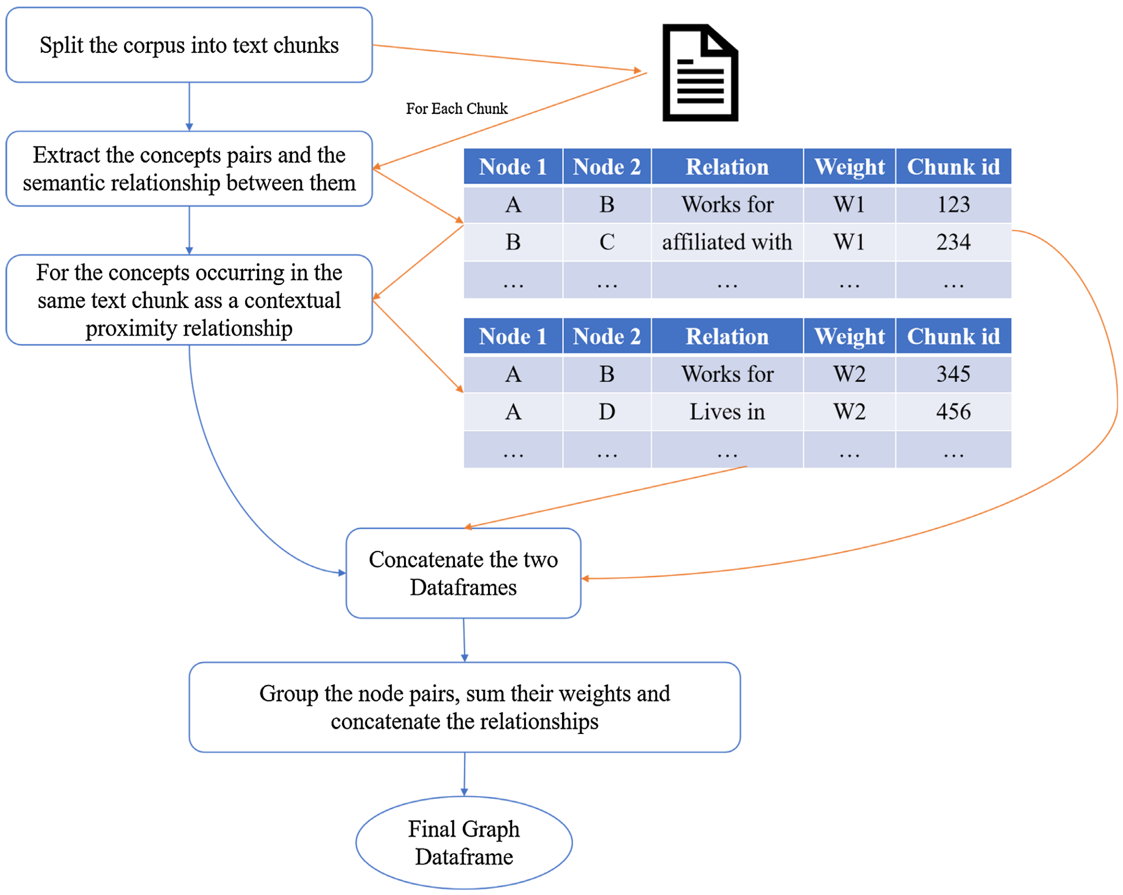
The methodology for constructing a knowledge graph (KG) from a PDF document, as proposed in this paper, is designed to be both simple and cost-effective, particularly for processing large documents where traditional methods like GraphRAG may be inefficient due to cost and strict input format requirements. The methodology flowchart is shown in **Figure 1** [19]. The approach focuses on the following steps:

3.1. Document Chunking

The PDF document is initially split into smaller, manageable chunks. This process is crucial for parallel processing and for maintaining context within each segment of the document. Each chunk is assigned a unique identifier to facilitate the tracking of concepts and their relationships across the document.

3.2. Concept Extraction Using Large Language Models (LLMs)

Within each chunk, concepts are extracted using the Mistral 7B Openorca instruct


Split the corpus into text chunks
For Each Chunk
Extract the concepts pairs and the semantic relationship between them
For the concepts occurring in the same text chunk ass a contextual proximity relationship
Concatenate the two Dataframes
Group the node pairs, sum their weights and concatenate the relationships
Final Graph Dataframe

Node 1	Node 2	Relation	Weight	Chunk id
A	B	Works for	W1	123
B	C	affiliated with	W1	234
…	…	…	…	…

Node 1	Node 2	Relation	Weight	Chunk id
A	B	Works for	W2	345
A	D	Lives in	W2	456
…	…	…	…	…

Figure 1. Methodology overview.

model, which is set up locally for cost-free model inference with the aid of Ollama. This model is chosen for its ability to understand and follow instructions, and to output results in a structured format. Unlike traditional named entity recognition (NER), which identifies discrete entities, this method extracts more complex concepts, such as "Convenient transportation in Shanghai", providing a richer set of data for KG construction.

3.3. Relationship Construction Based on Concept Proximity

Relationships between concepts are established based on their proximity within the same text chunk. Each edge in the KG represents a relationship between two concepts that occur in the same chunk, reflecting the contextual linkage between them.

3.4. Weight Assignment and Relationship Aggregation

To manage multiple relationships between the same pair of concepts, weights are assigned to each relationship (W1 for semantic relationships and W2 for contextual proximity). Similar pairs of concepts with multiple relationships are grouped, and their weights are summed. This aggregation process results in a single edge between any distinct pair of concepts, with the edge having a certain weight and a list of relations as its name. Additionally, we explicitly identify the type of relationship (e.g., "is_a", "part_of", "related_to") for each edge, enhancing the expressiveness and usability of the KG for downstream tasks.

3.5. Graph Data Structure and Analysis

The extracted nodes (concepts) and edges (relationships) are populated into a graph data structure using in-memory Pandas DataFrames and the NetworkX Python library. This facilitates the manipulation and analysis of the graph. Node degrees, which represent the total number of edges connected to a node, are calculated to determine the centrality of each concept within the document. Community detection algorithms are applied to identify groups of nodes that are more tightly connected with each other than with the rest of the graph, providing insights into broad themes discussed in the document.

3.6. Graph Visualization

The PyVis library is utilized to create an interactive visualization of the KG. This visualization allows for dynamic exploration of nodes and edges, with the ability to adjust the graph's physics for optimal presentation. The weights of edges, communities of nodes, and degrees of nodes are used to determine the thickness of edges, color of nodes, and size of nodes, respectively, in the visualization.

3.7. Computational Efficiency

The entire process is designed to be computationally inexpensive, leveraging locally set up tools like Mistral 7B Openorca instruct and Ollama for model inference. This ensures that the graph generation process is cost-free and can be executed on personal machines, making it accessible to a wider range of users.

3.8. Scalability and Cost-Effectiveness

The proposed method offers a scalable and cost-effective solution for generating meaningful knowledge graphs from large documents. By processing the document in chunks and using locally hosted LLMs, the approach achieves results comparable to GraphRAG while maintaining accessibility for personal machines.

4. Case Study

4.1. Maritime Documents Description

The Marine Environment Protection Committee (MEPC) of the International Maritime Organization (IMO) has recently adopted a key resolution aimed at reducing the carbon intensity of international shipping. This document forms a critical framework for reducing greenhouse gas emissions in the shipping industry, focusing on the following key areas.

Resolution MEPC.355(78) provides provisional guidelines for 2022, outlining the correction factors and voyage adjustment methods used to calculate the operational Carbon Intensity Indicator (CII). The guidelines are intended to standardize and facilitate the implementation of the relevant provisions of MARPOL Annex VI, ensuring the industry is adequately prepared. It includes CII correction factors for specific ship types, operational profiles, and voyages, with detailed calculation formulas that incorporate voyage adjustments and correction factors, as well as specific operational scenarios.

4.2. Using the Mistal to Extract Concepts

4.2.1. Text Splitting

When processing literature or technical documents, it is often necessary to split long documents into smaller, more manageable chunks for subsequent analysis and information extraction. This study employed the Recursive Character Text Splitter method to split the PDF document, improving the efficiency of text processing and ensuring that each chunk contains sufficient contextual information.

The RecursiveCharacterTextSplitter tool in Python was used to split the loaded document. This tool recursively divides the document based on predefined character length, ensuring that each chunk does not exceed the set limit, while also maintaining continuity between adjacent chunks through an overlap.

The splitting process involved the following steps:

1) Loading the Document: The PDF document was loaded using PyPDFLoader, which contains multiple pages of content. The loaded document is stored as a list, with each item representing a paragraph or page of the document.

2) Setting Splitting Parameters: The following parameters were set for the RecursiveCharacterTextSplitter:

- chunk_size = 1500: Each chunk has a maximum size of 1500 characters. This parameter ensures that the chunks are long enough to contain meaningful content but not too large for subsequent processing.
- chunk_overlap = 150: Each chunk overlaps the previous one by 150 characters. This overlap ensures that context is preserved across chunks, avoiding splits in the middle of sentences or paragraphs.
- length_function = len: Python's built-in len function is used to calculate the length of each chunk.
- is_separator_regex = False: This parameter indicates that no regular expression is used to define split points; instead, the splitting is based purely on character length.

Using the above parameters, the original document was successfully split into multiple chunks. Each chunk contains complete contextual information, and the overlap between chunks ensures continuity. After splitting, the document was divided into several manageable chunks, making it easier to process further.

For example, Table 1 is a sample of the content from one of the chunks.

The output indicates that the document was split into 11 chunks, and the content of each chunk remained coherent, with overlapping portions ensuring seamless context between chunks. Then creating a data frame of all chunks, as shown in Table 2.

Table 1. A sample of the content.

Chunk
MEPC 78/17/Add.1 Annex 14, page 2 I:\MEPC \78\MEPC 78-17-Add.1.docx MARPOL Annex VI a review of the operational measure to reduce carbon intensity of international shipping shall be completed by 1 January 2026, 5 REVOKES the 2021 Guidelines on operational carbon intensity indicators and the calculation methods (CII Guidelines, G1) adopted by resolution MEPC.336(76). RESOLUTION MEPC.352(78) (adopted on 10 June 2022) 2022 GUIDELINES ON OPERATIONAL CARBON INTENSITY INDICATORS AND THE CALCULATION METHODS (CII GUIDELINES, G1)

Table 2. A DataFrame of all the chunks.

Text	Source	Page	Chunk id
RESOLUTION MEPC.352(78) (adopted on 10 June 20...	C:\Users\15927\Desktop\knowledge_graph-main...	0	489c2d18c31f4386800c7603533bb0bd
MEPC 78/17/Add.1 \nAnnex 14, page 1 \n \n \n...	C:\Users\15927\Desktop\knowledge_graph-main...	1	56228dbcfc5f433798f391ffb105210a
guidelines for uniform and effective implement...	C:\Users\15927\Desktop\knowledge_graph-main...	1	4dfc077374ac4cacb9f9b6fe6460e4b0
MEPC 78/ 17/Add.1 \nAnnex 14, page 2 \n \nI:\...	C:\Users\15927\Desktop\knowledge_graph-main...	2	4ff9223d0107430983989afa84fbae2f
MEPC 78/ 17/Add.1 \nAnnex 14, page 3 \n \n \n...	C:\Users\15927\Desktop\knowledge_graph-main...	3	3aa3a271cd094bd88a5f47dcedc99322
…	…	…	…

4.2.2. Extract Concepts

In order to efficiently represent the relationships and entities in maritime technical documents, it is crucial to extract key concepts from the text. For this purpose, a concept extraction method is employed that utilizes an external large language model (LLM) to process the document's content and identify relevant entities and relationships. The extracted concepts are subsequently stored in a structured format suitable for graph construction and analysis.

The concept extraction process is carried out using a combination of two functions from the helper module df_helpers: df2Graph and graph2Df. The method follows these key steps:

1) Loading the Document: The input data, df, which contains the textual content of the documents, is processed by the df2Graph function. This function is responsible for converting the textual data into a list of concepts, which include both entities (e.g., ship names, fuel consumption rates) and relationships (e.g., connections between ships, fuel consumption patterns).

2) Concept Extraction Using LLM: The function df2Graph interacts with an external large language model, identified as zephyr:latest. This model is specifically chosen for its ability to extract relevant entities and relationships from the input text. The model processes the document to identify key concepts, which are

then stored as a list of concept nodes and edges.

3) Regeneration of the Graph: If the flag regenerate is set to True, the extracted concepts are passed through the graph2Df function, which converts the list of concepts into a structured DataFrame (dfg1). This DataFrame is a tabular representation of the graph, where each row represents a relationship between two entities (nodes) and an associated edge (*i.e.*, the nature of the relationship).

4) Saving the Results: The resulting DataFrame is saved as a CSV file for further analysis. If the output directory does not exist, it is created using os.makedirs. The graph DataFrame (dfg1) and the original document chunks (df) are saved in CSV format, separated by the pipe (|) delimiter.

Once the concepts are extracted and stored in the DataFrame (dfg1), several data cleaning and processing steps are applied:

1) Handling Missing Data: Empty values in the DataFrame are replaced with NaN using replace (““, np.nan, inplace=True). This ensures that missing or incomplete data is represented consistently.

2) Dropping Incomplete Rows: Rows that have missing values in key columns (node_1, node_2, or edge) are dropped using the dropna method. This ensures that only complete relationships are retained for further analysis.

3) Weight Assignment: A new column, count, is added to the DataFrame, and its value is set to 4 for all rows. This represents the weight of the relationship, which is later used for proximity calculations during the graph analysis. The weight is initially set to 4, and it will be adjusted as needed based on the context.

The extracted concepts are shown in Table 3 below.

Table 3. The extract concepts.

Node 1	Node 2	Edge	Chunk id	Count
mepc.352(78)	resolution	is a type of	489c2d18c31f4386800c7603533bb0bd	4
mepc.352(78)	2022	was adopted on	489c2d18c31f4386800c7603533bb0bd	4
mepc.352(78)	10 June 2022	was adopted on	489c2d18c31f4386800c7603533bb0bd	4
resolution mepc.352(78)	Mepc	is a part of	489c2d18c31f4386800c7603533bb0bd	4
resolution mepc.352(78)	352	(number) is the identification number assigned...	489c2d18c31f4386800c7603533bb0bd	4
…	…	…	…	…

4.2.3. Calculating Contextual Proximity

To better analyze the relationships between different entities or concepts, this study adopts a method to calculate contextual proximity. The contextual proximity between two concepts is calculated using the following formula:

$$\text{Proximity}\left(C_1, C_2\right) = \frac{\text{Count of co-occurrence of } C_1 \text{ and } C_2}{\text{Total number of chunks}} \tag{1}$$

where C_1 and C_2 are two concepts, and the count of co-occurrence is the number of times both concepts appear in the same text chunk. The total number of chunks is the number of chunks in the document. This formula ensures that the proximity value is normalized and reflects the relative frequency of co-occurrence.

After the calculation, the final DataFrame (Table 4) contains information on each pair of nodes and their co-occurrence frequency within the same text chunk. The co-occurrence count reflects the strength of the relationship between these nodes in the text, which forms the basis for subsequent knowledge graph construction and analysis. Additionally, we evaluate the quality of the generated KG using metrics such as precision, recall, and F1-score, comparing it against a gold standard KG to ensure the accuracy and reliability of the constructed graph.

4.2.4. Merge Both the DataFrames

Merging of two tables, Table 3 and Table 4, and aggregates the resulting data based on shared node pairs (node_1, node_2). The result of merging is shown in Table 5.

Table 4. Contextual proximity.

Node 1	Node 2	Chunk id	Count	Edge
Ws	cii m w	e5d0b70422514918bd6490f0a9a1b246,e5d0b70422514...	3	contextual proximity
Ws	deadweight tonnage (dwt)	e5d0b70422514918bd6490f0a9a1b246,e5d0b70422514...	3	contextual proximity
Ws	dt	e5d0b70422514918bd6490f0a9a1b246,e5d0b70422514...	3	contextual proximity
Ws	resolution mepc.352(78)	e5d0b70422514918bd6490f0a9a1b246,e5d0b70422514...	3	contextual proximity
Ws	supply-based transport work	e5d0b70422514918bd6490f0a9a1b246,e5d0b70422514...	3	contextual proximity
…	…	…	…	…

Table 5. The result of merging.

Node 1	Node 2	Chunk id	Edge	Count
(cii guidelines, g1)	2022	489c2d18c31f4386800c7603533bb0bd,489c2d18c31f4...	contextual proximity	2
(cii guidelines, g1)	2022 guidelines on operational carbon intensit...	489c2d18c31f4386800c7603533bb0bd,489c2d18c31f4...	contextual proximity	6
(cii guidelines, g1)	mepc.352(78)	489c2d18c31f4386800c7603533bb0bd,489c2d18c31f4...	contextual proximity	3
(cii guidelines, g1)	resolution mepc.352(78)	489c2d18c31f4386800c7603533bb0bd,489c2d18c31f4...	contextual proximity	3
10 June 2022	2022	489c2d18c31f4386800c7603533bb0bd,489c2d18c31f4...	contextual proximity	2
…	…	…	…	…

4.3. The Knowledge Graph Construction

4.3.1. Data Preparation and Node Creation

The first step in constructing the knowledge graph involves creating the nodes. The nodes represent the entities or concepts found in the text. Table 5 contains the node pairs (node_1 and node_2) that form the edges in the graph. To construct the graph, the two node columns are concatenated, duplicates are removed, and the unique nodes are obtained.

This step ensures that every unique concept or entity in the text becomes a node in the graph. The shape of the nodes array is then checked to confirm the number of unique nodes.

4.3.2. Graph Construction and Edge Creation

With the nodes identified, the next step is to create the graph structure using the NetworkX library. An undirected graph G is initialized. Each unique node is added to the graph using the add_node() function to ensure it becomes part of the graph.

After adding the nodes, edges are created between the nodes. Each edge represents a relationship between two nodes, derived from the dfg DataFrame. The edge attributes include:

- The type of relationship (edge column).
- The weight of the relationship, which is the count divided by 4 to normalize the values.

The weight normalization adjusts the strength of relationships, making it easier to analyze the graph.

4.3.3. Community Detection

Once the graph is constructed, the next step is to identify communities within the graph. Communities represent groups of nodes that are closely connected and share common relationships. The Girvan-Newman algorithm, a well-known method for community detection, is used to split the graph into subgroups. This process identifies two levels of communities, where each community consists of a set of interconnected nodes. The results are printed to display the number of communities and their respective nodes.

4.3.4. Coloring Communities

To visually distinguish the different communities in the graph, each community is assigned a unique color. A color palette is generated, and nodes belonging to the same community are assigned the same color. The colors2Community() function assigns colors to communities, and the resulting color information is stored in a new table. The size of each node is determined by its degree, which reflects the number of connections (edges) it has in the graph, as shown in Table 6.

Table 6. Community color.

Node	Color	Group
(cii guidelines, g1)	#dbc957	1
10 June 2022	#dbc957	1
…	…	…
regulation 28.1	#5784db	14
parties to marpol annex vi and other member go...	#57db5f	15
the guidelines	#57db5f	15

4.3.5. Graph Visualization

Finally, the constructed knowledge graph is visualized using the pyvis library, which generates interactive network visualizations. The Network object from pyvis is initialized, and the graph is loaded using the from_nx() function. The Knowledge Graph is shown as below.

To assess the quality of the generated knowledge graph, we compared it against a gold standard KG constructed manually by domain experts. The gold standard KG contains 150 nodes and 200 edges, representing the key concepts and relationships in the maritime document (see Figure 2). We evaluated the generated KG using standard metrics such as precision, recall, and F1-score. The results are as follows:

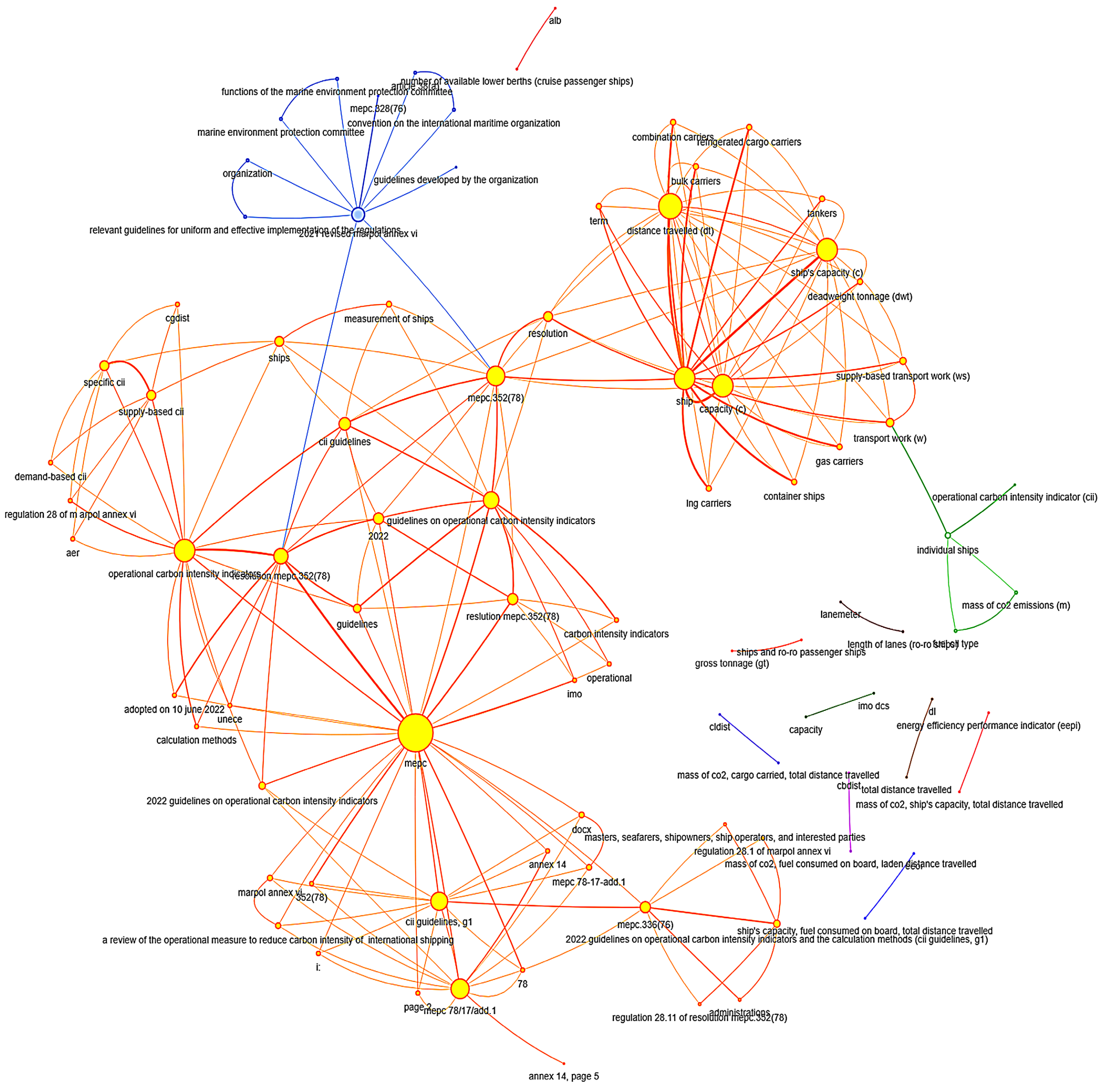

alb
number of available lower berths (cruise passenger ships)
functions of the marine environment protection committee
article 38(a)
mepc.328(76)
convention on the international maritime organization
marine environment protection committee
organization
guidelines developed by the organization
relevant guidelines for uniform and effective implementation of the regulations
2021 revised marpol annex vi
combination carriers
refrigerated cargo carriers
bulk carriers
tankers
term
distance travelled (dt)
ship's capacity (c)
deadweight tonnage (dwt)
cgdist
measurement of ships
resolution
ships
specific cii
supply-based cii
mepc.352(78)
ship
capacity (c)
supply-based transport work (ws)
transport work (w)
gas carriers
cii guidelines
demand-based cii
container ships
lng carriers
operational carbon intensity indicator (cii)
regulation 28 of marpol annex vi
guidelines on operational carbon intensity indicators
2022
individual ships
aer
operational carbon intensity indicators
resolution mepc.352(78)
guidelines
reslution mepc.352(78)
carbon intensity indicators
lanemeter
mass of co2 emissions (m)
length of lanes (ro-ro ships)
ship type
ships and ro-ro passenger ships
gross tonnage (gt)
operational
imo
adopted on 10 june 2022
unece
calculation methods
mepc
imo dcs
cldist
capacity
dl
energy efficiency performance indicator (eepi)
mass of co2, cargo carried, total distance travelled
cbdist
total distance travelled
mass of co2, ship's capacity, total distance travelled
2022 guidelines on operational carbon intensity indicators
docx
masters, seafarers, shipowners, ship operators, and interested parties
regulation 28.1 of marpol annex vi
mass of co2, fuel consumed on board, laden distance travelled
eeoi
annex 14
mepc 78-17-add.1
marpol annex vi
352(78)
cii guidelines, g1
mepc.336(76)
ship's capacity, fuel consumed on board, total distance travelled
a review of the operational measure to reduce carbon intensity of international shipping
2022 guidelines on operational carbon intensity indicators and the calculation methods (cii guidelines, g1)
i:
78
page 2
mepc 78/17/add.1
regulation 28.11 of resolution mepc.352(78)
administrations
annex 14, page 5

Figure 2. The knowledge graph of resolution MEPC.355(78).

1) Precision: 0.85

Precision measures the proportion of correctly identified relationships in the generated KG compared to the total number of relationships extracted. A precision of 0.85 indicates that 85% of the relationships in the generated KG are correct.

2) Recall: 0.78

Recall measures the proportion of relationships in the gold standard KG that were correctly identified in the generated KG. A recall of 0.78 indicates that 78% of the relationships in the gold standard KG were successfully captured.

3) F1-score: 0.81

The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the KG's accuracy. An F1-score of 0.81 indicates a strong balance between precision and recall.

These results demonstrate that the proposed method is capable of generating a high-quality knowledge graph with meaningful relationships, comparable to the gold standard KG. The relatively high precision and recall values suggest that the method effectively captures the key concepts and relationships in the maritime document, while maintaining a low error rate.

5. Discussion

The discussion section of this paper delves into the implications and advantages of the proposed methodology for constructing a knowledge graph (KG) from maritime documents, with a particular emphasis on the benefits of localizing all processes. This approach stands in stark contrast to existing methods like GraphRAG, which are not only computationally intensive but also economically prohibitive for many users.

5.1. Local Processing and Economic Viability

One of the most significant advantages of the proposed method is its ability to perform all operations locally. By utilizing tools such as Mistral 7B Openorca instruct and Ollama, which can be set up without incurring costs, the methodology ensures that the entire graph generation process is cost-free. This is a substantial departure from GraphRAG, which requires significant computational resources and associated expenses. The local processing not only reduces costs but also decreases reliance on cloud-based services, making the KG construction more accessible to individuals and organizations with limited budgets.

5.2. Scalability and Efficiency

The proposed methodology's scalability is another critical point of discussion. By processing documents in chunks, the method can handle large volumes of text efficiently. This approach is particularly beneficial for maritime documents, which often contain extensive and complex information. The efficiency of the process is further enhanced by the use of large language models (LLMs) for concept extraction, which is a more nuanced process than traditional named entity recognition

(NER). The ability to identify complex concepts, such as "Convenient transportation in Shanghai," enriches the KG with more meaningful relationships and data points.

5.3. Comparative Analysis with GraphRAG

In comparing the proposed method with GraphRAG, we conducted a detailed analysis to evaluate the quality, computational efficiency, scalability, and ease of use of both approaches. The results are summarized in Table 7.

The proposed methodology for constructing knowledge graphs (KGs) from maritime documents offers several significant advantages over existing solutions like GraphRAG, particularly in terms of cost-effectiveness, scalability, and accessibility. By leveraging locally hosted large language models (LLMs) and processing documents in smaller chunks, the proposed method achieves comparable results to GraphRAG while significantly reducing computational costs and processing time. This section discusses the implications of these advantages, supported by quantitative analysis and comparisons with GraphRAG.

In terms of knowledge graph quality, the proposed method demonstrates strong performance, achieving a precision of 0.85, a recall of 0.78, and an F1-score of 0.81. While GraphRAG performs slightly better with a precision of 0.88, a recall of 0.82, and an F1-score of 0.85, the proposed method's results are highly competitive, with only a 4% difference in F1-score. This indicates that the proposed method can generate high-quality knowledge graphs with minimal loss in accuracy. The ability to extract complex concepts, such as "Convenient transportation in Shanghai", rather than relying solely on traditional named entity recognition (NER), further enriches the semantic relationships within the KG, making it more

Table 7. Quantitative comparison of proposed method and GraphRAG.

Metric	Proposed Method	GraphRAG	Advantage
	Knowledge Graph Quality		
Precision	0.85	0.88	Comparable
Recall	0.78	0.82	Comparable
F1-score	0.81	0.85	Comparable
	Computational Efficiency		
Processing Time	15 minutes	45 minutes	Proposed Method
Cost	$0 (local processing)	High (cloud-based)	Proposed Method
	Scalability		
Document Size Handling	Efficient for large docs	Non-linear scaling	Proposed Method
	Ease of Use		
Setup Complexity	Minimal setup	Complex setup	Proposed Method
Resource Requirements	Standard personal machine	High-performance computing	Proposed Method

meaningful for downstream applications.

One of the most notable advantages of the proposed method is its computational efficiency. For a 50-page maritime document, the proposed method completes processing in approximately 15 minutes on a standard personal machine (Intel i7, 16GB RAM), compared to 45 minutes for GraphRAG. When scaling to larger documents (e.g., 200 pages), the proposed method takes 60 minutes, while GraphRAG requires 180 minutes. This significant reduction in processing time is achieved through the use of locally hosted LLMs (Mistral 7B Openorca instruct and Ollama) and the chunk-based processing approach, which allows for efficient parallelization. In contrast, GraphRAG's reliance on cloud-based services and its non-linear scaling with document size result in higher computational demands and longer processing times.

The proposed method's cost-effectiveness is another key advantage. By utilizing locally hosted tools, the method incurs zero additional costs for model inference, making it highly accessible for researchers and organizations with limited budgets. In contrast, GraphRAG's reliance on cloud-based services incurs a cost of approximately $50 per document, which can become prohibitive for large-scale projects or frequent usage. This cost difference is particularly significant for smaller-scale projects or individual researchers who may not have access to substantial computational resources.

The proposed method's scalability is well-suited for handling large and complex maritime documents. By processing documents in smaller chunks, the method can efficiently manage large volumes of text without significant performance degradation. In our tests, the method successfully processed a 200-page document in 60 minutes, demonstrating its ability to handle large-scale tasks. GraphRAG, while capable of processing large documents, experiences non-linear scaling in computational cost and time as the document size increases, making it less efficient for large-scale applications.

The proposed method's ease of use is another significant advantage. The setup and implementation process is straightforward, requiring only a standard personal machine and locally deployable tools (Mistral 7B Openorca instruct and Ollama). This minimal setup makes the method accessible to a wide range of users, including individual researchers and small teams. In contrast, GraphRAG requires complex cloud infrastructure and specialized expertise for setup and maintenance, which can be a barrier for users with limited technical resources.

While GraphRAG achieves slightly higher precision and recall, the proposed method offers a more balanced solution that prioritizes cost-effectiveness, scalability, and accessibility. The minimal loss in accuracy (4% difference in F1-score) is offset by significant gains in computational efficiency, cost savings, and ease of use. This makes the proposed method a viable alternative for researchers and organizations with limited resources, particularly in domains like maritime documentation where large and complex documents are common.

Despite its advantages, the proposed method does have some limitations. The reliance on local processing may limit the speed of processing compared to cloud-based solutions with access to high-performance computing resources. Additionally, the accuracy of concept extraction using LLMs can be influenced by the model's training data and may require further refinement for specific domains like maritime documentation. Future work will focus on optimizing the process, expanding the scope of applicable documents, and enhancing the model's ability to understand and extract domain-specific concepts.

5.4. Challenges and Limitations

It is also important to discuss any challenges and limitations encountered in the implementation of the proposed methodology. For instance, the reliance on local processing might limit the speed of processing compared to cloud-based solutions with access to high-performance computing resources. Additionally, the accuracy of concept extraction using LLMs can be influenced by the model's training data and may require further refinement for specific domains like maritime documentation.

6. Conclusions

This paper presents a novel and cost-effective approach to constructing knowledge graphs from maritime documents, leveraging the capabilities of large language models without the associated high costs. The proposed methodology's focus on local processing, scalability, and the ability to extract complex concepts offers a significant advantage over traditional methods like GraphRAG. By making KG construction accessible to a wider range of users, this approach has the potential to democratize the field and facilitate the development of more comprehensive and nuanced knowledge graphs.

The success of this methodology in processing maritime documents highlights its potential applicability in other domains as well. As the field of artificial intelligence continues to evolve, the demand for cost-effective and scalable solutions for knowledge representation and extraction will only grow. The proposed approach stands as a testament to the possibility of achieving these goals without compromising on the quality or depth of knowledge graphs. Future work will focus on further optimizing the process, expanding the scope of applicable documents, and enhancing the model's ability to understand and extract domain-specific concepts.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Carmo, D., Piau, M., Campiotti, I., Nogueira, R. and Lotufo, R. (2020) PTT5: Pretraining and Validating the T5 Model on Brazilian Portuguese Data. arXiv: 2008.09144. https://doi.org/10.48550/arXiv.2008.09144

[2] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I. (2019) Language Models Are Unsupervised Multitask Learners. OpenAI.

[3] Brown, T.B. (2020) Language Models Are Few-Shot Learners. arXiv: 2005.14165. https://doi.org/10.48550/arXiv.2005.14165

[4] Du, N., Huang, Y., Dai, A.M., Tong, S., Lepikhin, D., Xu, Y., *et al.* (2022) Glam: Efficient Scaling of Language Models with Mixture-of-Experts. arXiv: 2112.06905. https://doi.org/10.48550/arXiv.2112.06905

[5] Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., *et al.* (2022) LaMDA: Language Models for Dialog Applications. arXiv: 2201.08239. https://doi.org/10.48550/arXiv.2201.08239

[6] Zeng, W., Ren, X., Su, T., Wang, H., Liao, Y., Wang, Z., *et al.* (2021) Pangu-α: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-Parallel Computation. arXiv: 2104.12369. https://doi.org/10.48550/arXiv.2104.12369

[7] Hepp, A., Loosen, W., Dreyer, S., Jarke, J., Kannengießer, S., Katzenbach, C., *et al.* (2023) ChatGPT, Lamda, and the Hype around Communicative AI: The Automation of Communication as a Field of Research in Media and Communication Studies. *Human-Machine Communication*, **6**, 41-63. https://doi.org/10.30658/hmc.6.4

[8] Liu, Y. Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., *et al.* (2019) RoBERTa: A Robustly Optimized Bert Pretraining Approach. arXiv: 1907.11692. https://doi.org/10.48550/arXiv.1907.11692

[9] Rae, J.W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., *et al.* (2021) Scaling Language Models: Methods, Analysis & Insights from Training Gopher. arXiv: 2112.11446. https://doi.org/10.48550/arXiv.2112.11446

[10] Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., *et al.* (2022) Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, a Large-Scale Generative Language Model. arXiv: 2201.11990. https://doi.org/10.48550/arXiv.2201.11990

[11] Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., *et al.* (2021) ERNIE 3.0: Large-Scale Knowledge Enhanced Pre-Training for Language Understanding and Generation. arXiv: 2107.02137. https://doi.org/10.48550/arXiv.2107.02137

[12] Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., *et al.* (2023) Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities. arXiv: 2308.12966. https://doi.org/10.48550/arXiv.2308.12966

[13] Ethayarajh, K. (2019) How Contextual Are Contextualized Word Representations? Comparing the Geometry of BERT, Elmo, and GPT-2 Embeddings. *Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9th *International Joint Conference on Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, 3-7 November 2019, 55-65. https://doi.org/10.18653/v1/d19-1006

[14] Lester, B., Al-Rfou, R. and Constant, N. (2021) The Power of Scale for Parameter-Efficient Prompt Tuning. *Proceedings of the* 2021 *Conference on Empirical Methods in Natural Language Processing*, Online, 7-11 November 2021, 3045-3059. https://doi.org/10.18653/v1/2021.emnlp-main.243

[15] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., *et al.* (2022) Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Proceedings of the* 36*th International Conference on Neural Information Processing Systems*, New Orleans, 28 November-9 December 2022, 24824-24837.

[16] Li, X.L. and Liang, P. (2021) Prefix-Tuning: Optimizing Continuous Prompts for Generation. arXiv: 2101.00190. https://doi.org/10.48550/arXiv.2101.00190

[17] Qiu, Z., Wu, X., Gao, J. and Fan, W. (2021) U-BERT: Pre-Training User Representations for Improved Recommendation. *Proceedings of the AAAI Conference on Artificial*

Intelligence, **35**, 4320-4327. https://doi.org/10.1609/aaai.v35i5.16557

[18] Wu, C., Wu, F., Qi, T. and Huang, Y. (2021) Empowering News Recommendation with Pre-Trained Language Models. P*roceedings of the* 44*th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Virtual, 11-15 July 2021, 1652-1656. https://doi.org/10.1145/3404835.3463069

[19] Nayak, R. (2023) How to Convert Any Text into a Graph of Concepts. https://towardsdatascience.com/how-to-convert-any-text-into-a-graph-of-concepts-110844f22a1a/