ODKAR: “Ontology-Based Dynamic Knowledge Acquisition and Automated Reasoning Using NLP, OWL, and SWRL”

Claire Ponciano[a], Markus Schaffert[b] and Jean-Jacques Ponciano[c]

i3mainz, University of Applied Sciences, Germany

{*claire.ponciano, markus.schaffert, jean-jacques.ponciano*}*@hs-mainz.de*

Keywords: Ontology Generation, Natural Language Processing (NLP), OWL (Web Ontology Language), SWRL (Semantic Web Rule Language), Text-to-Ontology, Knowledge Extraction, ChatGPT Comparison, Knowledge Representation.

Abstract: This paper introduces a novel approach to dynamic ontology creation, leveraging Natural Language Processing (NLP) to automatically generate ontologies from textual descriptions and transform them into OWL (Web Ontology Language) and SWRL (Semantic Web Rule Language) formats. Unlike traditional manual ontology engineering, our system automates the extraction of structured knowledge from text, facilitating the development of complex ontological models in domains such as fitness and nutrition. The system supports automated reasoning, ensuring logical consistency and the inference of new facts based on rules. We evaluate the performance of our approach by comparing the ontologies generated from text with those created by a Semantic Web technologies expert and by ChatGPT. In a case study focused on personalized fitness planning, the system effectively models intricate relationships between exercise routines, nutritional requirements, and progression principles such as overload and time under tension. Results demonstrate that the proposed approach generates competitive, logically sound ontologies that capture complex constraints.

1 INTRODUCTION

The integration of Natural Language Processing (NLP)(Shamshiri et al., 2024; Chen et al., 2024; Yin et al., 2024; Osman et al., 2024) with Semantic Web technologies (Matthews, 2005) offers significant potential for creating intelligent systems that automatically convert unstructured text into formal knowledge representations. Ontologies (Fensel and Fensel, 2001), as a cornerstone of the Semantic Web, provide a structured, machine-readable format for representing domain knowledge, while automated reasoning (Wang et al., 2004) over these ontologies enables systems to infer new information, ensure logical consistency, and support sophisticated decision-making processes. This combination opens up new possibilities for dynamically acquiring, managing, and reasoning over knowledge extracted from natural language inputs.

Traditionally, the construction and management of ontology-based systems have required expert knowledge of formal description logic languages, such as OWL (Web Ontology Language)(Antoniou and Harmelen, 2009), RDF(Pan, 2009) and SWRL (Semantic Web Rule Language)(Horrocks et al., 2004). Creating ontological definitions, formulating logical rules, and implementing automated reasoning mechanisms are technically demanding tasks that necessitate a deep understanding of Semantic Web technologies. This complexity poses a barrier for domain experts who may excel in their fields but lack the specialized skills required to formalize knowledge in ontological formats. Consequently, there is a growing need for tools that facilitate the automatic generation of ontologies such as (Ponciano et al., 2022; Prudhomme et al., 2020; Prudhomme et al., 2017), lowering the technical barriers for non-experts in Semantic Web technologies.

This paper addresses this challenge by proposing a system for automatic ontology creation and reasoning. Instead of relying on manual input or conversational agents, the system uses NLP techniques to extract key concepts and relationships from unstructured text, translating them into formal OWL and SWRL representations. Automated reasoning is then applied to maintain logical consistency and infer new knowl-

[a] https://orcid.org/0000-0001-8883-8454
[b] https://orcid.org/0000-0002-7970-9164
[c] https://orcid.org/0000-0001-8950-5723

Ponciano, C., Schaffert, M. and Ponciano, J.
ODKAR: “Ontology-Based Dynamic Knowledge Acquisition and Automated Reasoning Using NLP, OWL, and SWRL”.
DOI: 10.5220/0013071500003825

In *Proceedings of the 20th International Conference on Web Information Systems and Technologies (WEBIST 2024)*, pages 457-465
ISBN: 978-989-758-718-4; ISSN: 2184-3252

edge. The system's ability to convert text into structured knowledge is evaluated through a comparative analysis, contrasting the results of our approach with those generated by ChatGPT and a Semantic Web technologies expert.

To demonstrate the system's capabilities, we apply it to the domain of personalized fitness planning, modeling intricate relationships between exercises, nutrition, and training principles such as progressive overload and time under tension. The resulting ontologies are used to generate tailored fitness plans that accommodate individual needs and goals. Our system contributes to the field of artificial intelligence and the Semantic Web by providing a framework for the automated creation of ontologies, enabling real-time reasoning and supporting the development of expert systems from unstructured text, even by non-experts in Semantic Web technologies.

2 RELATED WORK

The reviews and surveys about ontology learning techniques (Wong et al., 2012; Asim et al., 2018; Al-Aswadi et al., 2020) agreed on three main techniques for ontology learning from text: linguistics-based, statistics-based, and logic-based techniques. Statistics-based and logic-based techniques are classified as machine learning approaches in the review of (Al-Aswadi et al., 2020).

Linguistics-based techniques are mainly based on natural language processing (NLP) tools (Wong et al., 2012). NLP-based ontology generation (Osman et al., 2024; Yin et al., 2024; Chen et al., 2024; Shamshiri et al., 2024; Sui et al., 2010; Zhang et al., 2023) focuses on the automatic extraction of entities, relationships, and rules directly from unstructured text. +

Statistics-based techniques are mostly derived from information retrieval, machine learning, and data mining (Wong et al., 2012). They are mainly used for term extraction, concept extraction and taxonomic relationship extraction and most make extensive use of probabilities (Asim et al., 2018). They include C/NC, contrastive analysis, clustering, co-occurrence analysis, term subsumption and ARM (Asim et al., 2018).

Logic-based techniques are presented as the least common and based on knowledge representation and rule-based reasoning by (Wong et al., 2012), whereas in (Asim et al., 2018), the authors present Inductive Logic Programming as a discipline of machine learning that derives hypothesis based on background knowledge and a set of examples using logic programming, which is used to acquire general axioms from schematic axioms. For example, the approach (Shamsfard and Barforoush, 2004) is based on a Kernel ontology and a rule-based system for handling the linguistic, semantic, syntactic, morphological and grammatical aspects. The texts processed by this approach enrich the Kernel's knowledge.

In addition to these three techniques, (Wong et al., 2012) presents also, hybrid approaches that combine several of the previous presented techniques. These hybrid approaches can be explained by the different roles of the different techniques in the methodology for ontology learning as presented by (Asim et al., 2018). Linguistic-based techniques seem to be a must for the pre-processing in ontology learning method. Steps of concepts/terms and relations extraction are generally done through statistics, linguistic- based techniques or a combination of both. Finally, axiom step is done through inductive logical programming (Asim et al., 2018). One example of hybrid approach is Text2Onto (Cimiano and Völker, 2005), which integrates machine learning with basic linguistic processing (such as tokenization and shallow parsing) to model ontologies probabilistically. By adding probabilistic reasoning, Text2Onto allows for scalable and flexible ontology creation, making it a hybrid method between NLP, machine learning, and statistical analysis. In addition, methods like OpenIE (Etzioni et al., 2011) employ NLP to extract relationships and entities, which are then processed using learning-based systems to identify the arguments. These systems enhance the generation of knowledge graphs, but they often lack built-in support for rule-based reasoning, which limits their utility in domains requiring advanced logical consistency.

The review (Al-Aswadi et al., 2020) also highlights the benefit of deep-learning in comparison to machine learning (referred as a shallow learning).

Deep learning models, including BERT (Kenton and Toutanova, 2019) and GPT (Brown et al., 2020), have been applied to ontology generation and knowledge extraction due to their ability to learn complex patterns from large datasets. These models excel at representing semantic context within text, performing well across diverse domains for tasks such as entity extraction and relationship identification. Recent transformer-based models (Mihindukulasooriya et al., 2023; Yenduri et al., 2024) have shown improvements in domain-specific text understanding, but they often struggle to convert unstructured text into formal ontological models that adhere to strict logical constraints. Large Language Models (LLMs), such as GPT-4 (Baktash and Dawodi, 2023), can produce semantically rich outputs but still face limitations in formal reasoning and ontology-based rule enforcement

(Mukanova et al., 2024).

Our approach presented in this paper is a hybrid one, combining linguistic-based and logic-based techniques. It seeks to bridge these gaps by offering a fully automated system that uses semantic and rule-based reasoning to ensure logical consistency in ontologies generated from unstructured text. Unlike existing approaches, which rely heavily on deep learning and machine learning models, our system integrates real-time reasoning based on semantic rules, producing formal OWL and SWRL ontologies with minimal human intervention.

3 METHODOLOGY

This section details the methodology used to develop the ontology-based approach proposed, including the system architecture, natural language processing techniques, ontology management, and reasoning mechanisms. To illustrate the methodology, we take the following text that describes concept in human language as input:

The physical training component is detailed through various exercises and sessions. Exercises like the Bench Press and Squats are included in this structure, each targeting specific muscle groups. For example, the Bench Press targets the chest, and Squats target the legs. These exercises are categorized based on whether they are compound exercises, such as Bench Press and Squats, which engage multiple muscle groups, or isolation exercises like Bicep Curls, which target specific muscles.

3.1 System Architecture

The system architecture is designed with modularity and scalability in mind, ensuring that the various components responsible for natural language processing, ontology management, and reasoning can be independently developed, maintained, and extended. The architecture consists of three primary components: the Natural Language Processing (NLP) module, the Ontology Management module, and the SWRL-based Reasoning module. The two last modules use a reasoning engine for inference and consistency checking tasks. The ontology is stored into a knowledge base.

3.2 Natural Language Processing Module

The Natural Language Processing (NLP) module is essential for transforming natural language input into structured data with an ontological form. This module allows users to interact with the system using everyday language, without requiring expertise in ontology creation. It encompasses several stages that translate user inputs into formal knowledge representations, mapped to Web Ontology Language (OWL) classes, properties, and individuals, ensuring consistent and accurate updates to the knowledge base.

The NLP pipeline processes natural language inputs by breaking down text, identifying entities and relationships, and converting these into structured data. The steps involved are described in this section.

3.2.1 Step 1: Tokenization and POS Tagging

The input sentence is first broken into smaller units (tokens) during tokenization. Each token is assigned a Part-of-Speech (POS) tag, identifying its grammatical role, such as noun, verb, or adjective. For example, the sentence *"Exercises like the Bench Press and Squats are included in this structure, each targeting specific muscle groups"* would be tokenized as:

```
TOKENS = ["Exercises", "like", "the",
"Bench", "Press", "and", "Squats", "are",
"included", "in", "this", "structure", ",",
"each", "targeting", "specific", "muscle",
"groups", "."]
```

Each token is POS-tagged as follows: `"Bench"` (noun), `"Press"` (noun), `"targeting"` (verb), and so on. This establishes the syntactic structure of the sentence, which is further analyzed in the following steps.

3.2.2 Step 2: Named Entity Recognition (NER)

Once tokenization and POS tagging are complete, Named Entity Recognition (NER) identifies key entities in the text. Entities generally represent objects, people, or domain-specific terms. In this example, `"Bench Press"`, `"Squats"`, and `"muscle groups"` are identified as key entities. This step narrows down the critical elements of the sentence, helping the system focus on the most meaningful components.

The identified entities are:

```
ENTITIES = ["Bench Press", "Squats",
"muscle groups"]
```

3.2.3 Step 3: Dependency Parsing

Dependency parsing is then applied to analyze the grammatical structure and relationships between words. It builds a graph where tokens are connected by grammatical dependencies. For example, in the sentence *"Bench Press targets the chest"*, the system identifies that `"Bench Press"` is the subject, `"targets"` is the action, and `"chest"` is the object.

These relationships provide deeper insights into the meaning of the sentence.

```
DEP_GRAPH = [
 {"token": "BenchPress", "dep": "nsubj
     ", "head": "targets"},
 {"token": "targets", "dep": "ROOT"},
 {"token": "chest", "dep": "dobj", "
     head": "targets"},
]
```

3.2.4 Step 4: Entity and Relation Extraction

Next, the system extracts entities and their relationships based on the dependency parsing results. For instance, in the sentence *"Bench Press targets the chest"*, the system extracts that `"BenchPress"` is the subject, `"targets"` is the relation, and `"chest"` is the object. This extraction forms the basis for structuring the sentence into meaningful data.

```
SUBJECT = ["BenchPress"]
RELATION = ["targets"]
OBJECT = ["chest"]
```

3.2.5 Step 5: Mapping to OWL

The final step involves mapping the extracted entities and relationships to OWL. The system translates subjects, objects, and relationships into corresponding OWL classes, properties, and individuals. For example, `"BenchPress"` is mapped as an individual of the `ex:Exercise` class, `"chest"` as an individual of the `ex:MuscleGroup` class, and the `targets` relationship is mapped as an OWL object property. This step results in the following mappings that is referenced as (ΔO) later in this paper.

- **Classes:**
  - `ex:Exercise`
  - `ex:MuscleGroup`
  - `ex:ExerciseType`
- **Individuals (of Exercises):**
  - `ex:BenchPress`
  - `ex:Squats`
  - `ex:BicepCurls`
- **Individuals (of Muscle Groups):**
  - `ex:chest`
  - `ex:legs`
- **Individuals (without class):**
  - `ex:SpecificMuscles`
- **Object Properties:**
  - `ex:targets` (connects exercises to muscle groups)
- **Relationships:**
  - `ex:BenchPress ex:targets ex:chest`
  - `ex:Squats ex:targets ex:legs`
  - `ex:BicepCurls ex:targets ex:SpecificMuscles`

3.3 Ontology Management Module

The Ontology Management Module (OMM) is a core component of the system, responsible for overseeing the continuous management and evolution of the OWL ontology, which acts as the formal knowledge representation. This module dynamically integrates new information into the ontology, ensures logical consistency, and supports versioning and rollback mechanisms. By managing the dynamic construction of the ontology, enforcing logical constraints, and performing consistency checks, the OMM ensures the system remains robust and adaptable to new data, maintaining the overall integrity and coherence of the knowledge base.

The OMM integrates dynamically new knowledge (new classes, properties, individuals and relationship resulting from the NLP module) into the ontology. As each new text is processed by the NLP module, the ontology evolves without requiring manual intervention or redesign. This section presents the detailed functionalities for the process of dynamic construction and the integration of new knowledge into the existing ontology. These functionalities are illustrated through the process of the NLP module output (ΔO) that includes new classes, individuals, and properties, such as exercises and muscle group.

3.3.1 Functionality 1: Dynamic Ontology Construction

New knowledge from ΔO, such as new classes (e.g., `ex:Exercise`), new properties (e.g., `ex:targets`), new individuals (e.g., `ex:Bench Press`) and new relationships (e.g., `ex:Bench Press ex:targets ex:chest`) is added to the current ontology (O) dynamically.

3.3.2 Functionality 2: Consistency Checking

Each time that a new element (such as a new element from ΔO, a relationship or a constraint) is integrated, the OMM performs consistency checks. This is achieved by employing a description logic reasoner (e.g., Pellet or HermiT):

- If the consistency check passes, the newly added elements are fully integrated into the ontology.

- If an inconsistency is detected, the new element is rejected, ensuring the ontology remains logically sound.

3.3.3 Functionality 3: Versioning and Rollback

To maintain the integrity of the knowledge base, the OMM supports version control. After consistency checks, the updated ontology is saved as a new version. This feature ensures that the ontology's history is preserved and can be rolled back to a previous, stable version if inconsistencies arise in the future.

- **Version Control:** A new version of the ontology is created, incorporating the latest addition.
- **Rollback:** In case of error or inconsistency, the system can revert to the previous version, preserving the integrity of the knowledge base.

3.3.4 Functionality 4: Enforcing Logical Constraints

Once the new consistent elements from ΔO are integrated to the knowledge base, the OMM enforces logical constraints, through class hierarchies and property restrictions. This reinforces the logical structure of the ontology.

1. **Class Hierarchy:**
   - The class hierarchy construction uses the NER result from the NLP module (c.f. 3.2.2). Each entity (E) composed of several words and classified as a class is analyzed. If a class (C) exists whose name is the last word of entity E, then the class resulting from entity E is defined as a subclass of C.
   - For example, the two classes `ex:CompoundExercise` and `ex:IsolationExercise` (resulting from the two entities "Compound Exercise" and "Isolation Exercise" respectively) are defined as subclass of `ex:Exercise`.
2. **Property Restrictions:**
   - The property restriction construction uses the dependency parsing result from the NLP module (c.f. 3.2.3) to identify classes that fit the domain and range of a property.
   - For example, this process defines `ex:Exercise` as the domain of the property `ex:targets` and `ex:Muscle Group` as its range.

3.3.5 Functionality 5: Inference

Once the consistent constraints have been added to the knowledge base, an inference process is triggered by using Pellet reasoner engine. It allows for adding new knowledge. For example, the individual `ex:specificMuscle` becomes an instance of the class `ex:MuscleGroup`, due to `ex:MuscleGroup` is defined as the range of the property `ex:targets` and the knowledge base contains the axiom `ex:BicepCurls ex:targets ex:SpecificMuscles`.

3.4 SWRL-Based Reasoning Module

The SWRL-Based Reasoning Module plays a critical role in enhancing the system's ability to infer new knowledge, going beyond the standard reasoning capabilities provided by OWL alone. This module defines and applies logical rules, specified using the Semantic Web Rule Language (SWRL). It allows the system to autonomously generate new facts by evaluating the SWRL rules in conjunction with the ontology's structural aspects. By applying these rules, the system can infer additional knowledge that may not be explicitly stated in the ontology.

The SWRL-Based Reasoning Module follows a structured process to apply rules and infer new knowledge from the ontology. The key steps in the execution are outlined in this section.

3.4.1 Step 1: Rule Definition

This module defines SWRL rule, through a pattern search based on structures and keywords such as:

- `if` {A} `then` {B} that produces

$$\mathtt{A} \rightarrow \mathtt{B}$$

- `whether` {A}, `which/that` {B} `or` {C}, `which/that` {D} that produces

$$\mathtt{B} \rightarrow \mathtt{A}$$

$$\mathtt{D} \rightarrow \mathtt{C}$$

In the text provided in section 3, the analysis of the sentence:"*These exercises are categorized based on* ***whether*** *they are* ***compound exercises****, such as Bench Press and Squats,* ***which engage multiple muscle groups****, or* ***isolation exercises*** *like Bicep Curls,* ***which target specific muscles***." produces the definition of the following SWRL rules:

- **Rule R1:**

$$\mathtt{ex:targets}(?\mathtt{x}, ?\mathtt{y}) \wedge \mathtt{ex:MuscleGroup}(?\mathtt{y}) \rightarrow \mathtt{ex:CompoundExercise}(?\mathtt{x})$$

- **Rule R2:**

$$\mathtt{ex:targets}(?\mathtt{x}, \mathtt{ex:SpecificMuscles}) \rightarrow \mathtt{ex:IsolationExercise}(?\mathtt{x})$$

The rule R1 means that if X targets multiple muscle groups, X should be classified as an instance of compound exercise. Similarly, the rule R2 might infer that if X targets a specific muscle, it should be classified as an instance of isolation exercise.

3.4.2 Step 2: Rule Application

Once the rules are defined, the module applies them to the ontology using the Pellet reasoner engine. Based on the rules R1 and R2, the following inferences are made:

- `ex:Bench Press a ex:CompoundExercise` (from R1)
- `ex:Squats a ex:CompoundExercise` (from R1)
- `ex:BicepCurls a ex:IsolationExercise` (from R2)

3.4.3 Step 3: Consistency Checking

After new facts are inferred, the module performs consistency checks to ensure that the inferred knowledge does not introduce any logical contradictions. If inconsistencies are detected, the conflicting inferred facts and the rule it comes from are discarded, preserving the integrity of the ontology.

3.4.4 Real-Time Inference and Scalability Handling

The SWRL-Based Reasoning Module supports real-time inference, applying checked and consistent rules immediately after new data is integrated into the ontology. This real-time inference allows the system to update the ontology with inferred knowledge as new facts are added, enabling dynamic decision-making. For example, when a new exercise is added to the ontology, the system automatically infers its classification as either a compound or isolation exercise, depending on the muscle group it targets.

As the ontology grows, the number of SWRL rules may increase, requiring the system to handle reasoning efficiently. To address scalability, the SWRL-Based Reasoning Module employs incremental reasoning techniques, which focus on evaluating only the parts of the ontology that are affected by recent changes. This approach minimizes computation and ensures that the system remains responsive, even as the ontology becomes more complex.

4 EVALUATION

To assess the performance of the three ontologies generated from the same source text, a comprehensive evaluation was conducted comparing the ontologies produced by ChatGPT[1], an expert in Semantic Web technologies, and the proposed approach. The evaluation was based on several critical criteria, including completeness, consistency, correctness, richness, reasoning capabilities, and the level of human effort required to create each ontology. These criteria provide a detailed view of each approach's strengths and weaknesses. The input text and the corresponding ontologies used in this evaluation are publicly accessible via the following repository: https://github.com/JJponciano/ODKAR.

4.1 Completeness

Completeness *assesses how well each ontology captures the concepts and relationships described in the input text (cf. Physical Training Structure).*

Proposed Approach: The ontology generated by our system includes 15 classes, 3 object properties, 22 data properties, 11 individuals, and 158 axioms. It captures key entities such as *Person*, *Exercise*, *Muscle Group*. It includes object properties, such as *targets* and *partOfSession*, and data properties (e.g., *currentWeight*, *currentReps*).

ChatGPT: ChatGPT's ontology includes 9 classes, 3 object properties, 13 data properties, 11 individuals, and 84 axioms. It covers basic concepts such as *Person*, *Exercise*, and *MuscleGroup*, but lacks concepts such as Workout routine or time under tension for example. It includes properties such as *hasWeight* and *targetsMuscleGroup*, but misses a lot of data properties such as volume and start date for example.

Expert: The expert ontology is the most comprehensive, capturing all major concepts and relationships. It includes 10 classes, 10 object properties, 38 data properties, 21 individuals, and 178 axioms.

4.2 Consistency

Consistency *measures whether the ontology is logically sound and free of contradictions.*

Proposed Approach: The ontology produced by the presented approach is consistent with OWL restrictions that are included. It covers some interesting restrictions such as domain and range of properties. Processes of consistency checking intervening in

[1]ChatGPT, the official app by OpenAI, version 1.2024.275

Table 1: Comparison of Ontologies Generated by Different Approaches.

Evaluation Criteria	Proposed Approach	ChatGPT	Expert Approach
Completeness	High	Medium	Very High
Consistency	High	High	Very High
Correctness	High	Medium	Very High
Richness	Medium-High	Poor	Very High
Reasoning Capabilities	Medium	No	High
Human Effort Required	Low	Medium	High

OMM and in SWRL-based reasoning module allow to add elements, restrictions and constraints to an ontology with a better logically sound, while ensuring consistency.

ChatGPT: The ChatGPT-generated ontology has a good consistency due to the lack of logical restrictions. There is no inconsistency without restrictions and constraints.

Expert: The expert ontology enforces strict consistency through OWL restrictions. It ensures that logical constraints like *currentWeight* and *proteinIntake* follow domain-specific constraints and that no contradictory information is present.

4.3 Correctness

***Correctness** evaluates how well the ontology adheres to domain-specific knowledge.*

Proposed Approach: The generated ontology captures domain knowledge well, including correct classifications of exercises as compound or isolation exercises. It also correctly models exercise instances (e.g., *Bench Press*, *Squats*, and *BicepCurls*). However, it has some gaps, such as incomplete modeling of finer distinctions in muscle groups, like `SpecificMuscle`.

ChatGPT: While ChatGPT correctly identifies basic entities like *Bench Press* and *Squats* as instances of *ex:CompoundExercise*, it also creates *ex:CompoundExercise* and *ex:IsolationExercise* as two instances of *ex:Exercise* rather than creating a class hierarchy between the three classes.

Expert: The expert ontology reflects domain knowledge with high accuracy. It includes all classifications, logical relationships, and constraints (e.g., protein intake per body weight), making it the most correct and comprehensive.

4.4 Richness

***Richness** measures the complexity and detail within the ontology, including class hierarchy, constraints and restrictions to capture a fine domain knowledge.*

Proposed Approach: Although it retains a lower level of refinement than that of an expert, the ontology captures some fine knowledge such as the hierarchy of Exercise classes, some OWL restrictions such as domain and range of properties and some SWRL rules (R1 and R2, c.f. subsection 3.4.1).

ChatGPT: ChatGPT's ontology does not include class hierarchy, logical constraints and restrictions, limiting the detail and depth of its representation.

Expert: The expert ontology is the richest, with a well class hierarchy, an extensive use of SWRL rules and OWL restrictions. It models complex relationships and behaviors, such as exercise type classifications, progressive overload, and protein intake constraints.

4.5 Reasoning Capabilities

***Reasoning Capabilities** assesses the ontology's ability to support automated reasoning and infer new knowledge.*

Proposed Approach: The ontology includes some OWL restrictions such as domain and range of properties that allow for some inferences (c.f. specific muscle in subsection 3.3.5) and some SWRL rules (R1 and R2 to infer the classification of exercises as compound or isolation exercise, c.f. subsection 3.4.1). However, it lacks more advanced SWRL rules compared to the expert ontology.

ChatGPT: ChatGPT's ontology does not support reasoning. It lacks OWL restrictions or SWRL rules, meaning no logical inferencing or automated checks are possible.

Expert: The expert ontology includes advanced reasoning capabilities, using both OWL restrictions and SWRL rules to infer new knowledge and enforce domain-specific rules like progressive overload and protein intake ranges.

4.6 Human Effort Required

***Human Effort Required** compares the amount of manual work needed to produce the ontology.*

Proposed Approach: Our approach significantly reduces manual effort by automatically generating the ontology from input text. It requires minimal human intervention, making it a scalable solution.

ChatGPT: ChatGPT's ontology requires more manual intervention. While it can generate a basic structure, the lack of logical constraints and reasoning capabilities means that significant human effort is required to refine and complete the ontology.

Expert: The expert ontology requires the most human effort, as it is manually created by a domain expert. This results in high accuracy and completeness but is time-consuming and requires specialized knowledge.

In this evaluation, the term ”expert” refers specifically to an expert in Semantic Web technologies and ontology engineering, rather than a domain expert in the field of fitness or physical training. This distinction is important when considering the complexity and richness of the expert-generated ontology. It is also critical to note that the techniques applied in this evaluation are not limited to fitness but are broadly applicable to any domain requiring ontology-driven knowledge representation and reasoning.

5 DISCUSSION

The evaluation of the proposed approach to ontology generation reveals several strengths, particularly in terms of completeness, consistency, and correctness, achieved with minimal human intervention. By automating the extraction of structured knowledge from unstructured text, our system significantly reduces the manual effort typically required in ontology creation. While the expert-generated ontology demonstrates superior richness and advanced reasoning capabilities, it requires extensive human effort and domain expertise. This labor-intensive process is often not feasible for dynamic applications, where rapid ontology updates are necessary. In contrast, our approach effectively balances automation and accuracy, allowing domain specialists who may lack deep expertise in Semantic Web technologies to contribute to ontology development. The comparison with ChatGPT underscores the added value of our system. Although ChatGPT can identify key concepts, it lacks the intricate reasoning mechanisms and formal logical structures required for a fully coherent and logically sound ontology. The necessity for significant manual refinement with ChatGPT limits its usability in complex domains. Our system introduces domain-specific logical constraints and consistency checks, which enhance the quality of the generated ontologies and reduce reliance on manual post-processing.

However, these observations are only an initial assessment of the results of our approach. As the evaluation is based on a single text source and not various corpora, it would be necessary to broaden and deepen the evaluation of our approach on several text sources representing various domains in order to further validate and refine its performance. Therefore, our future work will aim to improve the experimental results by including quantitative measures, such as precision, recall and F1 scores, as well as qualitative analyses to support our conclusions. This dual approach will provide a better understanding of the performance of our system compared with existing methods, including other ontology generation systems. In addition, these initial results have enabled us to identify areas for improvement in future research, such as assertion enrichment, SWRL rule creation, and the system's ability to handle complex, multi-layered constraints. With these advances, we aim to increase the practical applicability and efficiency of our approach, ultimately contributing to the broader landscape of knowledge management and representation.

REFERENCES

Al-Aswadi, F. N., Chan, H. Y., and Gan, K. H. (2020). Automatic ontology construction from text: a review from shallow to deep learning trend. *Artificial Intelligence Review*, 53(6):3901–3928.

Antoniou, G. and Harmelen, F. v. (2009). Web ontology language: Owl. *Handbook on ontologies*, pages 91–110.

Asim, M. N., Wasim, M., Khan, M. U. G., Mahmood, W., and Abbasi, H. M. (2018). A survey of ontology learning techniques and applications. *Database*, 2018:bay101.

Baktash, J. A. and Dawodi, M. (2023). Gpt-4: A review on advancements and opportunities in natural language processing. *arXiv preprint arXiv:2305.03195*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chen, S., Zhang, Y., and Yang, Q. (2024). Multi-task learning in natural language processing: An overview. *ACM Computing Surveys*, 56(12):1–32.

Cimiano, P. and Völker, J. (2005). Text2onto: A framework for ontology learning and data-driven change discovery. In *International conference on application of natural language to information systems*, pages 227–238. Springer.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2011). Open information extraction: The second generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 3–10.

Fensel, D. and Fensel, D. (2001). *Ontologies*. Springer.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., Dean, M., et al. (2004). Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21(79):1–31.

Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Matthews, B. (2005). Semantic web technologies. *E-learning*, 6(6):8.

Mihindukulasooriya, N., Tiwari, S., Enguix, C. F., and Lata, K. (2023). Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. In *International Semantic Web Conference*, pages 247–265. Springer.

Mukanova, A., Milosz, M., Dauletkaliyeva, A., Nazyrova, A., Yelibayeva, G., Kuzin, D., and Kussepova, L. (2024). Llm-powered natural language text processing for ontology enrichment. *Applied Sciences (2076-3417)*, 14(13).

Osman, M., Noah, S., and Saad, S. (2024). Identifying terms to represent concepts of a work process ontology. *Indian Journal of Science and Technology*, 17(24):2519–2528.

Pan, J. Z. (2009). Resource description framework. In *Handbook on ontologies*, pages 71–90. Springer.

Ponciano, C., Schaffert, M., Würriehausen, F., and Ponciano, J.-J. (2022). Publish and enrich geospatial data as linked open data. In *WEBIST*, pages 314–319.

Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Cruz, C., and Roxin, A.-M. (2020). Interpretation and automatic integration of geospatial data into the semantic web: Towards a process of automatic geospatial data interpretation, classification and integration using semantic technologies. *Computing*, 102:365–391.

Prudhomme, C., Homburg, T., Ponciano, J.-J., Boochs, F., Roxin, A., and Cruz, C. (2017). Automatic integration of spatial data into the semantic web. In *WEBIST*, pages 107–115.

Shamsfard, M. and Barforoush, A. A. (2004). Learning ontologies from natural language texts. *International journal of human-computer studies*, 60(1):17–63.

Shamshiri, A., Ryu, K. R., and Park, J. Y. (2024). Text mining and natural language processing in construction. *Automation in Construction*, 158:105200.

Sui, Z., Liu, Y., Zhao, J., and Zhang, H. (2010). The development of an nlp-based chinese ontology construction platform. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 190–194. IEEE.

Wang, X. H., Zhang, D. Q., Gu, T., and Pung, H. K. (2004). Ontology based context modeling and reasoning using owl. In *IEEE annual conference on pervasive computing and communications workshops, 2004. Proceedings of the second*, pages 18–22. Ieee.

Wong, W., Liu, W., and Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM computing surveys (CSUR)*, 44(4):1–36.

Yenduri, G., Ramalingam, M., Selvi, G. C., Supriya, Y., Srivastava, G., Maddikunta, P. K. R., Raj, G. D., Jhaveri, R. H., Prabadevi, B., Wang, W., et al. (2024). Gpt (generative pre-trained transformer)–a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*.

Yin, M., Tang, L., Webster, C., Yi, X., Ying, H., and Wen, Y. (2024). A deep natural language processing-based method for ontology learning of project-specific properties from building information models. *Computer-Aided Civil and Infrastructure Engineering*, 39(1):20–45.

Zhang, Z., Elkhatib, Y., and Elhabbash, A. (2023). Nlp-based generation of ontological system descriptions for composition of smart home devices. In *2023 IEEE International Conference on Web Services (ICWS)*, pages 360–370. IEEE.