Chair of Spacecraft Systems
TUM School of Engineering and Design
Technical University of Munich

Knowledge Extraction Tool for System Model Generation in an LLM-Enhanced Spacecraft Design Assistant

Semester Thesis

Brynel Raj[✉], **Ramón María García Alarcia, and Alessandro Golkar**

TUM School of Engineering and Design, Technical University of Munich

✉ brynel.raj@tum.de

September 08, 2024

Abstract — The potential of Knowledge graphs (KGs) to provide sophisticated querying, intelligent data management, and effective knowledge representation is driving their use in a large number of domains. This paper presents a framework for building Knowledge Graphs (KGs) using Large Language Models (LLMs) and demonstrates how these KGs can be used to build space mission system models. The study introduces a knowledge extraction tool that uses LLMs to process unstructured text and turn it into structured knowledge graphs. These KGs—which are made up of entities and relationships —are then used to map data to OPM (Object Process Methodology) models. The OPM model's elements and hierarchical structure is used as an ontology to label the entities that are identified during the KG construction.
This study assesses the tool's ability to accurately map the retrieved data onto the OPM model in addition to producing KGs. Although the LLM worked well for extracting entities and relationships—especially for producing high-level system components—it had trouble labeling lower-level, domain-specific entities in the Knowledge Graph. Despite these limitations, this framework provides a promising initial solution and establishes the groundwork for future research and development targeted at obtaining a more thorough automatic population of OPM models.

1 Introduction

The rapid advancements in data processing have presented new opportunities for enhancing systems engineering, especially in the Space sector. Vast amounts of unstructured textual space mission data such as mission documents and reports are generated. Extraction of meaningful data from this corpus of unstructured text can be greatly beneficial in the early stages of spacecraft design [1]. Currently engineers spend a significant amount of time retrieving relevant data from these documents, which can slow down the design process. KGs offer a semantic method of organising data by capturing the intricate relationships between the entities [2]. They enable more efficient querying and retrieval of data. When combined with frameworks for system model generation, KGs can offer an automated and scalable approach to populate system models. This combination helps engineers make more informed decisions and offers a coherent representation of mission data.

1.1 State of the Art

Knowledge graphs defined as *a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent relations between these entities* [3], provide a semantic method of arranging and accessing data, which has completely changed database administration. They enable us to encode smart behaviour in the data directly rather than having to encode it into applications [4].
KGs, as opposed to conventional relational databases, are able to record the intricate relationships between entities, making data querying more user-friendly and effective. With diverse applications across different sectors, the most common uses of knowledge graphs include question answering, recommendation systems, semantic search, and advanced analytics. The concept of KGs has evolved significantly since the early 2000s, when researchers began to explore ways to make data more interconnected. One of the first significant milestones in this field was Google's introduction of its Knowledge Graph in 2012 [5] which aimed to enhance search by understanding the context of queries. Following this, many other companies announced the development of knowledge graphs for database management.
Over the years more sophisticated methods, such as the Graph Neural Networks [6], have been introduced for improving tasks such as entity linking and relationships extraction by using the inherent structure of the graphs themselves. Despite their great potential,

these approaches are severely constrained by the labor-intensive process of manually building and maintaining the knowledge graph itself. Time and effort are required by domain experts and graph developers to ensure quality of domain-specific models.
Recent research into the domain of KGs explores the automation of tasks such as named entity recognition (NER) [7] and relation extraction (RE) [8], which are the foundational elements for knowledge graph construction, by using LLMs [9]. Studies by [2] and [10] indicated that the use of generative LLMs outperformed other relation extraction methods and proved promising in reducing the manual effort and significantly automating the KG creation process. Trajanoska et.al [2] also highlighted the use of ontologies and that more specific prompts leads to better defined concepts, relations and instances. Although further refinements are needed, these techniques provide promising potential to scale KG creation across other domains, particularly in the space industry which is rich in ontologies and vast datasets.
Although the practical use of KG in the space industry is not extensively documented, its underlying principles and benefits are evident. As a part of the DEA project [1], the use of KGs in information extraction and organization highlights the potential use of this framework during the early stages of space mission design. Further research by [11] showed how NLP and semantic similarity techniques can be used to benefit from historical mission data within a unified Knowledge Graph during the initial conceptual design Phases 0 and A.
Adding to the enhancement of extraction and retrieval of information by integrating LLMs with KGs, Berquand et.al, [12] demonstrated the capability of transformer-based models to automatically parse documents related to space missions and populate KGs while enabling sophisticated information retrieval and inference. Utilizing the structured framework of the engineering models, Darm et.al, [11], presented a systems engineering approach based on linguistic similarity analysis between the metadata associated with each EM. This makes it easier to compare previous missions and suggest engineering components for new space missions and improves the efficiency of space mission design.

1.2 Proposed Framework

This term paper investigates a method for constructing Knowledge Graphs (KGs) specifically created for spacecraft design, with an emphasis on generating intricate KGs that can encompass multiple documents related to a single mission. The primary objective of developing these KGs is to use it to automatically populate OPM system models, providing a framework that is structured and captures all the relevant mission data. These models will serve as databases of structured data, expediting the analysis and the early design phases (Phases 0 and A in ECSS terms) in spacecraft design. KGs provide a means of methodically organizing complex and unstructured data, which is useful while designing space missions and improving the overall design process.
The production of Knowledge Graphs (KGs) from structured data found in Engineering Models (EMs) has been established in earlier work [11]. These structured knowledge graphs (KGs) analyze comprehensive subsystem information to facilitate activities like as design optimization and component suggestion. The early stages of spacecraft mission design have been expedited by this method. This study, on the other hand, aims to create a more comprehensive knowledge graph by obtaining information from unstructured mission documents (raw text), such as Word and PDF files. Its goal is to convert unstructured text into structured KGs using Large Language Models (LLMs), which will then be utilized to fill system models. This method aims at tackling the difficulty of transforming unstructured data while enhancing the corpus of knowledge, leading to more thorough and in-depth system models for spacecraft design.

2 Methodology

2.1 Approach

The presented work follows a four-step approach, the detailed workflow of the proposed framework is shown in Figure 1:

1. Text Extraction: Unstructured text from mission documents is extracted and cleaned before being divided into sections that follow the original document's structure in an ordered JSON format.

2. Ontology from OPL: The Object process language (OPL) generated by the Object process diagram (OPD) is used in this study to create an ontology. A unique identifier is assigned to each element of the system model which defines its location in the system model and is aimed at guiding the LLM while labeling the entities, and

further mapping them back to the system model elements.

3. KG construction: KGs in this study are generated by using an LLM (GPT-4o) [13] which essentially extracts entities and relationships from each section of the extracted text. The identified entities are labeled using a hierarchical ontology that is derived from the OPM system model for spacecraft design, which facilitates their mapping to system models. In addition to the text and the ontology, the whole document is also processed by the prompt to maintain overall consistency over the labelled entities.

4. Data mapping to OPM: The corresponding data in the KGs associated with the labelled entities is then mapped to the OPM system model. This step attempts at mapping the relevant extracted data alongside the elements in the OPM model ensuring all applicable mission details are represented within the system.

2.1.1 PDF extraction

This step transforms unstructured text from mission documents into a semi-structured format. This involves text cleaning and text segmentation. Text cleaning consists of removing unnecessary elements such as irrelevant symbols, redundant space and metadata. The document is split into sections of texts which are then processed for entity and relationship extraction. Accurate layout extraction is a challenge with multi-page, variably formatted PDFs. It is an issue yet to be addressed by open source technologies. Layout-PDFReader [14] and other AI-based applications try to automatically recognize paragraphs, headings, and sections in PDF files. These techniques, however, frequently fall short of the precision needed for lengthy, intricate papers. Regular expressions (RegEx) were used for section separation in this study because of their consistency in handling text files with consistent syntax. RegEx [15], while more reliable in identifying precise patterns, are also time-consuming and ineffective for documents with varying formats.

2.1.2 OPL Ontology

Object Process Methodology: Object-Process Methodology [16] (OPM, ISO 19450) is an MBSE language and methodology used to build domain independent conceptual models of various systems. It is used to depict the behaviour and strucure of complex systems. OPM consists of OPD (Object process Diagram) and OPL (Object process language). OPM systems are visualised by the use of OPD and OPL is the correspoding language generated by the OPD describing the object, processes and the relationships between them.

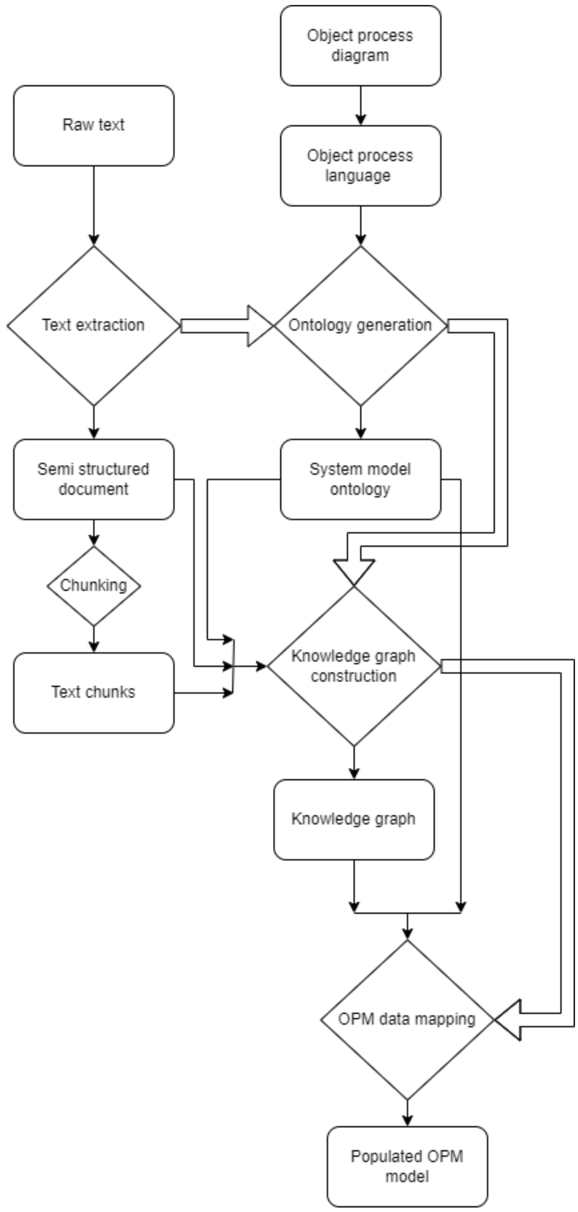

Figure 1 Flowchart of the framework for text extraction, ontology generation, KG construction, and OPM data mapping

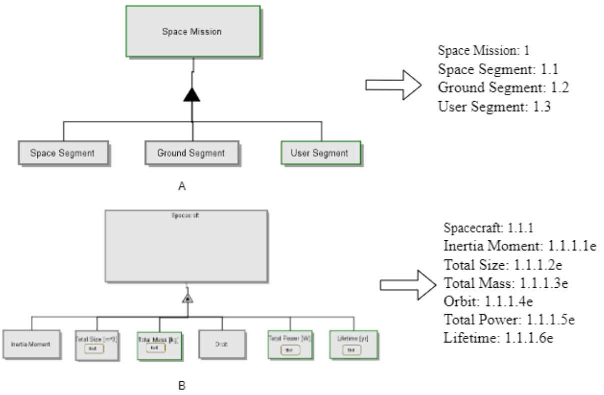

Figure 2 Hierarchical decomposition of the system model [17] of a space mission (A) and spacecraft (B) with corresponding identifiers for system segments and lower-level elements such as total mass, size, and power.

System Model: The system model being reconstructed is a hierarchical breakdown of a space mission [17]. As shown in the Figure: 2(A) the space mission consists of three segments: Space segments, Ground segment and a User segment. Each of these segments are further broken down into detailed lower level components maintaining a structured hierarchy. This model features two types of relationships between the objects: "consists of" and "exhibits". The higher level elements and their constituent parts are connected by the relationship "consists of" and the elements displaying metrics for example total mass and total size are connected to their parent elements by the relationship "exhibits". These are the elements which have the TBD (To Be Determined) values, which must be filled in with the relevant data in order to complete the system model.

OPL Ontology: An ontology is a formal representation of knowledge inside a domain, offering an organized framework of entities and their relationships. As mentioned in [2], including an ontology in the prompt greatly improves entity extraction accuracy. This ontology is defined and structured using the Object-Process Language (OPL), which is derived from the Object-Process Diagram (OPD). This method makes use of the system model's object titles and its hierarchical structure as the required ontology. A numerical identifier that corresponds to each object's place in the hierarchy is allocated to it in the system model as shown in Figure:2. In the hierarchy, the space mission is assigned the number "1," the space segment "1.1," the ground segment "1.2," and so on. Additionally, a "e" is used to distinguish items linked by

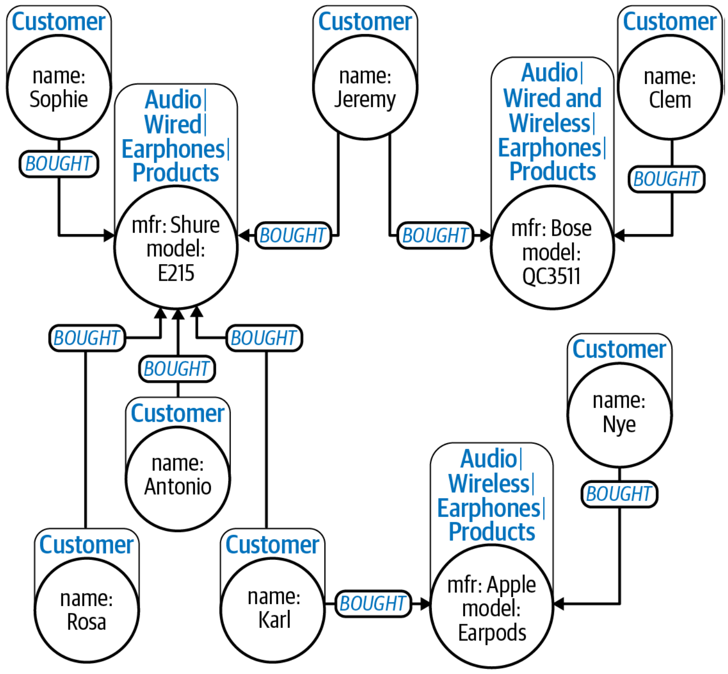

Figure 3 A property graph model [4] showing customer purchases of audio products

"exhibits" from the items linked by "consists of" relationships, which frequently indicate numerical values or certain qualities. For example, the identifier for "Spacecraft" would be "1.1.1," and its total mass would be "1.1.1.1e." These identifiers are made part of the prompt, with the aim of directing the Large Language Model (LLM) in correctly labeling the identified entities and mapping those entities to the appropriate locations within the system model.

2.1.3 KG construction

KG Organizing Principle The construction of knowledge graphs (KG) follows the general structure of a KG which consists of nodes which are connected to each other by edges. In this work the nodes are termed as entities and the edges are termed as relationships. This knowledge graph follows the property graph model [4] which supports labelled nodes (entities), directions for edges (relationships) provided by source and target entities and properties for both nodes (entities) and edges (relationships). Figure: 3 shows an example of the property graph model.

1. Entities consists of three main attributes
   a) Entity name
   b) Entity type
   c) Entity properties

2. Relationships consists of four main attributes
    a) Relationship name
    b) Relationship property
    c) Source Entity
    d) Target entity

```
[
  {
"Entities": [
      {
        "name": "Airbus Defence and Space SAS",
        "type": "Institution",
        "properties": "Lead of Concept A, mission prime for Consortium A"
      }
    ],
    "Relationships": [
      {
        "relationship_name": "leads",
        "source": "Airbus Defence and Space SAS",
        "target": "Concept A",
        "properties": "Lead of Concept A in CAIRT mission studies"
      }
    ]
  }
]
```

Figure 4 Example of an entity and relationship pair generated during Knowledge Graph construction

Entity and Relationship extraction Entity and relationship extractions are the two main tasks of knowledge graph construction. Many publicly available transformers based on BERT are available for performing tasks like Named Entity Recognition and Relationship extraction. Models such as BERT and RoBerta are pretrained on large language models and can be fine tuned for performing tasks on domain-specific data. For example, the SpaceTransformers introduced by [12] are trained specifically on a corpus of space systems data for the task of Concept recognition. Despite their potential, these models struggled with context sensitivity. Models like GPT 4 address these issues and represent a significant advancement in the domain of entity and relationship extraction tasks. Recent studies highlight that GPT 4 exhibits good performance in tasks related to KG construction [9]. These models further excel at reasoning tasks over complex context-based data.
GPT-4o was used for the presented work. One of the main benefits of using GPT models is their ability to generate strucutred data in JSON format by assigning JSON object as the response format. Since the results of Large Language Models (LLMs) can sometimes be unexpected or difficult to manage in free format , this feature ensures that the extracted entities and relationships are rendered in a structured and consistent format. An example of an entity and relationship pair created during the KG creation process is shown in Figure:4. Details such as the entity name ("Airbus Defence and Space SAS") and properties ("Lead of Concept A, mission prime for CAIRT mission") are included in the "entities" section. The relationship between two entities (in this case, "Airbus" and "Concept A") is shown in the "relationships" section, with the relationship being ("leads"), along with other properties that are associated with this link.

KG construction Prompt The structure of the prompt employed for the construction of the Knowledge Graph (KG) followed the principles essentials of effective prompting for Large Language Models (LLMs) [18]. According to some essential prompting rules for LLMs , it is important to be clear in defining tasks , provide clear context , and reduce ambiguity in order to guide the model ouputs towards the desired structure.
For this work, the prompt had three inputs and one output:

1. Inputs
    a) Extracted text chunk from the mission document
    b) OPL Ontology
    c) Full mission document
2. Output
    a) JSON object structure to hold a list of entities and relationships.

The prompt instructed the model to extract entities and relationships from the provided text chunk, while using the OPL Ontology to label the identified entities based on their corresponding elements in the Ontology. This was aimed at ensuring that the extracted entities maintained consistent labeling throughout the process. Since the prompt only processes one text chunk for the mission document at a time, the entire document was included as input to maintain context and consistency in entity labeling across the different sections of the document. This was done to ensure that the extracted entities and relationships aligned coherently even when processed in smaller chunks.

2.1.4 OPM Mapping

In the final stage, the mapped entities and relationships from the Knowledge graph (KG) are aligned with the OPM model. The mapping process involves searching the KG for entities that match labels taken from the OPL ontology . This set of ontological elements act as identifiers within the system model , directing the representation of entities in the KG to their respective components in the OPM hierarchy. The entities, their properties, and the properties of its relationships with other entities is extracted. This was intended to take in to account not only entities and their properties but also the properties of the relationships with other entities to maximize the extraction of relevant information form the text. The output of this stage includes the specific ontology element, its hierarchical identifier, the corresponding name of the entity, and all its associated properties.

3 Validation Approach

It is essential to validate the reliability of the generated KGs and final system models to ensure their accuracy and reliability , thereby ensuring the consistency of the structured data .
The validation process in this study is divided into two main parts: Validation of the knowledge graph itself and the validation of the mapped data for the OPM model. It is challenging to create a universal framework for validating KGs because each application has its own requirements, which makes validation methods contextual [19]. However, the commonly used methods are based on two main strategies : Intrinsic and Extrinsic evaluation.
Intrinsic: The intrinsic aspects are independent of the use case context, these criteria reflect the logical consistency. Such as **Semantic Validation** which evaluates the semantic correctness of the extracted entities and relationship pairs and **Completeness** which evaluates the comprehensive assessment of the model's ability to accurately identify the pairs.
Extrinsic: The extrinsic aspects depend on the context of the knowledge graph and evaluate the reliability of the KG applied to the domain specific use case. Such as **Accuracy of entity labeling** which measures how accurately are the entities in the KG recognized and categorized according to the predefined ontology.

4 Results

4.1 Validation: KG

Subsections from a mission document "Earth Explorer 11 Candidate Mission CAIRT: Report for Assessment" [20] were manually annotated and compared with the entity and relationship pairs produced by the model for the intrinsic validation of the Knowledge Graph.

Semantic Validation Semantic validation focused on assessing the accuracy of the retrieved relationships and entities. GPT-4o performed well in this domain, in line with other studies that shows its strength in reasoning and inference tasks—tasks that are intimately linked to the creation of entity-relationship pairings.

Accuracy of entity labeling The LLM model exhibited certain limitations in entity labeling from a given ontology. These limitations were mostly because of the highly domain specific nature of the task. This was made clear when the model had trouble correctly labeling components that were situated at lower levels of the model's breakdown structure. Rather than being given the precise, domain-specific identities that the OPM model requires, these lower-level elements were commonly misclassified or given generic labels. For example, the figure:5 shows one of the few misclassified low-level entities by the LLM.

```
{
  "name": "thermal gradients",
  "type": "new_entity",
  "properties": "minimised between different optical components"
},
```

Figure 5 Misclassified entity.

4.2 Validation: Mapped data onto the OPM model

The validation of the OPM model was conducted on a use case basis, where a sample text was generated using GPT-4o. This text was used as a test case to manually verify that the proposed framework appropriately populates the OPM model. The generated text was run through the framework to assess its performance by verifying the accuracy of the extracted data in the final OPM model.
50 random ontology elements were provided to guide the content of the generated text. The random elements were a mix of both high-level and low-level elements, 42 and 8 elements respectively. High-level elements refer to the system components without specific numerical property while the lower level elements represent the numerical data points, often associated with numerical values such as mass, size, or power.
Sample data corresponding to the chosen ontological elements was generated and then inserted into paragraphs across seven chapters . The generated text was then processed using the proposed framework to evaluate its ability to extract relevant entities and relationships and further map the generated data back to the OPM model.

The model was able to correctly identify and label 42 out of the 50 high-level elements in the generated text. All the 42 elements were the high-level elements which were correctly categorized within the hierarchical system. The remaining 8 lower level elements were identified as properties of the higher level elements rather than being labelled as a unique elements as shown in Table 1.

OPM Element Name	**Identified Entity Name**
Payload	Lunar Module
Unique Identifier	**Properties**
1.1.1.8	1. mass 15000 kg 2. measured 4m 3. Payload for lunar landing 4. power 200W

Table 1 The table presents the final mapped data for the element "Payload"

The framework successfully identified the relevant data corresponding to the high level OPM elements. However, it demonstrated limitations in its ability to identify the lower level elements. Instead of identifying the data for the lower level element as a separate entity, it assigned the data as properties of the element's higher level parent entities, leading to loss of the hierarchical precision.
While the sample text may not fully capture the complexity of a mission document, it does provide a controlled environment to evaluate the framework's early functionality. It is important to remember the input prompt structure and the generative model's inherent variability are two of the many variables that can affect the final result. This attempt was focused on understanding the behaviour of the framework in its early phase and identifying areas that need further refinement.

5 Conclusion

This work presents the effectiveness of the proposed framework in obtaining a complete set of entity and relationship pairs which are further used for mapping relevant data to OPM models. While providing insightful indications about the framework's alignment with the expectations of entity extraction and data mapping, the results also highlighted several areas for improvement and further exploration in the process of creating knowledge maps (KGs) and their adaptation to the Objective Process Method (OPM) model.

1. One of the challenges identified during the study is the need for advanced PDF text extraction techniques. Although the text extraction process used in this project was effective for a chosen document format, current methods remain limited when creating a general tool for text extraction from different PDFs.
   By incorporating advanced text processing tools such as Document AI [21] into the process , extraction can be greatly simplified by ensuring preservation of text hierarchy and efficient management of different layouts.

2. Prompt engineering [2] is of great importance when it comes to results generated by these generative models. Small variations in the input can cause significant differences in the output, making the change in results difficult to predict and quantify.
   Instead of relying solely on prompt engineering, incorporating fine tuned models trained on OPM model data may yield more accurate results.

3. Currently Knowledge graphs are playing an increasing role in RAG [22] for knowledge intensive tasks, where they are used as a basis for retrieving relevant information from different sources. Recent applications like GraphRAG [23] introduced by Microsoft, reports to enable automatic generation of knowledge graphs from raw text which is later leveraged upon to perform RAG-based tasks.
   RAG- based techniques are effective in managing the dispersion of mission data across multiple documents . With these methods, it would be possible to retrieve relevant parts of the data , which would allow for a better understanding of the entire dataset and improve the efficiency of system extraction and modeling.

4. Further multimodal data analysis , which combines text with other formats such as images or tables , can lead to more comprehensive KGs . Helping to integrate different data formats can result in richer and more nuanced system models , providing greater diversity of information and improving the overall accuracy of knowledge representation.

References

[1] F. Murdaca, A. Berquand, K. Kumar, A. Riccardi, T. Soares, S. Gerené, and N. Brauer. Knowledge-based information extraction from datasheets of space parts. In *Proceedings of the 8th International Systems & Concurrent Engineering for Space Applications Conference (SECESA 2018)*. University of Strathclyde, Glasgow, UK, 2018.

[2] Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. Enhancing knowledge graph construction using large language models, 2023.

[3] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graphs. *ACM Computing Surveys*, 54(4):1–37, July 2021.

[4] Jesús Barrasa and Jim Webber. *Building Knowledge Graphs: A Practitioner's Guide*. O'Reilly Media, 2021.

[5] Amit Singhal. Introducing the knowledge graph: things, not strings. *Google Blog*, 2012. Accessed: 2024-09-06.

[6] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[7] A. Mikheev, M. Moens, and C. Grover. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, 1999.

[8] G. Zhou, J. Su, J. Zhang, and M. Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, 2005.

[9] Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities, 2024.

[10] Roos M. Bakker and Daan L. Di Scala. From text to knowledge graph: Comparing relation extraction methods in a practical context. In *Proceedings of the GeNeSy Workshop, co-located with ESWC 2024*, Hersonissos, Crete, Greece, 2024. CEUR Workshop Proceedings. Accessed: 2024-09-06.

[11] P. Darm, A. Berquand, L. Mansilla, and A. Riccardi. A system engineering recommendation system based on language similarity analysis: An application to space systems conceptual design. In *Proceedings of the 10th International Systems & Concurrent Engineering for Space Applications Conference (SECESA 2022)*, 2022.

[12] A. Berquand et al. Spacetransformers: Language modeling for space systems. In *Proceedings of the 2021 Conference on Language Modeling for Space Systems*, 2021.

[13] OpenAI. Gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[14] LlamaIndex. Mastering pdfs: Extracting sections, headings, paragraphs, and tables with cutting-edge parser, 2024. Accessed: 2024-09-06.

[15] Python Software Foundation. *re - Regular Expression Operations*, 2023.

[16] Dov Dori. *Model-Based Systems Engineering with OPM and SysML*. Springer, New York, NY, 2016.

[17] Ramon Maria Garcia Alarcia and Alessandro Golkar. Architecture of a generative design tool for spacecraft and user front-end implementation through a chatbot smart design assistant. In *Proceedings of the IAC-23, D1, 4A. IAF Space Systems Symposium*, Baku, Azerbaijan, 2023. International Astronautical Congress.

[18] Kaiyan Chang, Songcheng Xu, Chenglong Wang, Yingfeng Luo, Tong Xiao, and Jingbo Zhu. Efficient prompting methods for large language models: A survey, 2024.

[19] Jack Boylan, Shashank Mangla, Dominic Thorn, Demian Gholipour Ghalandari, Parsa Ghaffari, and Chris Hokamp. Kgvalidator: A framework for automatic validation of knowledge graph construction, 2024.

[20] European Space Agency (ESA). Earth explorer 11 candidate mission cairt: Report for assessment. Technical Report ESA-EOPSM-CAIR-RP-4372, European Space Agency, September 2023. Issued on 15/09/2023.

[21] Microsoft. Document intelligence, 2024.

[22] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.

[23] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.