THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Knowledge Graph Ontology Extraction & Schema Alignment

Project Proposal

By Snehin Raj Singh Kukreja

School of Electrical Engineering and Computer Science

The University of Queensland

Submitted for DATA7901 - Data Science Capstone 1

1 May 2025

Contents

Chapter 1 - Introduction

1.1 Objective

Organisations have increasingly depended on structured knowledge to make decisions and streamline operations. This knowledge is often captured through ontologies. Ontologies are representations of concepts, their attributes, and the relationships among them. Many ontologies are created in a process where someone with domain knowledge would define the ontology. Constructing manual ontologies is time-consuming and requires specialized expertise (Noy & McGuinness, 2001).

The objective of this project proposal is to design and develop an automated pipeline for extracting ontologies from electrotechnical datasheets text using a combination of classical natural language processing (NLP) techniques and large language models (LLMs). The aim is to convert unstructured markdown-format documents into structured, machine-readable ontologies that are aligned with external standards like the International Electrotechnical Commission Common Data Dictionary (IEC CDD). The project will focus on reducing human intervention using automated evaluation, refinement, and alignment strategies to improve the scalability and adaptability of ontology extraction.

This project has the potential to reduce the reliance on manual ontology engineering by using both NLP and LLMs. As a result, there is potential for faster knowledge modelling, improved interoperability across systems in an organisation, and enhanced reusability of technical data. The methodology developed can be applied across different industries such as electronics, manufacturing, and energy. The project will contribute to advancing semantic technologies in engineering contexts while reducing time, cost, and domain dependency.

1.2 Scope

The scope of the project includes the design and implementation of preprocessing workflows, entity and ontology extraction using NLP and LLMs. It also includes performance evaluation using both automated metrics and domain expert validation. The project will not involve sensitive data or real-time system integration, but it will focus on developing a reusable, standards-compliant extraction method applicable to technical domains. The dataset will be limited to the domain of electrotechnical documents that are written in English.

With the goals of the project now in mind, the next section discusses previous work done and define key concepts in the field of ontology extraction.

Chapter 2 - Background & Literature Review

2.1 Introduction

As the volume of technical and scientific documentation continues to grow, there is increasing demand for methods that can convert unstructured and semi-structured texts into machine-readable ontologies. This chapter provides an overview of the key foundations, techniques, and challenges in the field of ontology extraction and schema alignment. It begins with foundational concepts, including knowledge graphs, schemas, and representation formats, followed by a historical progression from early rule-based approaches to modern deep learning and large language model (LLM) techniques. I also explore representation strategies, existing ontology standards, schema alignment methods, and the various strategies used to evaluate the quality and utility of extracted ontologies.

2.2 Theoretical Background

Knowledge graphs are structured representations where entities (nodes) are interconnected by relationships (edges). This structure allows for powerful applications like semantic search and recommendation systems which would allow systems to efficiently retrieve and relate information at scale (Vrandečić & Krötzsch, 2014).

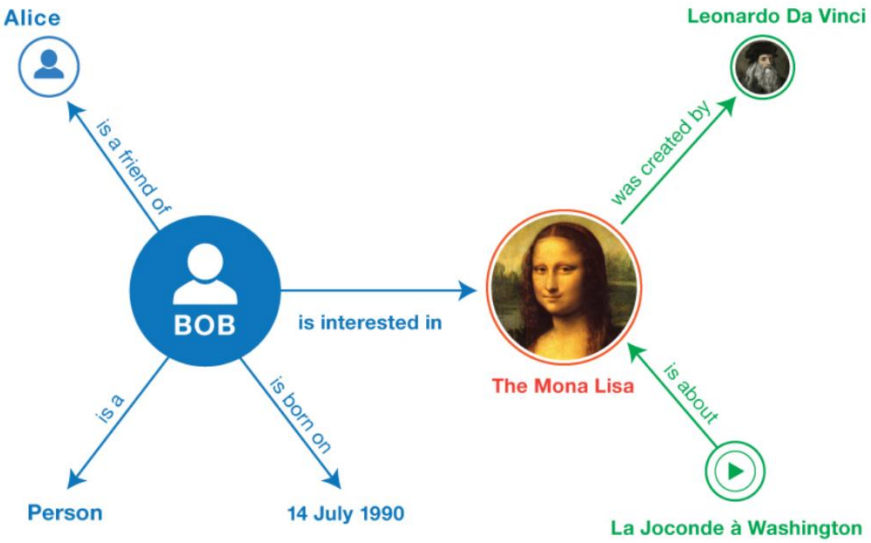

Figure 1: Example of a knowledge graph illustrating relationships between entities using RDF triples. From "RDF 1.1 Primer" (https://www.w3.org/TR/rdf11-primer/), by World Wide Web Consortium (W3C), 2014, CC BY 4.0.

Figure 1 demonstrates the structure of a simple knowledge graph, where entities (e.g., Bob, The Mona Lisa) are connected via labelled relationships ("is interested in", "was created by") forming Resource Description Framework (RDF) triples (W3C, 2014). These graphs are the target output of ontology extraction workflows, where semantic meaning is encoded as machine-readable relationships between concepts.

A schema is a framework that defines the organization of concepts, properties, and relationships within a domain. It's a blueprint that governs how information is structured and interrelated, while supporting consistency, integration, and effective querying of data (Noy & McGuinness, 2001).

Natural language processing (NLP) is a field of artificial intelligence that focuses on understanding, interpreting, and generating human language. It uses tasks such as tokenization, part-of-speech tagging, and named entity recognition, to transform raw text into structured data for further processing (Jurafsky & Martin, 2025).

Large language models (LLMs), including GPT-4 and Claude, are advanced AI systems trained on extensive text datasets. They excel at generating coherent and contextually relevant outputs, significantly pushing the boundaries of automated text analysis and enhancing the overall performance of NLP applications (Brown et al., 2020).

Evaluation or benchmarking is the process of assessing the quality, accuracy and effectiveness of an ontology. Evaluation encompasses structural validation (checking logical consistency and completeness), syntactic verification (compliance with ontology languages like OWL or RDF) and semantic assessment (correctness and relevance of the underlying domain representations). Evaluation can also include measuring how well the ontology supports information retrieval, integration or reasoning tasks.

Ontologies are not universally objective structures but are shaped by their intended purpose. A single source, such as a resistor datasheet, can produce different ontologies depending on the use case. These differences reflect the selective nature of ontology extraction, where the goal is not to reveal a single truth but to construct a model that best serves a specific task.

Ontology Purpose	Focus of Extraction	Example Classes and Properties
Engineering Specification Ontology	Physical and electrical properties for design and classification	Resistor, hasResistance, hasTolerance, hasPowerRating
Maintenance Ontology	Operational behaviour, failure conditions, and lifecycle tracking	Component, hasFailureMode, hasServiceInterval, hasState
Supply Chain Ontology	Manufacturer, packaging, and part number for procurement tracking	Part, hasManufacturer, hasPackageType, hasSKU

Table 1: Possible candidate ontologies generated from an extraction process depending on the use case

Table 1 shows how a single source document, such as a resistor datasheet, can be interpreted differently depending on the target ontology's purpose.

2.3 Ontology Extraction Techniques

2.3.1 Early Techniques

Early ontology extraction methods were mainly rule-based or heuristic. These approaches rely on predefined language patterns and expert-designed rules to identify and extract ontology elements like classes, properties, and relations from text.

Hearst introduced patterns like “X such as Y” (where Y is an example of X) to automatically extract taxonomy relations from text (Hearst, M. A., 1992). These “Hearst patterns” and other hand-crafted rules were effective for bootstrapping taxonomies (e.g. identifying that *“Bambara ndang”* is a kind of *“bow lute”)* (Riloff, E., 2014).

Many early systems also relied on dictionary lookups or regular expressions to find key domain terms and map them to ontology classes. Rule-based approaches benefit from precision (when patterns match, they are often correct) but suffer from low recall and domain portability, meaning that experts must devise new rules for each domain and language.

Statistical methods were introduced to the ontology extraction process. These methods treated ontology extraction as a machine learning or data mining problem: for example, clustering terms that co-occur in similar contexts to propose concept groupings. Association rule mining and other unsupervised algorithms were used to find candidate relations.

Text2Onto (Cimiano & Völker, 2005) is a framework that integrated algorithms like term frequency analysis and clustering to suggest an ontology candidate (Du, Rick et al, 2024). The candidates were ranked by confidence and an expert could curate the results. Other systems like OntoGen and OntoStudio also combined automated suggestions with user feedback to refine ontologies (Du, Rick et al, 2024). This semi-automatic strategy sped up ontology development while keeping a human in the loop.

Many ontology learning approaches have propped up by the end of the 2000s. These approaches often combined linguistic analysis (part-of-speech (POS) tagging and parsing) with statistical scoring to extract structured knowledge from text (Du, Rick et al, 2024). Statistical methods are referred to as shallow learning methods in contrast to the deep learning-based approaches discussed next (Du, Rick et al, 2024).

2.3.2 Deep learning techniques

In the last decade, ontology extraction methods have evolved significantly with the advent of deep learning techniques. Unlike the rule-based and statistical methods discussed earlier, deep learning models can learn complex linguistic patterns and semantic representations from large amounts of data. Deep learning models enhance the performance of ontology extraction.

Neural network architectures, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based architectures, have notably impacted various ontology extraction subtasks including concept extraction, relation extraction, and concept clustering.

One key area where neural networks have improved performance of ontology extraction is concept extraction. Concept extraction is identifying key domain-specific terms from text—has transitioned from reliance on heuristic rules or statistical methods to employing neural sequence labelling models, such as Bidirectional Long Short-Term Memory (Bi-LSTM) networks combined with Conditional Random Fields (CRFs) (Trisedya et al., 2019; Du, Rick et al., 2024).

BiLSTM-CRF models automatically detect multi-word domain concepts and terms in context, significantly outperforming earlier methods in domains such as healthcare, agriculture, and technical literature (Trisedya et al., 2019; Rick, Du et al., 2024).

Neural embedding techniques like Word2Vec, FastText, and BERT (Bidirectional Encoder Representations from Transformers) have transformed ontology extraction by capturing the deep semantic relationships between

words. Embeddings enable automated synonym detection and concept clustering, effectively addressing challenges like synonymy. For example, domain-specific BERT models trained on biomedical or legal corpora performed well at concept groupings as a result identifying subtle semantic relationships that not possible with earlier methods (Rick, Du et al., 2024).

Relation extraction has also seen great strides by using deep learning architectures. CNNs and RNNs perform well in capturing local and sequential context, respectively, enabling accurate identification of relations from structured and semi-structured texts.

Deep learning techniques still have challenges. They require a large and labelled training dataset, which is expensive and scarce. Additionally, neural networks are difficult to interpret due to their “black box” nature, making it hard to validate and trust the extracted ontologies (Rick, Du et al., 2024). Domain adaptation is also difficult, as models trained on one domain’s data (e.g., news articles) can struggle without retraining or fine-tuning when transferred to different domains (e.g. electronic components).

2.3.3 Part-of-Speech Tagging

Ontology extraction heavily relies on identifying meaningful linguistic patterns. Part-of-Speech (POS) tagging is a process where grammatical categories (such as nouns, verbs and adjectives) are assigned to categories (Jurafsky & Martin, 2021). POS tagging improves the ontology extraction process by creating a more systematic extraction of candidate classes, relationships and attributes. From the example in Listing 2, the sentence "The resistor has a tolerance of ±5%", POS tagging identifies "resistor" (noun) as a class candidate and "has" (verb) as a relational indicator. This information forms the basis for extracting ontology triples of the form (Resistor, hasTolerance, ±5%).

The SpaCy library is an open-source Python toolkit that provides efficient and accurate NLP tools. Its ability to process text and perform entity recognition will be useful for the analysis of large volumes of text is required (Explosion AI, 2020).

2.3.4 Large Language Models

Large Language Models (LLMs) like GPT-4, Claude, and Gemini have the potential to revolutionise ontology extraction methodologies. Unlike earlier deep learning approaches that relied heavily on task-specific fine-tuning, LLMs leverage their pre-training on vast and diverse text datasets, enabling general-purpose linguistic understanding and contextual reasoning capabilities (Devlin et al., 2018; Rick, Du et al., 2024).

LLMs have reshaped ontology extraction primarily through their prompt-based, zero-shot, and few-shot learning capabilities. Instead of requiring explicit training for each extraction task, LLMs can use carefullt designed prompts to perform ontology extraction tasks directly. For example, given a simple instruction such as "Extract the entities and their relationships from the following text," models like GPT-4 can readily generate structured triples (subject-relation-object) or conceptual hierarchies without additional task-specific training (Rick, Du et al., 2024).

Furthermore, the advanced semantic reasoning of LLMs allows them to infer implicit relationships from text, addressing limitations discussed earlier in pattern-based or supervised extraction methods. For example, presented with the sentence "Aspirin reduces the risk of heart attack," LLMs can infer a therapeutic or preventive relation ("treats" or "prevents") between "Aspirin" and "heart attack" even if the relations are not explicitly stated (Rick, Du et al., 2024).

Integrating LLMs into the ontology extraction process has led to innovative pipelines combining neural reasoning and structured knowledge representation. One example involved generating competency questions via an LLM, deriving structured ontology schemas from the resulting answers, aligning these schemas with Wikidata, and finally using the same LLM to populate ontology instances. The integrated pipelines demonstrate greater ontology quality, interpretability, and interoperability with existing structured knowledge bases (Chen et al., 2023).

Despite their impressive capabilities, LLMs introduce new challenges due to their generative nature. A big issue is "hallucination," where models generate plausible sounding but factually incorrect information. Addressing this risk requires using reference data or external validation sources to ensure the factual correctness of generated ontologies (Rick, Du et al., 2024). Furthermore, LLMs are computationally resource-intensive, which can limit their applicability to very large-scale ontology extraction tasks without considerable computational resources.

2.3.4 Challenges in Ontology Extraction

There are several challenges in the ontology extraction field. Some of these challenges discussed below are linguistic complexities, data limitations, model interpretability, domain specificity, and the integration of advanced AI techniques.

Linguistic Complexities

Natural language is ambiguous and depends on context. Words and phrases can have multiple meanings, and the same concept can be expressed in various ways which complicates the accurate identification and extraction of concepts and relationships. For example, the word "bank" can refer to a financial institution or the side of a river, depending on context (Xu et al., 2025). Figuring out the meaning of these terms requires a better understanding of their meaning and the situation they're used in.

Data Scarcity and Quality

High-quality, annotated datasets are crucial for training effective ontology extraction models. However, in many specialized domains, such datasets are scarce or non-existent. Data scarcity hampers the development of accurate models and limits their applicability. Moreover, existing datasets may contain noise, inconsistencies, or outdated information, further challenging the extraction process (Mai et al., 2024).

Model Interpretability and Validation

Deep learning models, while powerful, often operate as "black boxes," making it difficult to interpret their decisions and validate the extracted ontologies. This lack of transparency poses challenges in assessing the reliability and accuracy of the extracted knowledge structures. Ensuring that the ontologies align with domain expertise and real-world semantics requires additional validation mechanisms (Singh et al., 2024).

Domain Adaptability and Transferability

Models trained on data from one domain may not perform well when applied to another due to differences in terminology, structure, and context. Adapting ontology extraction models to new domains often necessitates retraining or fine-tuning with domain-specific data, which may not be readily available. This limitation restricts the scalability and generalizability of ontology extraction approaches (Mai et al., 2024).

Evaluation Metrics and Standards

Assessing the quality and completeness of extracted ontologies is complex. There are a lack of standardised evaluation metrics and benchmarks, making it difficult to compare different extraction methods objectively. Developing comprehensive evaluation frameworks is essential to advance the field and ensure the practical utility of extracted ontologies (Xu et al., 2025). Some evaluation techniques are discussed later in this chapter.

2.3.5 Representing Ontologies

File-Based Formats

One common approach to represent ontologies is to store them in file-based formats. The following file-based are widely used:

RDF/XML:

RDF/XML is a W3C standard that encodes ontologies as a collection of Resource Description Framework (RDF) triples (subject, predicate, object). It is expressive and is well supported by W3C's Semantic Web tools (Gruber, 1993). However, RDF/XML's XML-based syntax is verbose, making the files harder to read and modify manually.

Turtle:

Turtle offers a more compact and human-friendly syntax compared to RDF/XML. Turtle's readability makes it a good choice during prototyping and development and is great for manual editing. However, Turtle does not support some of the more advanced features, such as representing collections or containers, as explicitly as RDF/XML (Hogan et al., 2021).

JSON-LD:

JSON-LD is a JSON-based format designed for Linked Data. It is developer-friendly due to its adoption of conventional JSON structures and is well-suited for web applications. However, JSON-LD depends on the "@context" for semantic mappings. This may lead to occasional configuration challenges, especially when integrating with legacy systems (Hogan et al., 2021).

OWL/XML:

OWL stands for the Web Ontology Language. It is a formal language designed to represent rich and complex knowledge about a domain. OWL builds on the Resource Description Framework (RDF) and RDF Schema (RDFS)

by providing additional vocabulary and semantics that support advanced reasoning tasks.

OWL was developed and standardised by the World Wide Web Consortium (W3C), OWL enables the creation of explicit ontologies that describe classes, properties, and individuals (instances) along with the relationships between them (Gruber, 1993). This makes it possible to infer implicit knowledge, verify the consistency of the ontology, and enable interoperability across different systems.

This format is specifically tailored to represent OWL ontologies, ensuring that OWL's rich semantics are accurately captured. While OWL/XML supports advanced constructs, its rigid XML syntax makes it less approachable for human curation and quick modifications.

N-Triples:

N-Triples is a simple, line-based format that represents each RDF triple on a separate line. Its simplicity means it's easy to debug and bulk process data. However, it is not ideal for manual editing or for representing complex ontological relationships (Hogan et al., 2021).

Database Storage

For large-scale applications or when complex queries and reasoning are required, file-based storage is slower compared to database systems. The following types of database storage options are available:

Triple Stores:

Triple stores, such as Apache Jena Fuseki, GraphDB, or Stardog, are designed to store RDF triples and support SPARQL queries. These systems are optimized for handling vast amounts of semantic data and perform automated reasoning using RDF Schema (RDFS) or OWL entailment regimes (Wilkinson et al., 2016).

Graph Databases:

Graph databases are a specialised database management system (DBMS) that uses graph theory to store, represent, and query data. One example of a graph database is Neo4j (Neo4j, 2012). In Neo4j, entities (ontology classes) are stored as nodes and relationships (e.g., "is-a" or "related-to") as edges. This model allows users to efficiently graph traversals and provides visualization tools, which are especially beneficial during iterative ontology refinement (Neo4j, 2012). However, because Neo4j uses a property graph model rather than RDF, additional conversion steps will be needed if interoperability with standard RDF or OWL systems is required (Hogan et al., 2021).

Comparison of Formats

Interoperability:

RDF/XML and OWL/XML are highly standardized and promote interoperability, but they are less human-readable.

Readability and Ease of Editing:

Turtle and JSON-LD offer better readability and ease of manual editing, which can be essential during the development phase.

Query Efficiency and Reasoning:

Triple stores and graph databases provide robust querying and reasoning capabilities necessary for large knowledge graphs.

Conversion Overhead:

Graph databases like Neo4j offer significant advantages in terms of traversal and visualization, the property graph model may require additional processing for compatibility with RDF-centric systems.

For reasoning and interoperability with Semantic Web tools, RDF/XML will be a preferable solution. However, if editing and integration with web applications is a priority, Turtle or JSON-LD could be more suitable. Projects that require dynamic querying and visualization might benefit most from storing ontologies in a graph database like Neo4j, even if that means handling format conversion for certain applications.

Now that we have discussed how ontologies can be represented for machines and people to view, the next chapter looks at one existing ontology standard in the domain of electrotechnical documents.

2.4 Existing Ontology Standards

The International Electrotechnical Commission (IEC) has standards for ontologies that describe electrical components. The relevant standard in this context is the IEC 61360. IEC 61360 defines a general-purpose vocabulary through a reference dictionary for electrotechnology and related domains. It provides standardised data element types and associated classification schemes, that enables consistent data representation and exchange. The standard is structured into multiple parts, including:

- IEC 61360-1: Outlines principles and methods for defining properties and associated attributes (IEC, 2017).
- IEC 61360-4: Presents the IEC Common Data Dictionary (IEC CDD), an online repository of standardised terms (IEC, 2005).

IEC Property ID	**Preferred Name (English)**	**Description**
MDC_P001_6	Nominal Resistance	The resistance value of the resistor under standard conditions.
MDC_P002_1	Tolerance	The permissible variation from the nominal resistance, expressed as a percentage.
MDC_P002_2	Rated Power	The maximum power the resistor can dissipate without damage under specified conditions.
MDC_P004_1	Maximum Operating Temperature	The highest temperature at which the resistor can operate continuously without degradation.
MDC_P007_1	Rated Voltage	The maximum continuous voltage that can be applied to the resistor without exceeding its rated power.

Table 2. Example Properties from the IEC Common Data Dictionary for Resistors (IEC, 2005)

The IEC CDD serves as a centralised resource for standardised product descriptions. Table 2 above shows an example of some of the properties of a low-power resistor. The next section discusses how to compare a candidate ontology to standards like IEC-61360 and align it accordingly.

2.5 Schema Alignment Methods

Schema alignment is the process of finding connections between related entities across different schemas or ontologies (Du, Rick et. al, 2024). Schema alignment is important for the interoperability and knowledge management in knowledge systems across organisations.

Approaches to schema alignment can be categorised into several types:

- **Structural Approaches**: These focus on the relationships and hierarchies within schemas, analysing the context of elements to determine alignments(Shvaiko & Euzenat, 2005).
- **Hybrid Approaches**: Combining different methods with the aim to improve alignment accuracy by leveraging multiple sources of information (Rahm & Bernstein, 2001).
- **Machine Learning Approaches**: Use of supervised and unsupervised learning models, to predict correspondences based on training data. These approaches can adapt to complex and evolving schemas (Doan et al., 2003).
- **Embedding-Based Approaches**: These methods represent schema elements as vectors in a continuous space, capturing semantic similarities through distances in the vector space (Portisch et al., 2022).

Ontology alignment and schema matching have also significantly benefited from LLM capabilities. Recent studies have demonstrated that LLMs, given structured prompts, can effectively propose mappings between concepts across ontologies by assessing the conceptual similarity based on provided definitions or contextual descriptions (Shimizu & Hitzler, 2024).

Once an ontology has been aligned, it is ready to be evaluated to see if it accurately captures the knowledge of the dataset. The next sub-sections discusses evaluation strategies.

2.6 Evaluation Strategies

Evaluating automatically extracted or aligned ontologies is essential to ensure their quality, usability, and correctness. Various methodologies have been developed, each assessing different facets of an ontology's performance and structure. The primary evaluation strategies discussed below are the gold standard-based evaluation, application-based evaluation, data-driven evaluation, and hybrid or automated tool-based approaches.

Gold Standard-Based Evaluation involves comparing the generated ontology against a pre-existing, expert-validated ontology, often referred to as the "gold standard." This comparison uses metrics such as precision, recall, and F1-score to quantify the accuracy of the ontology's elements.

Ponzetto and Strube (2007) evaluated ontologies extracted from Wikipedia by comparing them to WordNet (Princeton University, 2010), measuring the correctness of subclass relationships recovered. This method provides objective and repeatable results when a suitable reference ontology is available. However, its applicability is limited in domains that don't have gold standards.

Application-Based Evaluation assesses an ontology based on its performance within a specific application or task. This approach measures how effectively the ontology enhances the application's functionality. Porzel and Malaka (2004) demonstrated this by integrating an ontology into a speech recognition system and evaluating improvements in semantic interpretation accuracy. While this method offers insights into practical utility, its results are often task-specific and may not generalize across different applications.

Data-Driven Evaluation uses a representative corpus to determine how well the ontology reflects the domain-specific content. This approach examines corpus coverage (e.g. how many concepts and relationships present in the corpus are captured by the ontology) and the semantic alignment between the ontology and the corpus. Brewster et al. (2004) proposed a probabilistic method to evaluate the structural fit between an ontology and a domain-specific corpus, emphasizing the importance of aligning ontology structures with real-world data. This strategy is beneficial when gold standards are unavailable but may be influenced by corpus bias or incompleteness.

Hybrid and Automated Tool-Based Approaches combine elements from the previous methods and incorporate automated tools to enhance evaluation efficiency. Tools like OOPS! (OntOlogy Pitfall Scanner!) automatically detect common pitfalls in ontologies, such as missing domain or range definitions and class cycles, providing a well-structured assessment of potential issues (Poveda-Villalón et al., 2014).

Strategy	Tests	Best Use Case
Gold Standard	Accuracy via precision/recall	Controlled, benchmark-based comparisons
Application-Based	Functional utility in end-use	Practical system integration
Data-Driven	Semantic alignment with real corpora	When there is no gold standard available
Hybrid/Tool-Based	Automated checks and multi-angle evaluation	Development-time quality control

Table 3: Summary of different evaluation strategies

Each evaluation strategy offers different insights, and using a combination of these strategies provides a well-rounded assessment of ontology quality and alignment accuracy. A robust evaluation will combine the methods discussed. For example, using a gold standard for class structure, data-driven analysis for coverage, and human review for semantic soundness.

2.7 Conclusion

In this literature review, I provided a background into the field of ontology extraction. Afterwards, I examined the evolution of ontology extraction from traditional rule-based approaches to deep learning and new large language model (LLM) techniques. While there have been big strides in the automation and scalability of ontology development, there are challenges, particularly in handling linguistic ambiguity, ensuring interoperability, and evaluating the quality of generated ontologies. Hybrid models that combine LLMs with neural networks shows potential to address these issues. However, the field still lacks standardised evaluation frameworks and comprehensive benchmarks to assess the performance of these methods. There is a gap in the development of robust evaluation metrics and exploring the applicability of emerging technologies to create more accurate and interoperable ontological structures. In the next section, I will utilise the ideas discussed to design and implement an automated ontology extraction process.

Chapter 3 - Methodology

3.1 Introduction

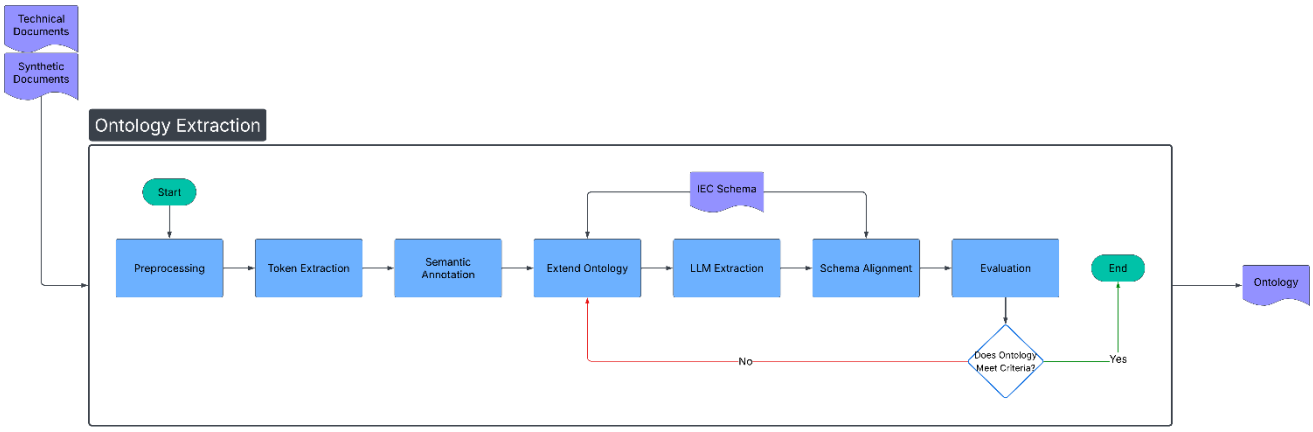

Figure 2: Proposed Ontology Extraction Process

Figure 2 shows an overview of the ontology extraction process that is detailed in this chapter. The data will be pre-processed to remove unnecessary text, then tokens (fundamental units of text) will be extracted and annotated. An initial ontology will be created and passed into an LLM to enhance the quality. Afterwards the candidate ontology will be evaluated. The next section discusses how the dataset will be prepared and how it will be augmented with synthetic data.

3.2 Data Preparation and Synthetic Data Generation

3.2.1 Data Sources and Metadata

This project will use both real-world technical documents and synthetically generated content to support the development and evaluation of ontology extraction workflows. The combined dataset reflects the semantic complexity and variability characteristics of electronic documentation.

Primary Data Sources

The full dataset has not been delivered at the time of proposal; the expected characteristics of the corpus will be inferred from the synthetic samples and domain standards. The documents will have technical writing patterns, including structured headings (e.g., "Specifications," "Features," "Applications"), tables of numeric values, and descriptive paragraphs relating components and their properties. A synthetic example, illustrating the expected structure and content style, is provided in the Appendix, Listing 3.

The primary dataset will compromise of technical texts that have been pre-processed into markdown format to enable consistent parsing and extraction. These include:

- **Scientific Texts**: Peer-reviewed research articles and technical papers in the fields of electronics, control systems, and embedded systems engineering. These documents are sourced from open-access repositories and offer structured domain knowledge that is ideal for extracting entities, relationships, and constraints.
- **Electrical Component Datasheets**: Manufacturer datasheets for passive and active components (e.g., resistors, capacitors, sensors, microcontrollers). These documents contain highly structured technical specifications, often presented in tables or diagrams, and are rich in semantic cues relevant for concept extraction.
- **Business and Technical Reports**: Industry publications such as system architectures, compliance requirements, and product integration strategies. These texts provide insights into component roles and a bigger picture view that is not seen in the other two sources.

The data will have the following metadata:

- Document Source Type (e.g., Scientific Paper, Datasheet, Report)
- Document Title or ID (if available)
- Document Length (e.g., word count, token count)
- Primary Language (expected: English)
- Domain Tags (e.g., Resistors, Sensors, Power Electronics, Control Systems)

These metadata elements will assist in categorising documents, segmenting workflows, and evaluating extraction performance across different source types.

All documents will be received initially as markdown text files. The Markdown format retains linguistic content while discarding layout noise (e.g., figures, watermarks, page headers/footers), simplifying the tokenisation and annotation stages later.

Synthetic Data Generation

To supplement the dataset, I will generate synthetic technical using ChatGPT (OpenAI, 2023). Prompt-engineered templates are used to simulate datasheet-style entries, structured component descriptions, and ontology-relevant sentence structures. The synthetic data serves three primary roles:

- Addressing coverage gaps in underrepresented classes or relationships

- Providing training and evaluation material for LLM-based extractors
- Supporting controlled experiments when prototyping schema alignment and semantic annotation

All synthetic outputs are also rendered in markdown format to maintain consistency across the datasets. One example of a prompt to generate a data sheet would be “Please generate a complex technical document, for example a datasheet for a non-existent electronic component, in markdown format. Include tables, headings and detailed technical information” the result can be seen in Listing 3.

Data Type	**Source**	**Format**	**Purpose**
Scientific Texts	arXiv, IEEE, open-access archives	Markdown (from PDF)	Extract terminology, definitions, and conceptual structure
Electrical Datasheets	Manufacturer websites (e.g., TI, ST)	Markdown (from PDF)	Extract specifications, component properties, roles
Business/Technical Reports	Industry publications, whitepapers	Markdown (from PDF)	Model system-level relationships and constraints
Synthetic Technical Descriptions	Generated via LLM prompts	Markdown	Augment dataset for coverage and evaluation consistency

Table 1: Summary of the sources of data used for this project

Table 4 shows a summary of the types of data being used to extract ontologies. This data will first be pre-processed to remove unnecessary data. The next sub-section explains the pre-processing stage further.

3.2.2 Data Preprocessing

To prepare for ontology extraction, the provided markdown documents will go through additional preprocessing to enhance the structured analysis. The preprocessing phase will focus on standardising the corpus, removing irrelevant content, and preparing the text for natural language processing (NLP) workflows.

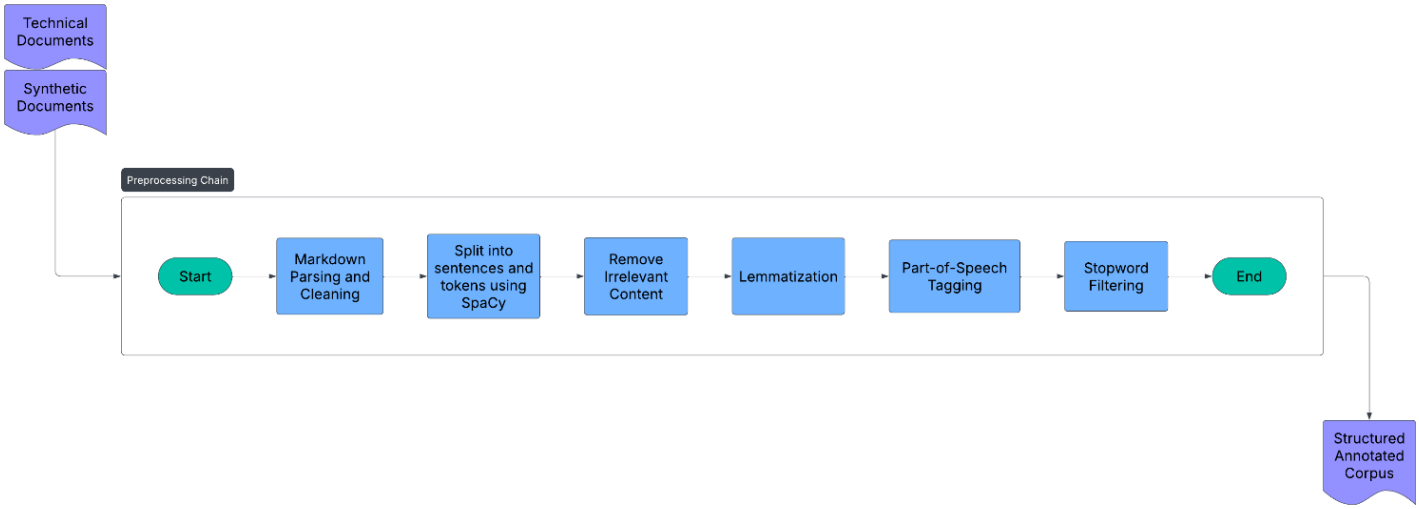

Figure 3: Planned preprocessing workflow for preparing the corpus dataset

Figure 3 shows the planned preprocessing workflow to transform the markdown documents into a structured format that will better aid the ontology extraction process. The following steps are planned:

1. **Markdown Parsing and Cleaning**

   The documents will be parsed to isolate the main textual content. Formatting artefacts such as headings, lists, and emphasis markers will be normalised, while non-informative sections (e.g., footers, watermarks) will be removed. Specific attention will be given to preserving semantically meaningful sections such as specifications, feature descriptions, and operational parameters.

2. **Sentence Segmentation and Tokenisation**

   The cleaned text will be segmented into sentences and tokenised into individual words and phrases using the SpaCy NLP library. This step will enable finer-grained analysis, laying the groundwork for subsequent entity and relationship extraction.

3. **Removal of Irrelevant Content**

   Any sections unrelated to component properties or technical descriptions—such as marketing language, legal disclaimers, or company profiles—will be filtered out using pattern-based rules and keyword heuristics.

4. **Lemmatisation**

   Lemmatisation is the process of determining the lemma of a word based on its intended meaning. This process will standardise variations in words (e.g., "measures" is transformed to "measure"), minimising vocabulary fragmentation and improving matching with domain ontologies.

5. **Part-of-Speech Tagging**

   Tokens will be annotated with their grammatical categories (e.g., noun, verb, adjective) using SpaCy's POS tagging functionality. This information will assist in identifying candidate classes, relationships, and attribute-value pairs for ontology construction.

6. **Stopword Filtering**

   English stopwords are common words that are usually ignored in text analysis and search engine indexing because they don't contribute significantly to the meaning of a text. These words include articles ('a', 'an', 'the'), prepositions ('of', 'in', 'for', 'through'), pronouns ('it', 'their', 'his'), and auxiliary verbs. Common English stop words will be removed where they do not contribute semantically to concept identification.

To guide the design of the preprocessing pipeline, preliminary synthetic examples, such as the one presented in Listing 3, have been analysed. Once the data has been preprocessed, we can move on to the next stage of extracting a candidate ontology from the data as discussed in the next sub-section.

3.3 Ontology Extraction Workflow

The objective of this workflow is to extract structured ontological representations of classes, properties, and relationships from the pre-processed technical documents. The planned workflow integrates NLP techniques with large language model (LLM)-based augmentation to maximise coverage and semantic accuracy.

3.3.1 Token Extraction

After preprocessing the dataset, candidate tokens will be extracted based on part-of-speech (POS) annotations. Nouns and noun phrases (e.g. Resistance) will be targeted as potential classes or entities, while numerical expressions (e.g. 50 Ohms) will be flagged as candidate attribute values. Tokens associated with the domain of electronics (e.g., "resistance," "voltage," "tolerance") will be prioritised for further semantic annotation.

3.3.2 Semantic Annotation

Extracted tokens will be annotated based on linguistic patterns and context. This stage will involve:

- Mapping identified entities to semantic types (e.g., Component, Property, Measurement).
- Recognising syntactic structures that indicate relationships, such as subject-verb-object triplets (e.g., "Resistor has Tolerance").
- Using rule-based templates to generate initial candidate triples (Class–Property–Value structures).

Where available, external references such as the IEC Common Data Dictionary (IEC CDD) will guide annotation, ensuring alignment with standardised electrotechnical concepts.

3.3.3 Extending Existing Ontologies

The extracted concepts and properties will be compared against the IEC CDD (IEC, 2005). If exact matches are found, extracted terms will be aligned; otherwise, new classes or properties will be proposed. This hybrid approach will support both ontology enrichment (introducing new concepts) and ontology alignment (linking to known standards).

3.3.4 LLM-Based Extractors

ChatGPT will be used to enhance extraction quality, particularly in handling complex, descriptive paragraphs where rule-based methods may underperform. LLMs will be prompted via OpenAI's API (OpenAI, 2023) to generate ontology triples directly from sentences or paragraphs, constrained by predefined templates to minimise hallucination.

3.3.5 Schema Alignment

After the extraction of ontology classes, properties, and relationships, there will be a schema alignment step to map the extracted ontology elements to the main alignment target, the IEC Common Data Dictionary (IEC CDD).

Alignment will proceed through a combination of automated and semi-automated methods:

- **Lexical Matching**: The extracted terms will be compared to standard ontology terms using string similarity measures (e.g., Levenshtein distance, token-based matching) and embedding-based similarity scoring where appropriate.
- **Semantic Type Checking**: The extracted entities will be compared against the constraints defined in target schemas (e.g., ensuring that properties like "Rated Voltage" align with component classes like "Resistor" in IEC CDD).
- **Manual Verification**: For cases where automated alignment produces ambiguous results, manual review will be conducted to confirm the correct mapping or to flag the extracted entity as a novel concept requiring ontology extension.

Through this schema alignment step, the extracted ontology will be grounded in recognised domain standards, ensuring semantic interoperability, reusability, and validation readiness.

3.4 Evaluation and Validation

3.4.1 Performance Metrics

To evaluate the effectiveness of the ontology extraction and schema alignment processes, a combination of standard information retrieval metrics and ontology quality metrics will be used. These metrics will assess both the syntactic accuracy (correctness of extracted triples) and the semantic quality (faithfulness to domain knowledge) of the outputs.

The primary performance metrics planned for use are:

1. Precision

Precision measures the proportion of extracted ontology elements (e.g., classes, properties, relationships) that are correct with respect to a gold standard or reference ontology.

$$Precision = \frac{True\ Positives + False\ Positives}{True\ Positives}$$

A high precision score will indicate that the extraction system produces few irrelevant or incorrect concepts.

2. Recall

Recall assesses the proportion of relevant ontology elements that were successfully extracted from the corpus.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

A high recall score will demonstrate the system's ability to comprehensively capture domain knowledge.

3. F1 Score

The F1 score, the harmonic mean of precision and recall, will be used as a balanced measure to account for both the accuracy and completeness of extraction.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

This metric is useful when there is a trade-off between precision and recall and can assess how the ontology performs in optimising both.

4. Ontology Coverage

Coverage will be measured by comparing the number of extracted concepts and properties to the expected concepts defined by reference ontologies such as the IEC Common Data Dictionary (IEC, 2005). Coverage evaluation will assess how much of the domain-specific knowledge space has been captured. For example, if the ontology of a resistor represents key ideas such as resistance or wattage.

5. Alignment Accuracy

For schema alignment tasks, the alignment accuracy will be evaluated by comparing the extracted schema mappings to manually verified mappings. The metric will assess the correctness of linking extracted concepts to existing ontological classes (e.g., IEC CDD classes).

If the ontology performs well during the evaluation stage, the candidate ontology will be delivered to the users. However, should the ontology perform poorly during the evaluation stage, the design may need to refine and go through the extraction process again. The next section discusses how this iterative process will work.

3.4.3 Iterative Ontology Refinement

This project will follow an automated, feedback-driven process, with two different strategies to incorporate iterative feedback:

1. The first is at a corpus-level, where the entire dataset is processed to generate an initial ontology. This global view enables a broad mapping of concepts and relationships, after which low-confidence elements can be revised.
2. The second strategy is to incrementally add documents, where the ontology is built progressively by processing one document at a time. Each document contributes new concepts or refines existing ones, allowing the system to update class hierarchies and property structures with each additional document.

Both approaches will be evaluated during development and may be combined depending on performance and scalability. The next section will discuss the resources that will be used to develop the process chain.

3.5 Resources

3.5.1 Tools Used

A range of open-source libraries and platforms will be used to implement the ontology extraction workflow:

Python: Python will be primary programming language for this project due to its large ecosystem for natural language processing, machine learning, and knowledge graph libraries. Python enables efficient scripting for preprocessing, integration of external APIs, and structured data manipulation.

SpaCy: SpaCy will be used to split markdown texts into sentences and tokens, assign grammatical categories to words, and identify candidate entities and relationships. SpaCy's optimised pipeline and linguistic accuracy make it ideal for large-scale NLP workflows (Explosion AI, 2020).

OpenAI API (ChatGPT): The OpenAI API will be used to generate synthetic technical data to supplement the corpus (OpenAI, 2023). The API will also be used to implement the LLM extraction process.

Protégé: Protégé will be used for the visual inspection and editing of OWL ontologies. It also supports logical validation using built-in reasoners (e.g., HermiT), which will allow me to assess and refine the ontology structure.

OWL / RDF / Turtle: The ontology will be represented in OWL using RDF-compatible serialisations such as Turtle or RDF/XML. Turtle will be used during development due to its readability, while RDF/XML may be used for compatibility with alignment tools and validators.

Neo4j: Neo4j will be explored as a possible backend for storing and querying extracted ontology triples. It supports RDF-like querying via Cypher and is useful for visualising relationships in graph form, especially during debugging and refinement stages.

OWLReady2: OWLReady2 is a Python library that will be used to create and manipulate OWL ontologies. It supports reasoning, ontology loading/saving, and alignment with other RDF sources.

HermiT / Pellet: OWL reasoners will be used to check the logical consistency of the generated ontologies. They will be integrated into the workflow via Protégé or programmatically using OWLAPI.

Pandas: Pandas will be used for general data manipulation tasks, such as formatting extracted triples, analysing token-level statistics, and generating structured output for evaluation.

Matplotlib / Seaborn: These libraries will support visualising evaluation metrics, token distributions, and ontology structure diagrams during the reporting phase.

The selected tools and libraries provide a robust foundation for implementing the ontology extraction pipeline, balancing open-source efficiency with advanced capabilities offered by commercial APIs. While many components like SpaCy, Protégé, and OWLReady2 are free to use, the OpenAI API will incur variable costs depending on usage volume, which will be discussed in the next section.

3.5.2 Costs

Most tools used in this project are open source and have no licensing fees. Similarly, code development, preprocessing, and ontology structuring will be performed locally on personal or university-provided computing resources like Rangpur or UQ Zones.

The main cost will be the use of the OpenAI API to generate synthetic technical data and to support natural language-driven ontology extraction using large language models. I expect to use the ChatGPT 4o model for complex concept generation and prompt-based ontology extraction. As of April 2025, OpenAI charges approximately $2.50 per 1,000 input tokens and $10.00 per 1,000 output tokens for GPT-4o (OpenAI, 2024). While GPT-3.5 is a cheaper alternative ($0.002 per 1,000 tokens), I prefer to use GPT-4o for its ability to reason over technical content more accurately and therefore provide better quality responses.

For the cost projections, I assume that approximately 100 documents (real and synthetic) will be processed. Each requiring 3 prompt interactions (e.g., one for class extraction, one for property-value extraction, and one for triple generation). For a baseline, each prompt will use an estimated 1,200 tokens, split equally between input and output. This results in an approximate token volume of 360,000 tokens (180,000 input + 180,000 output).

Component	Details	Estimated Cost (AUD)
OpenAI API (GPT-4o) – Input Tokens	180,000 tokens × $2.50 / 1,000	$450.00
OpenAI API (GPT-4o) – Output Tokens	180,000 tokens × $10.00 / 1,000	$1,800.00
Other Tools (SpaCy, Protégé, OWLReady2)	Open-source	$0
Infrastructure (Local or University VM)	Covered by university/student resources	$0
Total Estimated Cost		**$2,250.00**

Table 5: Cost breakdown for the project

Table 5 summarizes the cost breakdown for the project with a total estimated cost of 2 250 Australian dollars. The next section discusses the project timeline and the estimated time taken to complete milestones.

3.5.3 Project Timeline

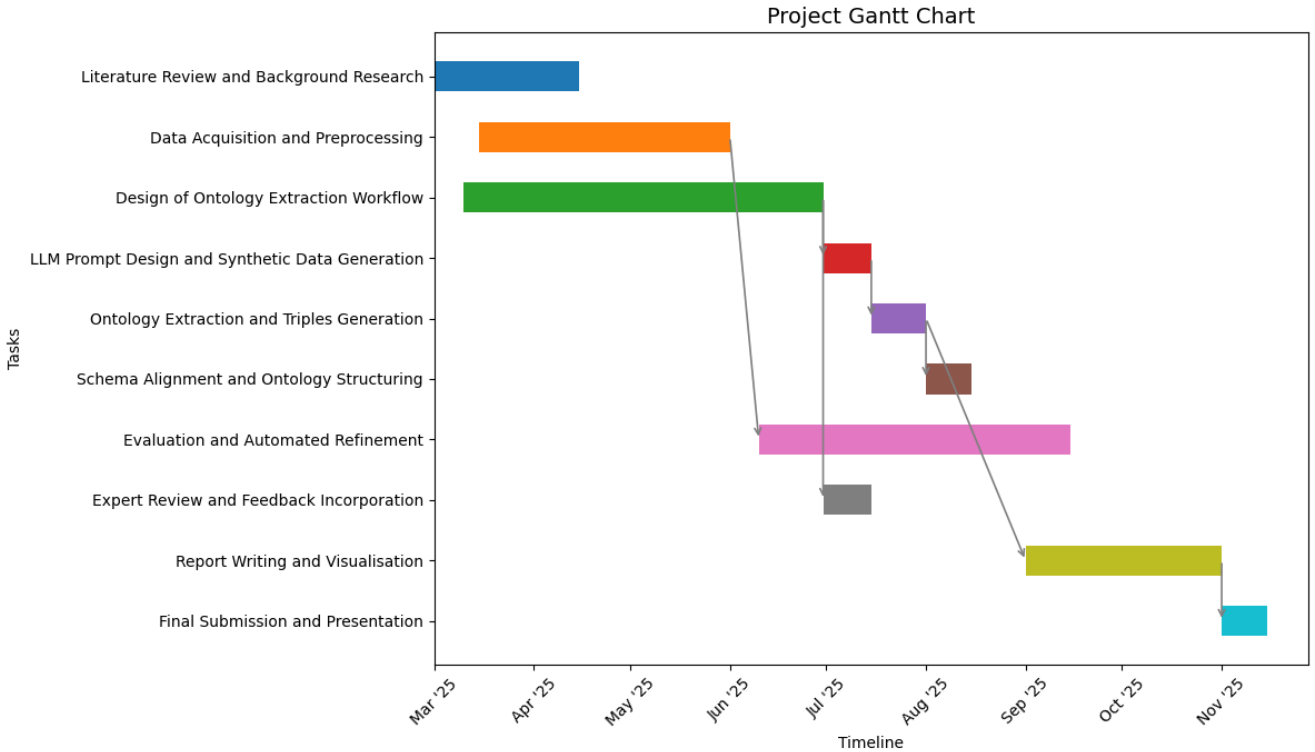

Figure 4: Gantt Chart showing the breakdown of tasks and time estimates

Figure 4 shows a breakdown of objectives and runs from March to November 2025 and follows a phased approach, beginning with background research and workflow design, followed by extraction, alignment, and evaluation. The next section discusses my qualifications and skills that will enable me to complete this project on time.

3.5.4 Research and experience qualifications

I bring a multidisciplinary background that will aid with both the technical implementation and domain knowledge expertise. As a Master of Data Science student, I have developed strong foundational skills in data analysis, machine learning, deep learning techniques and designing distributed systems to design scalable systems.

I hold a Bachelor's degree in Electrical and Computer Engineering, where I gained knowledge in electrical and electronic components and their datasheets which provides a background to the necessary domain expertise relevant to this project.

I am proficient in Python, SQL, C/C++ and Java. I have studied and tutored courses in data structures and database design which will aid me for the graph theory and schema analysis required in this ontology extraction project.

With these skills, I will be able to complete this project and meet all the objectives of this proposal. The next section considers any ethics and privacy concerns related to the project.

3.6 Ethics and Privacy Considerations

This project does not involve any private, confidential, or personally identifiable information. All documents used for ontology extraction (technical datasheets, academic articles, and synthetic markdown content) are publicly available or generated for research purposes. The data sources are either open access or produced using large language models without reference to any personal or sensitive inputs.

As a result, the project poses no ethical risk and does not require human participant involvement or ethical review under the university's research guidelines. All synthetic data generated via the OpenAI API will be used in accordance with OpenAI's terms of service and will not include or infer personal data. Outputs will be evaluated solely for semantic and structural correctness.

3.7 Conclusion

This chapter outlined the methodology for developing an automated ontology extraction and schema alignment pipeline using natural language processing and large language models as per the objective of the project. The project combines classical NLP techniques with LLM-based extraction. Data will be drawn from publicly available technical sources and enhanced with synthetic inputs, ensuring domain relevance and ethical compliance.

The proposed workflow includes preprocessing, entity and relation extraction, ontology construction, and alignment to standards such as the IEC CDD. Evaluation will be performed through both automated metrics and consistency checks, with minimal reliance on manual intervention. Tools such as SpaCy, Protégé, Neo4j, and the OpenAI API will form the technological stack to implement the project.

By focusing on automation, alignment accuracy, and iterative refinement, the project aims to produce a scalable solution for knowledge graph construction and set the foundation for broader applications in engineering and industrial documentation.

The next section discusses what sort of results are expected from the process chain and how to evaluate and benchmark the performance of candidate ontologies.

Chapter 4 – Expected Results

The primary expected outcome of this project is a machine-readable ontology that accurately represents knowledge of electronic components extracted from unstructured markdown documents. The resulting ontology will include domain-specific classes, properties, and relationships aligned to established external standards (e.g. IEC 61360). Ontologies will be expressed in OWL and serialised in Turtle for readability and RDF/XML for compatibility.

In addition to the ontology artefact, the project will generate performance metrics that evaluate the quality of the extraction process. These include precision, recall, F1 score and alignment accuracy. I anticipate that automated refinement and feedback loops will improve these metrics over successive iterations, with minimal manual correction required.

Evaluation will be conducted against reference data where available, and semantic alignment with external ontologies such as the IEC CDD will be used as an additional benchmark (IEC, 2005). The ontology is expected to capture core electrotechnical concepts such as component types (e.g., Resistor, Capacitor), properties (e.g., Rated Resistance, Tolerance), and relational structures (e.g., hasProperty, isSubtypeOf, alignedTo).

4.1 Sample Ontology Fragment (Turtle Syntax)

Below is an example of part of the expected ontology output in Turtle format:

```
@prefix ex: <http://example.org/ontology#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

ex:Resistor a owl:Class ;
    rdfs:label "Resistor" ;
    rdfs:subClassOf ex:PassiveComponent .

ex:hasResistance a owl:DatatypeProperty ;
    rdfs:domain ex:Resistor ;
    rdfs:range xsd:float ;
    rdfs:label "Rated Resistance" .

ex:Resistor_10k a ex:Resistor ;
    ex:hasResistance "10000.0"^^xsd:float ;
    rdfs:label "10kΩ Resistor" .
```

Listing 1: An example of an ontology defining a resistor

Listing *1* defines a class Resistor, a property hasResistance, and a specific instance Resistor_10k with a defined value. Similar structures are expected to be produced for other component types and their associated specifications.
Once an accurate ontology has been extracted from the dataset, that captures a hierarchy model of electronic components in the corpus, the process is complete and the project will have met it's primary objective. The next section provides a conclusion of the project and potential futureworks

Chapter 5 - Conclusion

This project proposes a hybrid methodology for ontology extraction and schema alignment using both classical NLP and LLMS. The approach combines traditional techniques such as tokenisation and part-of-speech tagging with prompt-driven LLM-based extraction and the generation of synthetic data to improve coverage and flexibility.

The objective is to convert unstructured technical documents, like as engineering datasheets, into structured, machine-readable ontologies aligned with standards like the IEC Common Data Dictionary (IEC CDD) (IEC, 2005).

A potentially innovating aspect is the automation of the iterative refinement process. By incorporating confidence-based filtering, logical validation, and schema alignment feedback loops, the system is designed to minimise human-in-the-loop intervention while maintaining semantic accuracy. Tools such as SpaCy, Protégé, and the OpenAI API are used to enable a scalable and efficient implementation.

The use of synthetic data and large language models allows the methodology to generalise across varied document structures and domains, thereby reducing the dependency on domain-specific annotated datasets. This has practical business value in industries that manage large volumes of technical documentation, offering opportunities for automation, interoperability of systems, and reduced manual overhead.

Future work could explore extending the framework to process multilingual text, integrate active learning tuning prompts given to LLM based extractors, or applying the approach to other domains such as biomedical devices or vehicle parts.

Bibliography

Brank, J., Grobelnik, M., & Mladenić, D. (2005). A survey of ontology evaluation techniques. *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*.

Brewster, C., Alani, H., Dasmahapatra, S., & Wilks, Y. (2004). Data driven ontology evaluation. *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, 79–82. https://aclanthology.org/L04-1476

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805. http://arxiv.org/abs/1810.04805

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2003). Learning to map between ontologies on the semantic web. *Proceedings of the 11th International Conference on World Wide Web*, 662–673. https://doi.org/10.1145/775152.775228

Euzenat, J., & Shvaiko, P. (2013). *Ontology matching* (2nd ed.). Springer. https://doi.org/10.1007/978-3-642-38721-0

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Hlomani, H., & Stacey, D. (2014). Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, 5(3), 255–263.

Hogan, A., et al. (2021). Knowledge graphs. *ACM Computing Surveys*, 54(4), 71:1–71:37.

International Electrotechnical Commission. (n.d.). *IEC Common Data Dictionary (IEC CDD)*. Retrieved April 27, 2025, from https://cdd.iec.ch

International Electrotechnical Commission. (2005). *IEC Common Data Dictionary (IEC CDD)*. Retrieved April 22, 2025. https://cdd.iec.ch

International Electrotechnical Commission. (2017). *IEC 61360 - Standard data element types with associated classification scheme for electric components*. Retrieved April 22, 2025. https://cdd.iec.ch/CDD/iec61360/iec61360.nsf/Welcome?OpenPage

Jurafsky, D., & Martin, J. H. (2021). *Speech and language processing* (3rd ed.). Draft version. Retrieved from https://web.stanford.edu/~jurafsky/slp3/

Mai, H. T., Chu, C. X., & Paulheim, H. (2024). Do LLMs really adapt to domains? An ontology learning perspective. *arXiv preprint*, arXiv:2407.19998. https://arxiv.org/abs/2407.19998

Maynard, D., Peters, W., & Li, Y. (2006). Metrics for evaluation of ontology-based information extraction. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*.

Neo4j. (2012). *Neo4j – The world's leading graph database*. http://neo4j.org/

Noy, N. F., & Musen, M. A. (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment. *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, 450–455.

OpenAI. (2023). *ChatGPT* [Large language model]. https://chat.openai.com/

OpenAI. (2024). *Pricing*. https://openai.com/pricing

Ponzetto, S. P., & Strube, M. (2007). Deriving a large-scale taxonomy from Wikipedia. *Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, 1440–1445. https://aaai.org/papers/01440-aaai07-228-deriving-a-large-scale-taxonomy-from-wikipedia

Porzel, R., & Malaka, R. (2004). A task-based approach for ontology evaluation. *Proceedings of the ECAI 2004 Workshop on Ontology Learning and Population*, 1–7. https://www.researchgate.net/publication/258927570_A_Task-based_Approach_for_Ontology_Evaluation

Portisch, J., Costa, G., Stefani, K., Kreplin, K., Hladik, M., & Paulheim, H. (2022). Ontology matching through absolute orientation of embedding spaces. *arXiv preprint*, arXiv:2204.04040. https://arxiv.org/abs/2204.04040

Poveda-Villalón, M., Gómez-Pérez, A., & Suárez-Figueroa, M. C. (2014). OOPS! (OntOlogy Pitfall Scanner!): An on-line tool for ontology evaluation. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2), 7–34. https://doi.org/10.4018/ijswis.2014040102

Princeton University. (2010). *About WordNet*. WordNet. https://wordnet.princeton.edu/citing-wordnet

Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4), 334–350. https://doi.org/10.1007/s007780100057

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 448–453.

Riloff, E. (2014). Semantic class learning from the web. *CS 6961: Information Extraction from Text*. https://my.eng.utah.edu/~cs6961/slides/hyponym-patterns.4.pdf

Shvaiko, P., & Euzenat, J. (2005). A survey of schema-based matching approaches. *Journal on Data Semantics IV*, 146–171. https://doi.org/10.1007/11603412_5

Singh, C., Inala, J. P., Galley, M., Caruana, R., & Gao, J. (2024). Rethinking interpretability in the era of large language models. *arXiv preprint*, arXiv:2402.01761. https://arxiv.org/abs/2402.01761

Spyns, P., Meersman, R., & Jarrar, M. (2002). Data modelling versus ontology engineering. *SIGMOD Record*, 31(4), 12–17. https://sigmodrecord.org/publications/sigmodRecord/0212/SPECIAL/2.Meersman.pdf

Wilkinson, M. D., et al. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3, 160018.

World Wide Web Consortium. (2014). *RDF 1.1 Primer* (W3C Working Group Note). https://www.w3.org/TR/rdf11-primer/

Xu, K., Feng, Y., Li, Q., Dong, Z., & Wei, J. (2025). Survey on terminology extraction from texts. *Journal of Big Data*, 12(29). https://journalofbigdata.springeropen.com/articles/10.1186/s40537-025-01077-x

Appendix A

A.1 Code Samples

```
import spacy
nlp = spacy.load("en_core_web_sm")

doc = nlp("The resistor has a tolerance of ±5%.")

for token in doc:
    print(token.text, token.pos_)
```

```
Output:
The       DET
resistor  NOUN
has       VERB
a         DET
tolerance NOUN
of        ADP
±5%       NUM
.         PUNCT
```

Listing 2: Demonstration of using SpaCy's pretrained model to assign a POS tag to every token

```
# XQ-214 Quantum Resistor Datasheet

**Model Number**: XQ-214
**Manufacturer**: Quantech Electronics
**Release Date**: March 2025
**Revision**: 1.2

---

## Overview

The XQ-214 Quantum Resistor is a next-generation passive component
designed for precision resistance control in high-frequency quantum
computing circuits. Engineered with a superconducting substrate and active
thermal damping, it ensures ultra-stable resistance under fluctuating
cryogenic temperatures.

---

## Key Features

- Nominal resistance: **214.7 Ohms ± 0.05%**
- Temperature coefficient: **0.2 ppm/°C**
- Operational temperature range: **-273°C to -100°C**
- Supports quantum entanglement-safe routing
- Compatible with Q-Bus V2 and V3 architectures
- Built-in EMI suppression lattice
- Moisture-resistant nano-coating

---

## Absolute Maximum Ratings

| Parameter                   | Value           | Unit      |
Notes                              |
|-----------------------------|-----------------|-----------|--------------
----------------------|
| Operating Voltage           | ±12             | V         | Differential
input                 |
| Continuous Power Dissipation| 0.8             | W         | At 25°C
ambient                    |
| Peak Pulse Current          | 150             | mA        | For 10 µs
pulse width              |
| Storage Temperature Range   | -300 to +50     | °C        | Long-term
storage in vacuum       |

---
```

Listing 3: Sample Synthetic Datasheet (Markdown) generated using ChatGPT (OpenAI, 2023)