LLM Assisted Vocabulary Harmonization

Maria Claudia Reis Cavalcanti[1], Samir de Oliveira Ramos[1], Ronaldo Ribeiro Goldschmidt[1], Wallace Anacleto Pinheiro[1], Alexandra Miguel Raibolt da Silva[1], Alex Garcia[1], Bernardo Alkmim[2], Robinson Callou[2], Edward Hermann Haeusler[2], Cecília de Azevedo Castro César[3], Ferrucio de Franco Rosa[3] and José Maria Parente de Oliveira[3,†]

[1]*Military Institute of Engineering (IME), Praça Gen. Tiburcio, 80, Rio de Janeiro - RJ, 22290-270, Brazil*
[2]*Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rua Marquês de São Vicente, 225, Rio de Janeiro -RJ, 22451-900, Brazil*
[3]*Aeronautics Institute of Technology (ITA), Praça Marechal Eduardo Gomes, 50, São José dos Campos - SP, 12228-900, Brazil*

Abstract

One of the early challenges in ontology creation is developing a standardized vocabulary with clear definitions, especially when integrating concepts from various and often conflicting sources. We propose the LLM-Assisted Vocabulary Harmonization (LAVOHA) method, which leverages large language models (LLMs) to systematically analyze and reconcile concept definitions in natural language. Our approach is demonstrated in the cybersecurity domain, where we harmonized definitions from multiple established cybersecurity vocabularies. In a case study, the LAVOHA definitions were evaluated against a human consensus using criteria such as clarity, completeness, and alignment with expert understanding. The results indicate that LAVOHA produces definitions that are more consistent and comprehensive than those generated by an LLM without harmonization guidance. These findings suggest that LAVOHA can significantly enhance the quality and interoperability of ontology vocabularies in complex domains.

Keywords

LLM, Ontology, Vocabulary Harmonization, BM25, RAG

1. Introduction

According to the literature on ontology engineering [1, 2, 3], when formalizing an ontology, building or reusing a glossary of terms is demanded. However, existing glossaries may have problematic definitions due to ambiguous and conflicting terminology. In the cybersecurity incident response domain, many official standardization documents point to conflicting definitions of terms. Identifying the best definition of terms in this context is hard, time consuming, and requires great human effort.

Upon identifying the definitions for the same concept, merging them is a complex task fraught with several key problems, such as inconsistencies, name conflicts, and redundant hierarchies, to name a few. In summary, merging different descriptions of the same concept risks either loss of detail or overgeneralization. The process may fail to distinguish between closely related but distinct concepts, collapsing them into a single broad concept and losing important nuances. In contrast, merging truly identical concepts could be neglected, leading to unnecessary duplication.

In our literature mapping, we identified the need for agile approaches aimed at helping humans

Proceedings of the 18th Seminar on Ontology Research in Brazil (ONTOBRAS 2025) and 9th Doctoral and Masters Consortium on Ontologies (WTDO 2025), São José dos Campos (SP), Brazil, September 29 – October 02, 2025.
*Corresponding author.
†These authors contributed equally.
yoko@ime.eb.br (M. C. R. Cavalcanti); samir.ramos@ime.eb.br (S. d. O. Ramos); ronaldo.rgold@ime.eb.br (R. R. Goldschmidt); wallaceapinheiro@gmail.com (W. A. Pinheiro); raibolt@ime.eb.br (A. M. R. d. Silva); garcia@ime.eb.br (A. Garcia); balkmim@inf.puc-rio.br (B. Alkmim); robinson.rcmbf@gmail.com (R. Callou); hermann@inf.puc-rio.br (E. H. Haeusler); cecilia@ita.br (C. d. A. C. César); ferrucio@ita.br (F. d. F. Rosa); parente@ita.br (J. M. P. d. Oliveira)
0000-0003-4965-9941 (M. C. R. Cavalcanti); 0009-0004-6055-7182 (S. d. O. Ramos); 0000-0003-1688-0586 (R. R. Goldschmidt); 0000-0001-7076-8785 (W. A. Pinheiro); 0000-0002-8982-596X (A. M. R. d. Silva); 0000-0002-2649-1106 (A. Garcia); 0000-0002-6927-6174 (B. Alkmim); 0009-0005-3941-3309 (R. Callou); 0000-0002-4999-7476 (E. H. Haeusler); 0000-0002-0332-3759 (C. d. A. C. César); 0000-0001-9504-496X (F. d. F. Rosa); 0000-0002-7803-1718 (J. M. P. d. Oliveira)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-03-03

CEUR-WS.org/Vol-4177/paper2.pdf

agree on consensual definitions. Natural language processing techniques have been used in ontology engineering and, when effectively fine-tuned, LLMs might work as suitable assistants for ontology construction [4].

We propose the LLM Assisted Vocabulary Harmonization (LAVOHA) method to harmonize definitions from various conflicting sources (e.g., vocabularies and glossaries). Documents are segmented into smaller chunks and transformed into vector embeddings for efficient storage. When a query is received, the system retrieves the most relevant chunks using a similarity function and provides them as a context.

When building an ontology from natural language definitions, engaging in full alignment with existing ontologies is typically premature. At this stage, the focus should be on clarifying concepts, normalizing terminology, and establishing initial structures, tasks that require semantic flexibility and lack the grounding needed for reliable correspondence with formal models. Premature ontology alignment risks distorting the intended meaning or introducing misinterpretations.

To mitigate these risks, the early phase must prioritize vocabulary construction: identifying and standardizing domain-relevant terms to ensure internal coherence. Although this process has not yet involved mapping to external ontologies, it lays the foundation for future alignment by creating a stable semantic base. In this sense, vocabulary construction acts as a form of pre-alignment, shaping definitions and relationships that will later facilitate integration.

For these reasons, this work does not address the ontology alignment itself but the prior step of harmonizing vocabulary definitions. Alignment becomes feasible only once the core vocabulary and conceptual structures have stabilized.

The remainder of this paper is organized as follows. Section 2 provides theoretical foundations; Section 3 presents a synthesis of the literature review, focusing on related work; Section 4 introduces the LLM-Assisted Vocabulary Harmonization method; Section 5 describes a case study on applying the proposed method and discusses the results; and Section 6 presents the conclusions.

2. Foundations

2.1. Ontology Building Methodologies

Most of the methodologies for ontology engineering found in the literature [1, 2, 3] include sub-processes, ranging from requirements elicitation to testing. Usually, in the requirements elicitation sub-process, they gather existing vocabularies and other related standard and reference documents. Figure 1 shows a variation of the main sub-processes (white boxes) of the SABIO methodology [1] in BPMN notation. The conceptualization/formalization sub-process embeds a vocabulary construction task, in which the ontologist must define the terms that can be covered by the ontology. Note that in Figure 1 this task is reified (gray box) and represented as a sub-process. We treated this task as a sub-process because its complexity is greater than it first seemed

According to some authors [5] [6], building a vocabulary, i.e. a list of terms and their corresponding definitions, is not a trivial task and may affect the quality and consistency of the ontology under construction. Many challenges emerge and must be solved with the support of domain experts. Some of the main issues that should be addressed are [5]: (i) *Inconsistencies* – there are plenty of terminological resources, and different definitions coming from these resources can be conflicting and lead to logical contradictions; (ii) *Homonyms* – two different concepts might share the same name but refer to different things, resulting in either unintended duplication and ambiguity, and should be distinguished by using different terms; (iii) *Synonyms* - multiple terms that refer to the same concept may create confusion, thus a preferred term must be chosen; (iv) *Recursive or Circular Definitions* - definitions that define a term in terms of itself, or in terms of another that references it, can create logical inconsistencies, and should be avoided; (v) *Redundant Hierarchies* – merging can introduce multiple paths between concepts or duplicate subclass relationships, cluttering the structure and making maintenance difficult.

Based on these issues, a set of tasks should be planned, such as reconciling definitions, checking for cycles and inconsistencies, choosing preferred terms, disambiguating homonyms, among others.

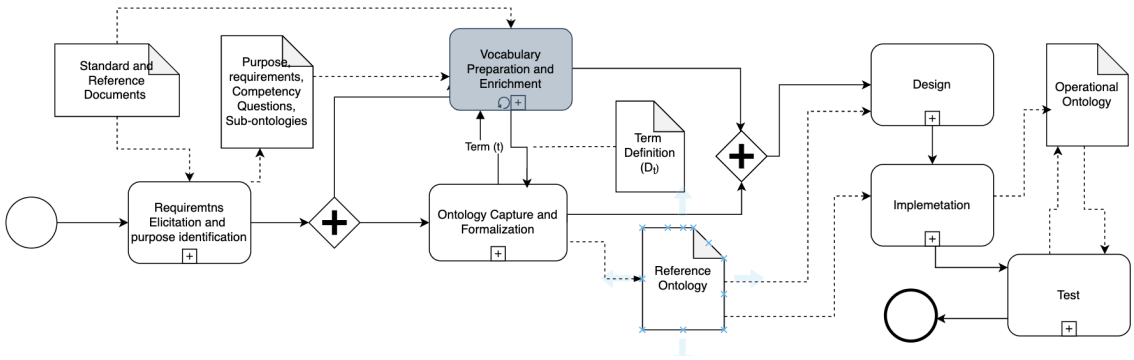

Figure 1: Ontology Engineering Process

Moreover, especially in the case of cybersecurity domain, there are many reference and terminological documents to take into account, which turn the Vocabulary construction into an even more expensive and time-consuming task. In the present work, we intend to address some of these problems in an agile way, by proposing a Vocabulary preparation and enrichment sub-process using a RAG/LLM approach.

2.2. RAG

Based on insights derived from references [7, 8, 9, 10, 11], Retrieval-Augmented Generation (RAG) is a technique that enhances text generation in large language models (LLMs) by integrating information from external, private, or proprietary data sources that are reliable, up-to-date, and provide additional context. This approach improves the accuracy and reliability of the model's output by referencing an external knowledge base before generating a response. Additionally, RAG promotes transparency by linking the generated text to specific, relevant sources, offering users insight into the model's generative process.

The RAG technology has been developing rapidly. RAG originated alongside the Transformers framework, designed to enhance text generation by incorporating external context [12]. Subsequently, the technology focused on prioritizing the most relevant information to enhance LLM responses. RAG was later utilized to assist in fine-tuning LLMs, further improving their contextual awareness and accuracy [13]. Thus, there are different types of RAG, which can be classified based on their implementation, architecture, or approach to integrating external data. Among the various types of approaches, the following stand out [14, 15, 16]:

- **Naïve RAG**: Documents are segmented into smaller chunks and transformed into vector embeddings for efficient storage. When a query is received, the system retrieves the most relevant chunks using a similarity function and provides them as context for a language model.
- **Advanced RAG**: enhances the base model with sophisticated preprocessing (e.g., query reformulation) and post-processing (e.g., document re-ranking) to optimize retrieval accuracy. It can integrate LLMs, Large Retrieval Models (LRMs), and Small Language Models (SLMs) to generate precise, coherent, and contextually enriched responses.
- **Modular RAG**: Allows individual components to be replaced or fine-tuned independently, including the retriever (which fetches relevant data), the processor (which pre-processes information), and the generator (which leverages an LLM or LRM to produce coherent and contextually accurate text).
- **Corrective RAG**: After generating a response, the system cross-references it with trusted data sources to identify and correct inaccuracies, ensuring greater factual accuracy and reliability.
- **Speculative RAG**: Typically utilizes an LLM or LRM to generate plausible responses by combining retrieved information with pattern-based reasoning and informed assumptions.

2.3. Information Retrieval and Ranking Functions

Information Retrieval (IR) [17, 18] is an area concerned with the extraction of desired information from various sources, but mainly textual content. The most used models in IR are vector space models and probabilistic models, although there are techniques that do not involve embedding of text, such as ranking functions.

Vector space models[19, 20] aim to *embed* words (or even entire fragments of text) as vectors in a high-dimensional space. *Relevance* or *similarity* are then translated to *distance* between such vectors, which can be represented in many ways, but mostly via the dot-product between two vectors.

Probabilistic models are based on the principle that documents in a corpus should be ranked by decreasing probability of their relevance to a queried term - the Probabilistic Ranking Principle [21]. Different implementations provide their own probability estimation techniques, and each domain requires adequate crafting of them.

Ranking Functions are another way to retrieve information from texts - usually paired together with word embeddings. Given a set (usually called a *corpus*) of fragments of text (*documents*)[1] and a certain term to be queried in these documents, this function creates a score for each - indicating the relevance of each to the queried term.

Tf-idf[22] is a ranking function that stands for a mix of *term frequency* and *inverse document frequency*. It is, in fact, a way to take both these concepts into account when scoring documents regarding certain queried terms. Term frequency is the amount of times the queried term appears in each document. Inverse document frequency is the logarithm of the inverse of the frequency with which the term appears in all of the documents - indicating how relevant the document in question is compared to the others. These scores are, then, multiplied (let N be the number of documents, d a document and q a queried term):

$$\text{TF-IDF}(d, q) = \text{tf}(d, q) \cdot log\left(\frac{N}{\text{df}(d)}\right)$$

Over decades, Tf-idf has been used in several applications (e.g.[23]), but it does have certain limitations. It does not consider concepts such as normalization and saturation. Normalization is related to the size of each document, i.e. if a term appears 10 times in a document that has 100 words, it should be more relevant than appearing 11 times in 1000 words. Saturation indicates that there must be a threshold up to which the term is still relevant in the document - the more it appears above this point, the less important the syntactical presence of the term in the document is to its semantics.

BM25[24] stands for *Best Match 25* (25 indicates that it was the 25th iteration of the refinement of this algorithm). It is a series of improvements over Tf-idf considering concepts such as the ones described above:

$$\text{BM25}(d, q) = \sum_{t \in q} log\left(\frac{N}{\text{df}(t)}\right) \cdot \frac{(k_1 + 1) \cdot \text{tf}(t, d)}{k_1 \cdot \left((1 - b) + b \cdot \frac{len(d)}{len_{avg}}\right) + \text{tf}(t, d)}$$

k_1, b - parameters
$len(d)$ - size of document d
len_{avg} - average document size

There have been many implementations of BM25 over the years [25, 26, 27], each adjusting parameters, considering different domains, and considering different linguistic concepts. However, most of them have better performance than plain Tf-idf in most scenarios.

[1] In this section we will utilize these terms to define ranking functions, but they are confusing when taking our implementation into account. In the following sections, we will refer to a *corpus* as a *document*, and the documents in it exclusively as *fragments of text* or *sentences*.

2.4. Cybersecurity

Cybersecurity, the domain addressed in this article, is the prevention of damage to, protection of, and restoration of computers, electronic communications systems, electronic communications services, wire communication and electronic communication, including information contained therein, to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation [28]. The ontological analysis of the cybersecurity domain is highly valued, as it reveals how different initiatives model reality in distinct ways, which can directly impact the understanding of the domain, the interoperability of security systems, and the subsequent implementation of policies and actions. We will extract fundamental concepts from these initiatives; therefore, we provide a brief description of each.

Several initiatives have been proposed to guide best practices in cybersecurity, providing guidelines for identifying, preventing, and responding to threats. Some of the most prominent cybersecurity frameworks and standards include the MITRE strategies, NIST approaches, ISO/IEC 27001, CIS Controls, COBIT, and OWASP.

MITRE is an American corporation that has developed two complementary frameworks: MITRE ATT&CK and MITRE D3FEND. The philosophy of MITRE ATT&CK is to move from a reactive defense to a proactive defense based on adversary behavior [29]. Therefore, they focus on how adversaries achieve their objectives, i.e., the tactics, techniques, and procedures (TTPs) employed in real-world attacks. MITRE D3FEND, on the other hand, focuses on defensive countermeasures mapped onto ATT&CK offensive tactics and techniques [30]. Once one knows how adversaries attack, one can detail how to defend against them.

NIST (National Institute of Standards and Technology) produces a wide range of publications, standards, and frameworks related to cybersecurity. The NIST Cybersecurity Framework (CSF) is a high-level strategic guide for organizations to manage and mitigate cyber risk [31]. It is organized into six main functions: Govern (establishing strategy, expectations, and policies), Identify (mapping assets, risks, and vulnerabilities), Protect (implementing controls such as encryption and access management), Detect (continuous monitoring to identify incidents), Respond (executing mitigation and communication plans), and Recover (restoring services and improving resilience). The CSF acts as an aggregator, pointing to other standards and norms for the implementation details. The NIST Special Publication (SP) 800 Collection is the most comprehensive collection of NIST cybersecurity guidelines and recommendations, such as NIST 800-53, a catalog of security controls that organizations should implement [28]. This extensive catalog provides technology-agnostic controls used by risk teams, while security operations teams more commonly use MITRE mitigations. It is possible to map between the two to find a common language. The glossary of terms [32] used in all technical publications is useful for ontology design.

ISO/IEC 27001 [33], developed by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), is an international standard that defines requirements for an Information Security Management System (ISMS), focusing on the confidentiality, integrity, and availability of data. It covers risk management and security controls organized into domains (such as policies, asset management, incident response, and certification). An organization may seek an ISO 27001 certification to demonstrate compliance with an international ISMS standard, which often covers a large portion of the NIST CSF requirements and implements many of the controls in NIST SP 800-53.

CIS Controls [34], developed by the Center for Internet Security (CIS), provide a set of practical and prioritized controls to protect systems and data. It is focused on immediate implementation, making it accessible for organizations seeking quick and effective solutions. The CIS Controls are closely aligned with the NIST approaches, sharing common goals and complementary structures.

Control Objectives for Information and Related Technology (COBIT), developed by the Information Systems Audit and Control Association [35], is an IT governance framework that combines cybersecurity with organizational strategic objectives. It promotes effective technology and information management, aligning security with business processes.

OWASP (Open Web Application Security Project) [36] is an organization focused on software security, particularly in web applications and APIs, providing guidelines and tools for developers and businesses. Its most well-known project, the OWASP Top 10, lists the primary vulnerabilities in web applications,

such as SQL injection and authentication failures, helping prioritize mitigation actions.

These initiatives, each with its specificities, are complementary and can be combined to create robust cybersecurity strategies tailored to the needs and objectives of each organization. In the context of ontologies, they provide systematic frameworks that organize, standardize, and implement security concepts consistently and interoperable, facilitating the development and maintenance of ontologies.

3. Related Work

A literature mapping was conducted by analyzing 19 selected articles, following the principles and phases proposed by Kitchenham (2004) [37]. We explored well-known scientific databases, including Scopus, Web of Science, IEEE Xplore, SpringerLink, and Google Scholar. Our review identified key challenges and opportunities for using AI approaches, such as large language models (LLM) and retrieval-augmented generation (RAG) to support ontology development. We analyze works addressing various aspects of the ontology development process, including studies focused specifically on vocabulary concept harmonization. Of the related works reviewed, 12 studies focused on building or improving ontologies using LLM during the initial phases of ontology engineering, such as specification and conceptualization.

LLMs are proven to be a promising approach for ontology learning and engineering, as they combine efficient extraction of structured knowledge from natural text with human collaboration for refinement and validation. Techniques for automatic discovery of taxonomic relationships and dynamic generation of ontological components using RAG have been proposed in recent studies. Giglou et al. (2023) [4] propose an approach that uses LLMs for Ontology Learning (OL). The authors investigate whether LLMs can automatically extract and structure knowledge from natural language text. Nine LLM families were evaluated for three main OL tasks: term typing, taxonomy discovery, and extraction of non-taxonomic relations. Doumanas et al. (2024) [38] present an LLM-enhanced ontology engineering (OE) approach, aiming to identify how OE tasks can be completed with LLM and human collaboration. LLMs are employed to generate domain ontologies for modeling Search and Rescue (SAR) missions in wildfire incidents. The authors analyze LLM capabilities to OE and evaluate the human-machine synergy to represent knowledge, focusing on the SAR domain. Toro et al. (2024) [39] present an ontology generation method employing LLM and RAG (DRAGON-AI) aiming at generating textual and logical ontology components. According to the authors, DRAGON-AI has high precision for relationship generation, but has slightly lower precision than from logic-based reasoning; evaluators with the highest level of confidence in a domain were better able to discern flaws in AI-generated definitions. Mateiu and Groza (2023) [40] enrich ontologies by translating Natural Language (NL) into Description Logic. A GPT model is fine-tuned to convert NL into OWL. Pairs of sentences in NL and the corresponding translations for fine-tuning are designed. The training pairs cover aspects of ontology engineering, such as instances, domain and range of relations, and object property relationships. The resulting axioms were used to enrich an ontology, supervised by human experts.

Abolhasani and Bran (2025) [41] propose the OntoKGen platform, which uses LLMs to extract ontologies from technical texts and create new branches of these ontologies through user interaction and validation to define concepts, relationships, and properties. Based on the confirmed ontology, the platform generates KG in an automated, interactive, and adaptive manner, reducing user intervention while allowing necessary adjustments. In future work, the authors propose to integrate the OntoKGen-generated KGs into RAG systems, enabling dynamic data manipulation via an interface. Although OntoKGen does not directly work with RAG systems, it represents an advance in the extraction and creation of ontologies using LLMs.

Vrolijk et al. (2023) [42] present an ontology learning system for the job market based on the ESCO ontology, which uses LLM combined with RAG techniques to extract, classify and relate mentions of skills and occupations from online job advertisements. The system proposes a three-layer architecture that integrates automatic processing and human interaction to keep the ontology updated, identifying new entities and relationships. The experiments indicate that the method improves performance in

extracting mentions, classifying relationships, discovering knowledge, and suggesting new entities for ontology extension.

Bran et al. (2025) [43] introduce the OntoRAG methodology, which combines LLM with ontologies to enhance knowledge generation in scientific domains, where the goal is to mitigate "hallucination" problems. The approach was tested on a benchmark in the Single Atom Catalysis (SAC) domain, showing its effectiveness in predicting synthesis procedures. The results indicate that OntoRAG outperforms traditional RAG methods, highlighting the potential of integrating ontologies as knowledge representation alongside LLM models. In addition, the authors present the OntoGen tool, which allows the automatic generation of ontologies from documents. The OntoGen process can be divided into three stages, namely: (a) vocabulary extraction; (b) category generation; and (c) taxonomy extraction. These stages facilitate the adaptation of the OntoRAG method when applied to new domains. Despite the results presented, the authors note that user supervision is still needed when creating ontologies.

LLMs have supported the expansion and enrichment of ontologies, demonstrating effectiveness in the automated generation of ontological components (e.g., competency questions and RDF mappings) and the structured extraction of knowledge from unstructured texts, with applications in diverse domains. Yang et al. (2024) [44] propose an LLM-based ontology expansion method. LLMs are used to formulate competency questions (CQs) and to extend the initial ontology. The authors created a knowledge graph for breast cancer treatment. Mukanova et al. (2024) [45] propose an LLM-powered NLP method for ontology enrichment. The authors aim to process natural language texts and extract data from the text that matches the semantics of an ontological model. LLM extracts data from a Web page and converts it into lists with information relevant to an ontology. The proposed method is implemented using the example of an ontological model that describes a geographical configuration. Val-Calvo et al. (2025) [46] use LLM to aid in the development of ontology from data sets, increasing automation of ontology-based KG generation. The authors developed an LLM method to enhance ontology engineering through data pre-processing, ontology planning, building, and entity improvement. The proposed method can generate mappings and RDF data, but the authors focus on ontologies.

LLMs in conjunction with semi-automated approaches can support KG and ontology engineering, e.g., formulating CQs, developing or evaluating KGs with lower human intervention, and enabling conversational frameworks for eliciting requirements in ontologies. Kommineni et al. (2024) [47] present an LLM-supported approach for semi-automatically building an ontology and KG. The proposed approach involves: i) formulating competency questions (CQs); ii) developing an ontology (TBox) based on these CQs; iii) constructing KGs using the developed ontology; and iv) evaluating the resultant KG with minimal to no involvement of human experts. To evaluate the answers generated via RAG and the KG concepts automatically extracted using LLMs, the authors designed a judge LLM that rates the generated content. Zhang et al. (2024) [48] present a framework for conversational ontology engineering (OntoChat), aimed at supporting requirement elicitation, analysis, and testing. OntoChat aids users in creating user stories and extracting competency questions. The authors replicated the engineering of the Music Meta Ontology and collected preliminary metrics on the effectiveness of each component from users.

Our approach (LAVOHA) is focused on defining concepts from cybersecurity vocabularies and ontologies. We harness LLM to analyze and harmonize concept definitions in natural language and propose term relationships of a security incident response glossary. LAVOHA supports ontologists in the specification and conceptualization phases of the ontology engineering process. LAVOHA shares similarities with other works, particularly in leveraging LLMs for ontology-related tasks. Unlike related works, our method employs LLMs to process natural language and extract structured knowledge, supporting ontology engineering. The focus on automating parts of the ontology development process aligns with [46], [38], and [47], which also aim to reduce human effort through LLM assistance. Similarly to [44] and [47], LAVOHA involves defining and refining concepts (in our case, cybersecurity terms) using LLM-generated insights. The emphasis on human-AI collaboration is another shared aspect, as seen in [38] and [48], where human expertise guides and validates the LLM outputs.

Although related work (e.g. [4], [45], [39], [40]) focuses on general ontology learning or enrichment, our approach is domain-specific, targeting cybersecurity vocabularies and incident response terminology.

Unlike [39] and [40], which use LLMs for the generation or translation of logical axioms, our method focuses on conceptual harmonization and proposal of relationships, supporting the early stages of the ontology engineering. [47] and [48] present automated evaluation mechanisms (e.g., judge LLM or conversational interfaces), whereas our approach prioritizes ontologist-guided refinement rather than full automation. This distinguishes our work from [47], which minimizes human involvement, and aligns more closely with the emphasis on human-machine synergy from [38]. Finally, [44] and [46] integrate ontologies with KGs, and our current scope is limited to glossary and ontology conceptualization, although future extensions could explore KG integration. Our approach shares foundational LLM-based strategies with other works, but distinguishes itself through its cybersecurity focus, conceptual harmonization goals, and balanced human-AI collaboration. Despite the large number of studies on the use of LLMs in supporting ontology development, the specific issue of concept harmonization has been little explored in the literature. Table 1 presents a comparative analysis of related work.

Table 1
Related Work

Ref.	Purpose	Input	Output	Application Domain
[4]	Extracting and structuring knowledge	NL text	Structured knowledge	Geographic and medical
[44]	Formulating CQ and extending ontology	Ontology	CQ and extended ontology	Breast cancer treatment
[45]	Processing natural language texts	NL text and ontological model	Information lists relevant to a domain ontology	Geographic
[46]	Automating ontology-based KG generation	Datasets and ontology	Ontology-based KG	Commercial activities
[38]	Identifying how OE tasks can be completed with LLM	NL text	Domain ontology	SAR missions in wildfire incidents
[39]	Generating textual and logical ontology components	NL text	Ontology components	Basic Science (e.g., Gene, Biological, Environment)
[40]	Translating NL into DL	NL text	OWL ontology	Family relations (e.g., father, sister, etc.)
[41]	Extraction and creation of ontologies and generation of knowledge graphs	Complex technical documents (text)	Ontology and KG	Reliability and Maintenance in semiconductor manufacturing equipment
[42]	Automatic learning and updating of ontologies	Job advertisements (text)	Ontology components	Labor market, job ontologies
[43]	Reduction of hallucinations with ontologies and automatic generation of ontologies	Responses generated with greater accuracy and relevance and ontology components	Accurate and relevant responses and ontology components	Single Atom Catalysts (SAC)
[47]	Semi-automatically building ontology and KG	CQs	Ontology and KG	DL methodologies
[48]	Supporting requirement elicitation, analysis, and testing	User stories	CQs	Music metadata
This Work	*Harnessing LLM to analyze and harmonize concept definitions in NL text*	*NL text*	*Glossary of a domain ontology*	*Cybersecurity Incident Response*

4. The LAVOHA Method

This section introduces *LAVOHA*, a simplified version of an Advanced RAG-inspired method (see subsection 2.2) designed to support vocabulary harmonization in the ontology creation process.

4.1. Conceptual Description

Given a term t to be incorporated into a vocabulary, together with $Q_t = \{q_1, q_2, \ldots, q_{|Q_t|}\}$[2], a set of *queried terms* (i.e. words related to the definition of term t), and a corpus $D = \{d_1, d_2, \ldots, d_{|D|}\}$ consisting of relevant documents within the target domain, *LAVOHA* extracts relevant sentences from

[2]In this article, $|X|$ denotes the cardinality of any arbitrary set X

documents in D and uses them to query an LLM for suggested definitions for t. It is important to highlight that if t is a single word term then $t \in Q_t$, so that the algorithm will return the sentences related to t itself. In case t is a compound name, then each word in t must be in Q_t. Figure 2 presents a modular overview of the designed method.

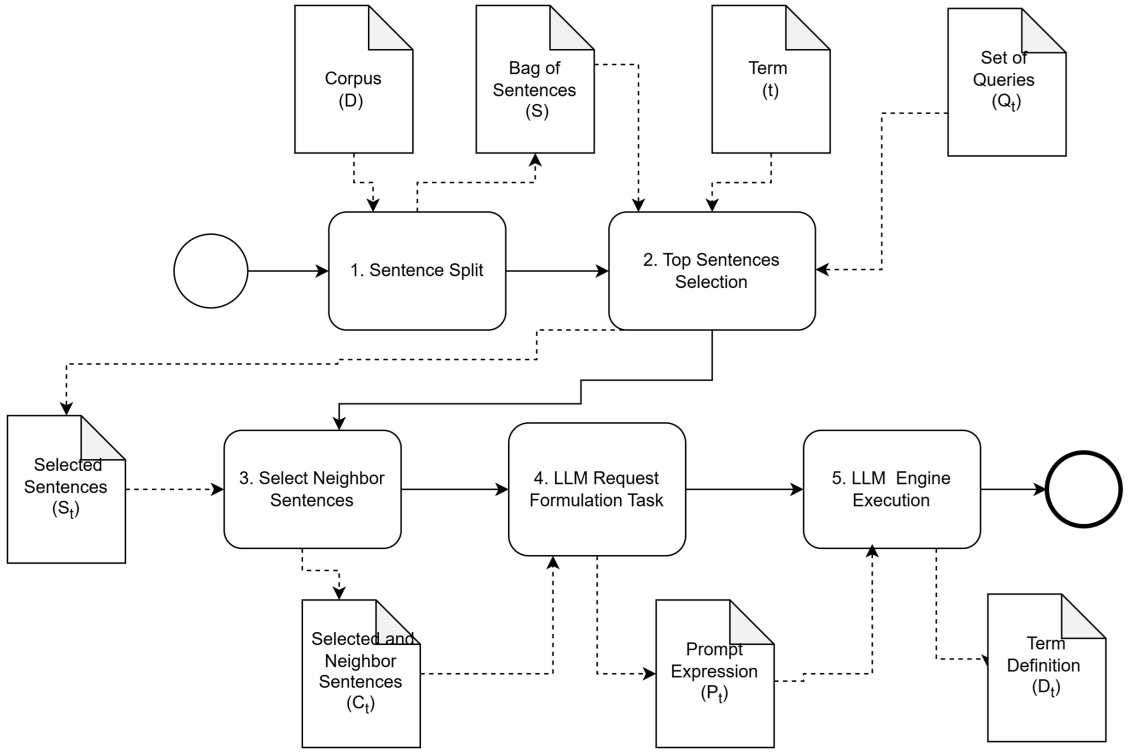

Figure 2: Vocabulary Harmonization Process - Modular Overview

Step 1 (Sentence Split) splits each document $d \in D$ into sentences, yielding a *Bag of Sentences* S, the set of all sentences in D. In *Step 2 (Top Sentences Selection)*, given S, t and Q_t, this step evaluates a relevance score for each pair $(q, s) \in Q_t \times S$. For each $q \in Q_t$, the top N sentences are returned, resulting in a set S_t of up to $|Q| \times N$ distinct sentences. Then, for each sentence $s_t \in S_t$, *Step 3 (Select Neighbor Sentences)* retrieves the M sentences that precede s_t and the M sentences that succeed s_t in d_{s_t}. The output of this step is, therefore, a set C_t of up to $|S_t| \times (2M + 1)$ distinct sentences, since the original sentence in S_t is also included. The next step *(Step 4 - LLM Request Formulation Task)* constructs P_t, the prompt expression for an LLM L chosen by the user. To this end, the following strings are concatenated: *"You are a specialist in "; name-of-the-domain-of-the-application; ". "; "Define the term "; t; ", based on the definitions stated in the following set: "; C_t*. Note that *name-of-the-domain-of-the-application*, t, and S_t are variables and are not strings themselves. Indeed, they contain the strings to be concatenated to form the prompt. Finally, *LAVOHA's* last step *(Step 5 - LLM Engine Execution)* consists of the execution of $L's$ engine, given the prompt P_t. Its output is D_t, the definition for t suggested by L.

4.2. Implementation Details

We provide further details on the implementation of the most challenging steps of the LAVOHA method. The LAVOHA implementation artifacts are available in the GitHub repository [3]. The system was developed in Python (version 3.12), due to its vast availability of useful packages and documentation online.

[3] https://github.com/anonymous_for_double_blind_revision

Pre-processing

Before Sentence Split, the pdf documents are converted to text files. This process is done only once per document, as a pre-processing step, since it is highly time-costly. Whenever a new document is added to the pool, it is quickly converted to a text file. To convert `pdf` files to `txt`, we make use of the `PyMuPdf`[4] in version 1.25.5.

Step 1 (Sentence Split)

In order to properly break the textual content of each file, we make use of the `nltk`[5] NLP package version 3.9.1, providing the appropriate language identifier depending on the document (`"english"` for documents in English, and `"portuguese"`, for Portuguese) in order to find and properly ignore the correct stopwords. The results presented in section 5 concern only the english version of the documents.

Step 2 (Top Sentences Selection)

The selection of the best sentences was made using the BM25 algorithm. Our choice of BM25 implementation is the package `rank-bm25`[6], in version 0.2.2.

The definitions of the BM25 parameters for each term are in a module (`queries.py`), containing a dictionary with information pertaining to the BM25 parameters for the 37 terms initially assigned to the glossary. This module centralizes and makes it easier to change and adapt BM25 parameters should the result of a BM25 call be unsatisfactory - this happens mostly in situations where not enough relevant sentences are selected, or when many irrelevant sentences are selected via the algorithm. The number of sentences retrieved in this module can be set in a property file.

Step 3 (Neighbor Sentences Selection)

The number of neighbors recovered in this step can be adjusted via a property file. In the experiment reported in the present paper, the number of neighbors was set to 0 (no neighbors), since only in special cases does the context surrounding the selected sentences provide useful information. This evaluation should be conducted individually for each case.

5. Case Study

We are developing an ontology for the cybersecurity domain, following a 4-phase methodology: specification, conceptualization, formalization, and implementation. In this methodology, the glossary of terms should be drafted preliminarily during specification and finalized during conceptualization. During the conceptualization phase, we faced the difficulty of reconciling different definitions of terms originating from documents assigned as knowledge sources. At this point, we created the LAVOHA method to support this task. This section presents the results of the LAVOHA method applied to this difficulty.

5.1. Experiment Configuration

As the document corpus D for the cybersecurity domain, we used the following set of documents:

- ATTACK_Design_and_Philosophy_March_2020.pdf
- getting-started-with-attack-october-2019.pdf
- mitre_TTPs.pdf
- NBRISO-IEC 27035.pdf

[4] https://pypi.org/project/PyMuPDF/
[5] https://www.nltk.org/
[6] https://pypi.org/project/rank-bm25/

- NIST.SP.800-61r2.pdf

The following list shows each term t used in the experiment, together with its set of related words (queries) Q_t. It is worth recalling that the terms and their respective related words, defined by the user, are precisely the concepts that require harmonization, since the meanings may vary in each document.

- Attack: "attack", "attempt", "access", "damage", "interrupt", "malicious", "degrade", "destroy",
- Attack vector: "vector", "attack", "method", "methods", "technique", "techniques", "access"
- Campaign: "campaign", "grouping", "activities", "intrusion", "period", "targets", "objectives"
- Damage: "damage", "effect", "event", "incident", "occurrence", "loss"
- Event: : "event", "occurrence", "observable", "indication", "incident", "suspicion", "adverse"
- Incident: "incident", "occurrence", "risk", "confidentiality", "integrity", "availability", "information", "violation", "threat", "policies"
- Information Asset: "asset", "information", "value", "person", "organization", "organisation", "medium", "resource", "critical"

The configuration properties used in the experiment were the following:

- N = 6. Step 2 selects the top six sentences.
- M = 0. Step 3 adds no neighbors.
- $L \in \{$*GPT-4o, DeepSeek-V3*$\}$. In Step 6 we used both GPT-4o and DeepSeek-V3.

5.2. Results

To assess whether LAVOHA indeed improved the LLM output, we generated the output (reconciled definitions) in three different ways:

- Using only the LLM, without LAVOHA;
- Using the LLM with LAVOHA;
- Through human discussions until reaching a consensual definition.

The human consensual definition was considered the appropriate response, and it was later compared to the first two definitions to evaluate whether the use of LAVOHA improved the LLM response for this task. To achieve this, we calculated the sentence embedding as the average of its word vectors, the cosine similarity between the responses generated by LLMs, and the consensus definition. The definition embeddings were computed with the pre-trained FastText library [49, 50].

Table 2 shows the mean value and standard deviation of the cosine similarity over the seven terms' definition. We note that with the use of LAVOHA, the quality of GPT4-o output improved from a 0.9261 mean similarity with the consensus definition to 0.9317, whereas DeepSeek improved from 0.9279 to 0.9443. Therefore, the DeepSeek model responded better with the help of LAVOHA. This result may be due to DeepSeek having a smaller training set than GPT4-o.

Table 2
LLM's definitions mean similarity to consensual definitions

	Mean	**SD**
GPT-4o	0.9261	0.0516
DeepSeek-V3	0.9279	0.0242
GPT-4o+LAVOHA	0.9317	0.0519
DeepSeek-V3+LAVOHA	0.9443	0.0364

We performed the automated evaluation because it is difficult to analyze manually the nuances that differentiate the two definitions. Still, it may be useful to present definitions produced with and without LAVOHA to illustrate the quality improvement achieved by the method. As an example, we list two definitions of attack: the first was generated by DeepSeek-V3 alone, and the second by DeepSeek-V3 assisted by LAVOHA:

1. A deliberate, malicious attempt by an individual, group, or system to exploit vulnerabilities in a computer system, network, or digital infrastructure with the intent to: - Compromise confidentiality, integrity, or availability (CIA triad) of data or services. - Gain unauthorized access, disrupt operations, steal information, or cause harm. - Deploy malware, execute code, manipulate systems, or escalate privileges. Attacks can be active (directly altering or damaging systems) or passive (eavesdropping without modification). Common types include phishing, ransomware, DDoS, SQL injection, and zero-day exploits.
2. A deliberate, malicious attempt by an adversary to compromise, disrupt, or destroy the confidentiality, integrity, or availability of systems, networks, or data. Attacks may employ various tactics, techniques, and procedures (TTPs), such as exploiting vulnerabilities, deploying malware, or leveraging social engineering, to achieve objectives like unauthorized access, data theft, service disruption (e.g., Denial of Service), or system destruction. These actions can target technical infrastructure (e.g., endpoints, cloud resources) or human elements (e.g., phishing), and often mimic normal activity to evade detection.

The second definition feels cleaner and more precise. In the first sentence, it defines attack, in the second, it demonstrates how attacks may be performed, and in the third sentence, it enumerates the possible targets of an attack. Furthermore, it uses the verb "target", an important predicate, because it characterizes the relation between a typical attack and the attacked assets, suggesting a possible triple modeling (Attack, targets, Asset). Then it identifies "technical infrastructure" as a target, a general term that encompasses the vulnerable entities listed in the first definition. It also includes "human elements", which is a conceptual gap in the first definition.

The other six term definitions generated by DeepSeek and the seven term definitions generated by GPT presented similar improvement with the aid of LAVOHA. Furthermore, DeepSeek answers for the seven terms without LAVOHA seem to vary little, as if reading from the same source. For instance, DeepSeek mentions "confidentiality, integrity, or availability (CIA triad)" in 4 of the 7 definitions without LAVOHA. However, when assisted by LAVOHA, it mentions the CIA triad in only one of the definitions, which shows a better separation of concepts by the different definitions.

6. Conclusion

We introduced LAVOHA, a method designed to harmonize conflicting concept definitions, specifically applied within cybersecurity vocabularies and ontologies. The approach leverages natural language analysis to produce unified consensus-based definitions of security concepts, as shown through a case study that compares LAVOHA-generated definitions with human consensus. The results favor LAVOHA over relying solely on large language models (LLMs), showing improved suitability.

As future work, we intend to extend our approach to other phases of Ontology Engineering, investigating how LLMs can assist in writing an ontology and whether LAVOHA-like methods can enhance their performance. Regarding the focus of the present work, namely vocabulary harmonization, future research could explore additional quantitative evaluation methods, such as measuring the perplexity of the consensus definition when submitted to the LLM in different scenarios. We expect that better-equipped LLMs (possibly enhanced by LAVOHA) will exhibit lower perplexity scores for the consensual definition. It would also be worthwhile to test alternatives to the BM25 algorithm for sentence selection, such as using LLM-based embeddings to represent queries and candidate sentences in Step 2 of LAVOHA.

Acknowledgments

This work was supported by the Brazilian Funding Authority for Studies and Projects (FINEP) trough the projects [CyberSemantics - Contract No. 0.1.22.0335.00/Ref. FINEP No. 0172/22] and [S2C2 - Contract No. 0.1.20.0272.00/Ref. FINEP No. 2904/20] and the Systems Development Center of the Brazilian Army. E.H.Haeusler was partially suported by CNPq grant 309287/2023-5.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o for grammar and spelling checks and text translations. After using this tool, the authors reviewed and edited the content as needed and assume full responsibility for the content of the publication.

References

[1] R. de Almeida Falbo, Sabio: Systematic approach for building ontologies, in: G. Guizzardi, O. Pastor, Y. Wand, S. de Cesare, F. Gailly, M. Lycett, C. Partridge (Eds.), Proceedings of the 1st Joint Workshop ONTO.COM / ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering co-located with 8th International Conference on Formal Ontology in Information Systems, ONTO.COM/ODISE@FOIS 2014, Rio de Janeiro, Brazil, September 21, 2014, volume 1301 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2014. URL: https://ceur-ws.org/Vol-1301/ontocomodise2014_2.pdf.

[2] M. C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The neon methodology framework: A scenario-based methodology for ontology development, Applied Ontology 10 (2015) 107–145. URL: https://journals.sagepub.com/doi/abs/10.3233/AO-150145. doi:10.3233/AO-150145.

[3] N. J. Mariano Fernández-López, Asunción Gómez-Pérez, Methontology: From ontological art towards ontological engineering, Miscellaneous (1997).

[4] H. Babaei Giglou, J. D'Souza, S. Auer, Llms4ol: Large language models for ontology learning, in: International Semantic Web Conference, Springer, 2023, pp. 408–427.

[5] A. De Nicola, M. Missikoff, A lightweight methodology for rapid ontology engineering, Commun. ACM 59 (2016) 79–86. URL: https://doi.org/10.1145/2818359. doi:10.1145/2818359.

[6] P. M. L. Scheidegger, M. L. M. Campos, M. C. Cavalcanti, An approach for systematic definitions construction based on ontological analysis, in: E. Garoufallou, S. Virkus, R. Siatri, D. Koutsomiha (Eds.), Metadata and Semantic Research - 11th International Conference, MTSR 2017 Tallinn, Estonia, November 28 - December 1, 2017, Proceedings, volume 755 of *Communications in Computer and Information Science*, Springer, 2017, pp. 87–99. URL: https://doi.org/10.1007/978-3-319-70863-8_9. doi:10.1007/978-3-319-70863-8_9.

[7] Amazon Web Services, What is RAG? - Retrieval-Augmented Generation explained, https://aws.amazon.com/what-is/retrieval-augmented-generation/, 2023. Acessado em 10 de outubro de 2025.

[8] IBM Research, What is retrieval-augmented generation (RAG)?, https://research.ibm.com/blog/retrieval-augmented-generation-RAG, 2023. Acessado em 10 de outubro de 2025.

[9] Intel, What is RAG? Retrieval-Augmented Generation explained, https://www.intel.com/content/www/us/en/learn/what-is-rag.html, 2025. Acessado em 10 de outubro de 2025.

[10] Elastic, What is retrieval-augmented generation?, https://www.elastic.co/what-is/retrieval-augmented-generation, 2025. Acessado em 10 de outubro de 2025.

[11] Oracle, What is Retrieval-Augmented Generation (RAG)?, https://www.oracle.com/artificial-intelligence/generative-ai/retrieval-augmented-generation-rag/, 2023. Acessado em 10 de outubro de 2025.

[12] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates, Inc., 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

[13] Y. Gao, et al., Retrieval-augmented generation for large language models: A survey, https://arxiv.org/abs/2312.10997, 2024. Acessado em 10 de outubro de 2025.

[14] IBM, What are RAG techniques?, https://www.ibm.com/think/topics/rag-techniques, 2025. Acessado em 10 de outubro de 2025.

[15] Weka, What is Retrieval-Augmented Generation (RAG)?, https://www.weka.io/learn/guide/ai-ml/retrieval-augmented-generation/, 2024. Acessado em 10 de outubro de 2025.
[16] S. Homayoun, 6 types of Retrieval-Augmented Generation (RAG) techniques you should know, https://homayounsrp.medium.com/6-types-of-retrieval-augmented-generation-rag-techniques-you-should-know-b45de9071c79, 2024. Acessado em 10 de outubro de 2025.
[17] A. Singhal, Modern information retrieval: A brief overview, IEEE Data Eng. Bull. 24 (2001) 35–43.
[18] K. Hambarde, H. Proença, Information retrieval: Recent advances and beyond, IEEE Access 11 (2023) 76581–76604. doi:10.1109/ACCESS.2023.3295776.
[19] A. Neelima, S. Mehrotra, A comprehensive review on word embedding techniques, in: 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), 2023, pp. 538–543. doi:10.1109/ICISCoIS56541.2023.10100347.
[20] L. Gutiérrez, B. Keith Norambuena, A Systematic Literature Review on Word Embeddings: Proceedings of the 7th International Conference on Software Process Improvement (CIMPS 2018), 2019, pp. 132–141. doi:10.1007/978-3-030-01171-0_12.
[21] A. de Vries, A. Wilschut, On the integration of ir and databases, in: Database issues in multimedia, 1999, pp. 16–31.
[22] G. Salton, C. Buckley, Term Weighting Approaches in Automatic Text Retrieval, Technical Report, USA, 1987.
[23] R. A. Shahzad Qaiser, Text mining: Use of tf-idf to examine the relevance of words to documents, International Journal of Computer Applications 181 (2018) 25–29. URL: https://ijcaonline.org/archives/volume181/number1/29681-2018917395/. doi:10.5120/ijca2018917395.
[24] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at TREC-3, in: D. K. Harman (Ed.), Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994, volume 500-225 of *NIST Special Publication*, National Institute of Standards and Technology (NIST), 1994, pp. 109–126. URL: http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.
[25] X. H. Lù, Bm25s: Orders of magnitude faster lexical search via eager sparse scoring, 2024. URL: https://arxiv.org/abs/2407.03618. arXiv:2407.03618.
[26] A. Trotman, A. Puurula, B. Burgess, Improvements to bm25 and language models examined, in: Proceedings of the 19th Australasian Document Computing Symposium, ADCS '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 58–65. URL: https://doi.org/10.1145/2682862.2682863. doi:10.1145/2682862.2682863.
[27] M. Taylor, H. Zaragoza, N. Craswell, S. Robertson, C. Burges, Optimisation methods for ranking functions with multiple parameters, in: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM '06, Association for Computing Machinery, New York, NY, USA, 2006, p. 585–593. URL: https://doi.org/10.1145/1183614.1183698. doi:10.1145/1183614.1183698.
[28] National Institute of Standards and Technology, Security and Privacy Controls for Information Systems and Organizations: NIST Special Publication 800-53, Revision 5, Technical Report, National Institute of Standards and Technology, Gaithersburg, 2020.
[29] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, C. B. Thomas, MITRE ATT&CK: Design and Philosophy, Technical Report, MITRE Corporation, 2020. URL: https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf.
[30] P. E. Kaloroumakis, M. J. Smith, Toward a Knowledge Graph of Cybersecurity Countermeasures, Technical Report, MITRE Corporation, 2023. URL: https://d3fend.mitre.org/resources/D3FEND.pdf.
[31] National Institute of Standards and Technology, Cybersecurity framework, 2025. Available at: https://www.nist.gov/cyberframework, accessed on May 28, 2025.
[32] Computer Security Resource Center, Glossary, 2025. URL: https://csrc.nist.gov/glossary.
[33] International Organization for Standardization, Iso/iec 27001:2022 – information security, cybersecurity and privacy protection, 2022. Available at: https://www.iso.org/standard/27001, accessed on May 28, 2025.

[34] Center for Internet Security, Cis critical security controls, 2025. Available at: https://www.cisecurity.org/controls, accessed on May 29, 2025.
[35] ISACA, Cobit 2019 framework: Introduction and methodology, 2019. Available at: https://www.isaca.org/resources/cobit, accessed on May 29, 2025.
[36] OWASP, Owasp top ten, 2025. Available at: https://owasp.org/www-project-top-ten/, accessed on May 29, 2025.
[37] B. Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele University 33 (2004) 1–26.
[38] D. Doumanas, A. Soularidis, K. Kotis, G. Vouros, Integrating llms in the engineering of a sar ontology, in: IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer, 2024, pp. 360–374.
[39] S. Toro, A. V. Anagnostopoulos, S. M. Bello, K. Blumberg, R. Cameron, L. Carmody, A. D. Diehl, D. M. Dooley, W. D. Duncan, P. Fey, et al., Dynamic retrieval augmented generation of ontologies using artificial intelligence (dragon-ai), Journal of Biomedical Semantics 15 (2024) 19.
[40] P. Mateiu, A. Groza, Ontology engineering with large language models, in: 2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, 2023, pp. 226–229.
[41] M. S. Abolhasani, R. Pan, Ontokgen: A genuine ontology and knowledge graph generator using large language model, in: 2025 Annual Reliability and Maintainability Symposium (RAMS), IEEE, 2025, pp. 1–6.
[42] J. Vrolijk, V. Poslavsky, T. Bijl, M. Popov, R. Mahdavi, M. Shokri, Ontology learning for esco: Leveraging llms to navigate labor dynamics, Proceedings of the 2nd workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM 2024) (2023).
[43] A. M. Bran, A. Oarga, M. Hart, M. Lederbauer, P. Schwaller, Ontology-retrieval augmented generation for scientific discovery, Under review as a conference paper at ICLR 2025 (2025).
[44] H. Yang, L. Xiao, R. Zhu, Z. Liu, J. Chen, An llm supported approach to ontology and knowledge graph construction, in: 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2024, pp. 5240–5246.
[45] A. Mukanova, M. Milosz, A. Dauletkaliyeva, A. Nazyrova, G. Yelibayeva, D. Kuzin, L. Kussepova, Llm-powered natural language text processing for ontology enrichment (2024).
[46] M. Val-Calvo, M. E. Aranguren, J. Mulero-Hernández, G. Almagro-Hernández, P. Deshmukh, J. A. Bernabé-Díaz, P. Espinoza-Arias, J. L. Sánchez-Fernández, J. Mueller, J. T. Fernández-Breis, Ontogenix: Leveraging large language models for enhanced ontology engineering from datasets, Information Processing & Management 62 (2025) 104042.
[47] V. K. Kommineni, B. König-Ries, S. Samuel, From human experts to machines: An llm supported approach to ontology and knowledge graph construction, arXiv preprint arXiv:2403.08345 (2024).
[48] B. Zhang, V. A. Carriero, K. Schreiberhuber, S. Tsaneva, L. S. González, J. Kim, J. de Berardinis, Ontochat: a framework for conversational ontology engineering using language models, in: European Semantic Web Conference, Springer, 2024, pp. 102–121.
[49] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146. URL: https://aclanthology.org/Q17-1010/. doi:10.1162/tacl_a_00051.
[50] C. De Boom, S. Van Canneyt, S. Bohez, T. Demeester, B. Dhoedt, Learning semantic similarity for very short texts, in: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 2015, pp. 1229–1234. doi:10.1109/ICDMW.2015.86.