ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240639

Ontology Text Alignment: Aligning Textual Content to Terminological Axioms

Jieying Chen[a,b,*], Hang Dong[a,c], Jiaoyan Chen[a,d] and Ian Horrocks[a]

[a]University of Oxford, UK
[b]Vrije Universiteit Amsterdam, Netherlands
[c]University of Exeter, UK
[d]University of Manchester, UK
ORCID (Jieying Chen): https://orcid.org/0000-0002-2497-645X, ORCID (Hang Dong): https://orcid.org/0000-0001-6828-6891, ORCID (Jiaoyan Chen): https://orcid.org/0000-0003-4643-6750, ORCID (Ian Horrocks): https://orcid.org/0000-0002-2685-7462

Abstract. Despite the impressive advancements in Large Language Models (LLMs), their ability to perform reasoning and provide explainable outcomes remains a challenge, underscoring the continued relevance of ontologies in certain areas, particularly due to the reasoning and validation capabilities of ontologies. Ontology modelling and semantic search, due to their inherent complexity, still demand considerable human effort and expertise. Addressing this gap, our paper introduces the problem of ontology text alignment, which involves finding the most relevant axioms with respect to the given reference text. We propose an advanced Retrieval Augmented Generation framework that leverages BERT models and generative LLMs, together with ontology semantic enhancement based on atomic decomposition. Additionally, we have developed benchmarks in geology and biomedical areas. Our evaluation demonstrates the positive impact of our framework.

1 Introduction

In the ever-evolving landscape of artificial intelligence, the development of ontologies represents a frontier of both opportunity and challenge. Particularly in domains where generative LLMs have demonstrated significant success, the role of these models in ontology construction is becoming increasingly pertinent. Despite their effectiveness in generating knowledge graphs, LLMs encounter substantial difficulties when grappling with the complexities of ontology development. Ontologies, set apart from the relatively simpler structure of knowledge graphs composed of subject-predicate-object triples, are characterized by their elaborate configurations and intricate semantic elements, including but not limited to negation, existential restrictions, and universal restrictions. This complexity introduces a notable disparity in the application of LLMs for ontology creation, with considerable reliance still placed on human expertise and effort.

The construction of a precise and meaningful ontology is not a trivial endeavor. It necessitates not only in-depth domain-specific knowledge but also a comprehensive understanding of formal logic to accurately interpret and represent complex semantic concepts. The task becomes increasingly arduous when dealing with intricate elements such as negation and universal restrictions, which are pivotal to the ontology's integrity yet challenging to master. In the current market, there is a obvious demand for skilled ontologists capable of crafting such detailed and accurate ontologies, but the availability of such expertise is limited.

An additional challenge in real-world ontology applications is their sheer size. Many ontologies are vast and complex, making them too cumbersome for human users to fully comprehend and navigate. This adds another layer of difficulty, as the management and understanding of these large-scale ontologies become increasingly daunting for even the most skilled ontologists.

To address these challenges, several solutions like ontology modularity [33, 12, 17] and ontology alignment [10] have been put forth. Ontology modularity, in particular, involves creating subsets within an ontology that align with specific human interests, enabling targeted functionality for diverse applications. While these modular approaches maintain logical consistency in reasoning, they are constrained by the need for precise specifications and often struggle with ambiguous or incomplete information. For example, a module that includes *soymilk* but overlooks *soy* can only provide limited insights. A significant challenge lies in assisting users to define these specifications, a task that existing tools are yet to properly manage.

Another notable limitation within ontology modularity is the difficulty in constraining the size of ontology modules [3]. Adjustments in the *signature*, a set of entities that users are interested in, can influence the module's size, but without any definitive guarantees. The concept of *ontology excerpts* [3] has emerged as a potential solution to this problem. These are constrained ontology modules containing a maximum number of axioms. However, even this approach does not fully address the challenge of signature selection and its applicability remains narrow.

In real-world scenarios, ontologists frequently rely on academic literature and online resources, summarizing and assimilating relevant knowledge into text, which they then convert into an ontology format. This process also involves identifying relevant axioms in both the current and similar ontologies. Drawing inspiration from these practical methodologies, our paper introduces a novel problem: ontology text alignment. This innovative problem aims to extract the most relevant axioms that correspond to the given reference text, pre-

* Corresponding Author. Email: j.y.chen@vu.nl

senting a unique solution to bridge the gap between theoretical ontology development and practical application. The problem of ontology text alignment has the following applications:

Ontology Modeling: For ontology modelling, the ontology text alignment serves as a powerful tool for ontologists. By leveraging advanced LLMs like GPT models, they can efficiently summarize extensive literature into coherent texts that forms the basis for ontology construction. The ontology text alignment technique enables the identification of relevant axioms from existing ontologies, thereby reducing the need to model every axiom from scratch and facilitating the development of comprehensive and relevant ontologies.

Ontology Selection: The ontology text alignment technique also plays a critical role in ontology selection. Faced with an array of ontologies, determining the most suitable one for a particular use case can be overwhelming. The extractive summary method can generate concise overviews of various ontologies based on specific texts, such as research queries or project requirements. This not only aids in understanding the core aspects of each ontology but also helps in identifying the most relevant ontology, streamlining decision-making processes in extensive ontology databases.

Semantic Search Enhancement: In semantic search engines, ontology text alignment can enhance search accuracy and relevance. By extracting key axioms aligned with search queries, they offer more precise search results, catering to users' specific information needs.

Through these applications, ontology text alignment emerges as a transformative tool in ontology development, offering efficient and precise solutions that bridge theoretical concepts with practical implementations. The contributions of our paper are as follows:

1. We introduce the ontology text alignment problem, a novel issue that arises from the practical needs of ontology modelling in real-world applications.
2. We propose an innovative framework that leverages the BERT model alongside LLMs to effectively address this challenge. This framework also enhances axiom ranking by incorporating structural and logical semantic insights derived from the syntactical structure and atomic decomposition of the ontology.
3. We develop three unique benchmark datasets tailored to different domains and scales. In particular, we employ justifications to construct semantically accurate benchmark sets, Anatomy.
4. Our evaluation highlights the significant benefits of our framework, particularly emphasizing the role of additional ontology semantic enhancement. This enhancement substantially improves the framework of ontology text alignment.

2 Related Work

BERT and LLMs. BERT (Bidirectional Encoder Representations from Transformers) is a sophisticated model that leverages deep bidirectional transformers for generating contextualized word embeddings [7]. It is typically pre-trained on a vast corpus of general-purpose text, utilizing a dual-task framework to enhance its language understanding capabilities. The architecture of BERT incorporates a [CLS] token at the start of each sequence for classification tasks, and a [SEP] token to delineate the end of each sentence within the pair. During pre-training, BERT learns the contextual relationships between words through two self-supervised tasks: Next Sentence Prediction (NSP) and Masked Language Modeling (MLM). NSP involves predicting whether sentence B logically follows sentence A based on the representation derived from the [CLS] token, while MLM focuses on predicting the identity of randomly masked tokens within both sentences A and B. Since the introduction of BERT, the model has garnered significant interest within the research community. Numerous variants of BERT have been developed, each fine-tuned for specialized domains and diverse downstream tasks. Notable adaptations include Sentence-BERT (SBERT) [26], optimized for sentence embeddings to enhance semantic textual similarity assessments; and SapBERT [19], which focuses on improving domain-specific entity linking tasks through pre-training using an integrated medical ontology.

Recent LLMs (like GPT and LLaMA series) generally follow the Transformer architecture of BERT, but are using the decoder, instead of the encoder of the Transformer. LLMs have significantly greater parameter sizes and are trained with unprecedented amounts of data on the Web. The field of LLMs is fast-moving and we refer readers to the recent survey of LLMs with their relation to KGs [23]. We examine the BERT variations and LLMs for ontology text alignment.

Ontology Modularity. Ontology modularity has significantly advanced to tackle the challenges of reusability and interoperability across various systems, a topic well-covered in the scholarly literature [33, 12, 17, 16, 5, 22, 18, 4, 2]. Several module notions have been proposed, including deductive modules [18], semantic modules [15], and locality-based modules [28]. However, those modules are often computationally expensive or excessively large. For locality-based modules, the total number of potential modules within an ontology can grow exponentially with respect to the number of terms or axioms it contains [12]. To manage this complexity, atom decomposition [35] has been introduced as a succinct method to represent all possible modules of an ontology efficiently. It helps streamline the modularization process by providing a more manageable and computationally feasible framework for handling large ontologies.

Entity Linking. Entity linking is the task of matching a mention in the text to an entity or class in the ontology [30]. Entity linking can support ontology construction and enrichment by associating text mentions to existing classes or identifying new classes in an ontology [9, 8]. The input mentions can come from texts in corpora or tables: for texts in corpora, a step of Named Entity Recognition (NER) or mention detection may be needed to identify the mentions before their linkage to an ontology; alternatively, a joint, end-to-end NER and entity linking locates the span of mention together with the class [30, 29]. Traditional entity linking methods are rule-based (by string matching and rules) and feature-based (using lexical features), and recent methods leverage deep learning, especially embedding and BERT-based methods [30, 31, 29]. The usage of generative LLMs for entity linking remains an open question, with still scarce research, e.g., on biomedical ontologies [36]. The task of ontology text alignment is more complex than entity linking, as the former locates a subset of axioms which usually have more entities with their relations in the ontology, and need to consider a paragraph of texts instead of one or few mentions each time in entity linking.

3 Problem Statement

Consider a scenario where an ontologist aims to incorporate the concept 'Abnormal blood oxygen pressure' into an existing ontology. Rather than building the representation from scratch, the ontologist seeks to leverage existing biomedical ontologies for any predefined notions. Upon locating the desired definition in textual format, as shown below, the ontologist would employ the ontology text alignment process to identify relevant axioms in the SNOMED CT ontology. An example of such text that requires alignment is as follows:

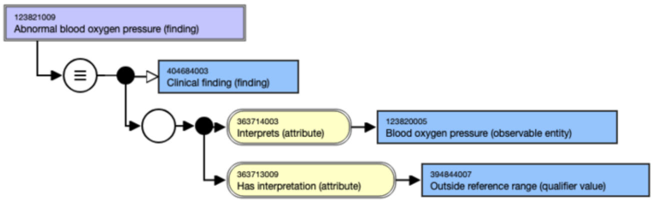

Figure 1: Illustrative example of the most relevant axiom diagram related to the reference text, presented through the SNOMED CT browser interface.

> "Abnormal blood oxygen pressure is a medical condition that reflects a deviation from the normative range of oxygen partial pressure in the bloodstream, with 'abnormal' indicating a measurement outside the accepted reference values used by medical practitioners for defining physiological health."

Unlike traditional named entity recognition and entity linking problems, our problem focuses on inferring the text's claims from axioms rather than simply identifying concept mentions. This adds complexity and increases the challenge. The diagram of the axiom can be visualized in Figure 1 using the SNOMED CT browser. Formally, the definition of ontology text alignment is defined as follows:

Definition 1 (Ontology Text Alignment) *Let $\mathcal{O}$ be an ontology, $\mathcal{E} \subseteq \mathcal{O}$, ref be the reference text, and let $k \in \mathbb{Z}^+$. Additionally, let $\mu : (\mathcal{O}, \mathcal{E}, \mathsf{ref}) \mapsto \mathbb{R}_{\geq 0}$ be a relevance measure function. The task* ontology text alignment *w.r.t. ref under μ is to identify an $\mathcal{E} \subseteq \mathcal{O}$ satisfying the following condition:*

$$\mu(\mathcal{O}, \mathcal{E}, \mathsf{ref}) = \max\{\, \mu(\mathcal{O}, \mathcal{E}', \mathsf{ref}) \mid \mathcal{E}' \subseteq \mathcal{O}, |\mathcal{E}'| \leq k \,\}.$$

The objective is to identify a subontology that achieves the highest relevance score w.r.t. the given reference text, subject to the size constraint k. Note that our paper primarily focuses on terminological axioms (TBox axioms) characterized by their rich semantics and complex structures. We do not consider factual assertions, namely, class/role assertions (ABox axioms).

4 Framework

We tackle this problem by identifying the most relevant, top-ranked axioms, inspired by recent advancements in generative LLMs and the Retrieval-Augmented Generation framework. Our approach includes the following components, based on these key considerations:

Verbalization. Complex axioms often contain complex logical operations like conjunctions and existential restrictions, and classes are encoded as URLs, making them challenging and not inherently readable by current LLMs. Through verbalization, we transform axioms into descriptive natural language texts.

Indexing. Handling large ontologies, like SNOMED CT with more than 300K axioms, is challenging for current generative LLMs due to their limitations with long texts. We address this by employing pre-trained models like BERT for initial indexing.

Axiom Type	Verbalization
$C \sqsubseteq D$	"Every $\mathbb{V}(C)$ is a $\mathbb{V}(D)$."
$C \equiv D$	"$\mathbb{V}(C)$ is defined as $\mathbb{V}(D)$."
$C \sqsubseteq \exists r.\top$	"$\mathbb{V}(C)$ is the domain of the property $\mathbb{V}(r)$."
$\top \sqsubseteq \forall r.D$	"$\mathbb{V}(D)$ is the range of the property $\mathbb{V}(r)$."
$C \sqcap D \sqsubseteq \bot$	"$\mathbb{V}(C)$ and $\mathbb{V}(D)$ are disjoint."

Table 1: Verbalization Rule for different axiom types.

Integration of generative LLMs. We aim to improve the accuracy and relevance of indexing by integrating semantically rich ontology graphs into LLM prompts. This enriches the indexing process with contextually relevant knowledge, enhancing the understanding and use of ontological structures.

The subsequent sections will elaborate on each process and describe the methodologies and technologies applied.

4.1 Ontology Axiom Verbalization

We leverage recent advancements in the field of natural language processing to bridge the gap between complex ontology knowledge and understandable language. This process, known as ontology verbalization, involves translating ontological axioms into text that is easily comprehensible by individuals without specialized knowledge in the field. The main challenge in this process is to preserve the original meanings of the axioms while making them more accessible. We use natural language processing techniques to decode and rephrase the often intricate and abstract structures of ontology axioms. Although existing tools are available in this area, they often lack maintenance.

In our approach, we start by verbalizing each concept name using their RDFS labels. The function to verbalize a logical formula f is denoted as $\mathbb{V}(f)$. We adopt methods from prior work [13] to verbalize complex concepts C, represented as $\mathbb{V}(C)$. The rules for generating these verbalizations are outlined in [13].

For each axiom α, we further apply the rules listed in Table 1 to verbalize axioms of different types, denoted as $\mathbb{V}(\alpha)$. Notably, the axiom types $C \sqsubseteq \exists r.\top$ and $\top \sqsubseteq \forall r.D$ are specialized cases within the general category $C \sqsubseteq D$, representing the domain and range of the property r, respectively. Our study primarily focuses on terminological axioms, also known as TBox axioms.

4.2 Indexing: Axiom Text Embedding and Ranking

After the axioms are verbalized, they are processed through a language model for encoding, with the goal of representing each axiom sentence as a vector. Given the versatility and robustness demonstrated by BERT and its derivatives in capturing contextual relationships within text, we have selected BERT and its variants for the task of vectorizing ontology axiom sentences. This approach leverages the deep learning capabilities of BERT to enhance semantic understanding and representation in ontologies.

For a robust representation of sentence-level information, we employ a BERT model to vectorize an axiom sentence $\mathbb{V}(\alpha)$ for axiom α through a technique known as *mean pooling*. Initially, an axiom sentence $\mathbb{V}(\alpha)$ is processed through a specified BERT tokenizer, transforming $\mathbb{V}(\alpha)$ into a sequence of tokens. We denote this tokenization function of an axiom sentence $s = \mathbb{V}(\alpha)$ as $\mathbb{T}(s) = [t_1, t_2, \ldots, t_n]$, where t_i represents the i-th token. The tokens are subsequently passed through the BERT model, which outputs a sequence of embeddings $[\overrightarrow{t_{\text{CLS}}}, \overrightarrow{t_1}, \overrightarrow{t_2}, \ldots, \overrightarrow{t_n}, \overrightarrow{t_{\text{SEP}}}]$. Each embedding

$\overrightarrow{t_i}$ corresponds to its respective token t_i. To synthesize a single, fixed-size representation of the axiom sentence $\mathbb{V}(\alpha)$, mean pooling is applied over the embeddings corresponding to the original input tokens, excluding any special tokens such as [CLS] and [SEP]. In summary, the embedding of each axiom sentence and the reference text ref is calculated as:

$$\overrightarrow{\mathbb{V}(\alpha)} = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{t_i}, t_i \in \text{TOK}(\mathbb{V}(\alpha)) \setminus \{[\text{CLS}], [\text{SEP}]\},$$

$$\overrightarrow{\mathsf{ref}} = \frac{1}{n}\sum_{i=1}^{n}\overrightarrow{t_i}, t_i \in \text{TOK}(\mathbb{V}(\mathsf{ref})) \setminus \{[\text{CLS}], [\text{SEP}]\},$$

where TOK(t) denotes the tokenization function, which is utilized to decompose the text t into a sequence of discrete tokens t_i, and $n = |\text{TOK}(\mathbb{V}(\alpha)) \setminus \{[\text{CLS}], [\text{SEP}]\}|$.

By leveraging various BERT models, we can capture the nuanced semantic and syntactic features of the sentences, facilitating a more nuanced understanding of the underlying ontological concepts. After mean pooling, the resulting vector effectively encapsulates the semantic properties of the axiom sentence in a form conducive to downstream tasks such as text classification or similarity analysis.

Axiom Ranking. Once each axiom sentence is encoded, the next step is to compute the semantic similarity between these verbalized axiom sentences and the reference text. This process involves measuring the semantic closeness of the vectors representing the axiom sentences to the reference text vector. The sentences are subsequently ranked based on their similarity scores, where higher scores denote a greater alignment with the reference text.

Several methods exist for computing text similarity, including cosine similarity, Euclidean distance, Jaccard similarity, BM25, and BM25+ [27]. Cosine similarity measures the cosine of the angle between vectors, which effectively captures orientation rather than magnitude, making it suitable for high-dimensional text data. Euclidean distance, by contrast, measures the geometric distance between points and can be overly sensitive to vector magnitude. Jaccard similarity, often used for binary data, fails to effectively capture the nuanced semantic differences in textual data. BM25 and BM25+ are information retrieval functions that score documents based on the query terms appearing in each document, considering both term frequency and inverse document frequency. However, they may not perform optimally in scenarios where semantic richness and context from longer text segments need to be assessed, as they traditionally focus on keyword matching rather than contextual meaning.

Given these factors, we employ cosine similarity due to its robustness in handling the directional properties of semantic vectors, making it highly effective for the comparison of textual data where the semantic context is more significant than vector magnitude.

To assess the similarity between each axiom α and the reference text ref, we compute the cosine similarity, for the embedding of each axiom sentence $\mathbb{V}(\alpha)$ and the reference text as follows:

$$\text{SIM}(\alpha, \mathsf{ref}) = \frac{\overrightarrow{\mathbb{V}(\alpha)} \cdot \overrightarrow{\mathsf{ref}}}{\|\overrightarrow{\mathbb{V}(\alpha)}\|\|\overrightarrow{\mathsf{ref}}\|},$$

where $\vec{a} \cdot \vec{b}$ represents the dot product of $\vec{a}$, $\vec{b}$ and $\|\vec{a}\|$ represents the Euclidean norms of vector $\vec{a}$. Once the similarity between each axiom and the input reference text has been calculated, axioms can be ranked based on their values of similarity w.r.t. the reference text.

4.3 Semantic Enhancement

In earlier phases of axiom retrieval, the internal structure of the ontology and semantics of axioms were not taken into account. To address this limitation, we propose to enhance the top-k-ranked axioms by integrating the internal structure of the ontology. This enhancement further involves leveraging semantic values of the ontology axioms, which are enriched through methodologies developed within the domain of knowledge representation and reasoning. Consequently, we constructed ontology atomic graphs to refine the results obtained in previous axiom retrieval phases.

Before we construct ontology atomic graphs, we first introduce *Ontology Syntax Graphs*, which is designed to elucidate the direct syntactical relationships between axioms. Formally, the *Ontology Syntax Graph* is defined as follows:

$$G_S(\mathcal{O}) = \{(\alpha_i, \alpha_j) \mid \text{sig}(D_i) \cap \text{sig}(C_j) \neq \emptyset, \alpha_i, \alpha_j \in \mathcal{O}\},$$

where $\alpha_i := C_i \sqsubseteq D_i$, $\alpha_j := C_j \sqsubseteq D_j$ and $\text{sig}(C)$ is the function that retrieves the concept and property names of concept C. Each axiom is represented as a node. An edge is drawn between two nodes, axiom α_i and axiom α_j, both of which are of the form $C \sqsubseteq D$, if the symbols on the right-hand side of α_i intersect with those on the left-hand side of α_j. Note that each concept definition of the form $C \equiv D$ is logically transformed into the combination of $C \sqsubseteq D$ and $D \sqsubseteq C$ before constructing the Ontology Syntax Graphs.

Ontology Atomic Graph. To impose semantic relationships between axioms in the graph, we utilize a technique known as atom decomposition, developed within the field of knowledge representation and reasoning. An atom [35, 20] of an ontology, defined under the framework of locality-based modules [28], is a subset of axioms that exhibits either full inclusion or non-intersection with any given module. In our paper, we opted for star modules due to their relatively smaller size compared to bottom- or top-local modules, while they still preserve essential logical consequences and can be computed with high efficiency. Formally, a set of axioms $\mathcal{A}$ constitutes an atom of an ontology $\mathcal{O}$ if, for every locality-based module $\mathcal{M} \subseteq \mathcal{O}$, if $\mathcal{A} \subseteq \mathcal{M}$ or $\mathcal{A} \cap \mathcal{M} = \emptyset$ holds [35]. Additionally, for any atom $\mathcal{A}_1$ and $\mathcal{A}_2$ of an ontology $\mathcal{O}$, $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$ holds. This definition encapsulates the notion that an atom is a decomposable part of an ontology, invariant across all locality-based modules, ensuring that the axioms within a single atom always co-occur. Dependency between atoms is defined as follows: an atom $\mathcal{A}_1$ is said to depend on another atom $\mathcal{A}_2$, denoted $\mathcal{A}_1 \preccurlyeq \mathcal{A}_2$, if $\mathcal{A}_1 \subseteq \mathcal{M}$ implies $\mathcal{A}_2 \subseteq \mathcal{M}$ for every module $\mathcal{M}$, i.e., the inclusion of $\mathcal{A}_1$ in any module $\mathcal{M}$ necessarily implies the inclusion of $\mathcal{A}_2$ in $\mathcal{M}$. This relation captures the foundational dependencies among atoms within the ontology.

Based on the foundational notions described previously, we propose the construction of an ontology atomic graph, where edges are established between two axioms if they consistently co-occur within atoms or if a dependency relationship exists between them, as dictated by the ontology's syntactic structure. This construction facilitates the linking of axioms based on both syntactic structure and semantic relationships inherent in the ontology. Formally, the ontology atomic graph is defined as follows:

$$\begin{aligned} G_A(\mathcal{O}) = &\{(\alpha_i, \alpha_j) \in G_S(\mathcal{O}) \mid \exists \mathcal{A}, \text{ such that } \alpha_i, \alpha_j \in \mathcal{A}\} \\ &\cup \{(\alpha_i, \alpha_j) \in G_S(\mathcal{O}) \mid \exists \mathcal{A}_1, \mathcal{A}_2, \text{ such that } \alpha_i \in \mathcal{A}_1, \\ &\quad \alpha_j \in \mathcal{A}_2, \mathcal{A}_1 \preccurlyeq \mathcal{A}_2\}, \end{aligned}$$

where $\mathcal{A}, \mathcal{A}_1, \mathcal{A}_2$ are atoms of the given ontology $\mathcal{O}$.

Algorithm 1 Enhance Candidate List via Ontology Atomic Graph

procedure EXPANDCANDIDATELIST(L, k, ref, G_A)
  $n \leftarrow \lceil k/2 \rceil$
  $\mathcal{L} \leftarrow L[1:n]$
  while $|\mathcal{L}| < k$ **do**
    $T \leftarrow \emptyset$
    for $\alpha \in \mathcal{L}$ **do**
      for each β such that in (α, β) or $(\beta, \alpha) \in G_A$ **do**
        $T \leftarrow T \cup \{\beta\}$
    $\mathcal{L} \leftarrow \mathcal{L} \cup T$
  Sort $\mathcal{L}$ by SIM(α, ref) descending
  return $\mathcal{L}[1:k]$

Expansion of Candidate Lists Using Ontology Atomic Graphs. Once the ontology atomic graph has been established, it serves as a crucial tool for enhancing the top k axiom candidates derived from embedding phases. These candidates are further enriched through their integration with the structural and semantic insights provided by the ontology atomic graph. To systematically expand the initial set of candidates, we employ the Expand Candidate List algorithm, which effectively broadens the search space while maintaining relevance to the reference ontology.

As detailed in Algorithm 1, the input to the algorithm consists of a ranked list of axiom candidates, denoted by L, which is sorted by SIM($\mathbb{V}(\alpha), \mathsf{ref}$), where $\alpha \in \mathcal{O}$ and k represents the desired number of refined candidates to be returned. The goal of the algorithm is to enhance this list using the ontology atomic graph, G_A.

The process begins by selecting the top $\lceil k/2 \rceil$ candidates [1], chosen for their high similarity scores. The candidate list is then dynamically expanded by iteratively adding adjacent axiom nodes from the Ontology Atomic Graph. This expansion continues until the list $\mathcal{L}_k$ contains exactly k candidates. Each candidate is chosen based on direct connections within the graph, which ensures their relevance in terms of both semantic and structural attributes. The list is subsequently refined by re-sorting the candidates based on their similarity to the reference text, culminating in the retention of the top k candidates.[2]

4.4 Zero-Shot Prompting of LLMs

Upon obtaining the top k candidate axioms, we construct a structured prompt to guide LLMs in generating the most relevant axioms, refined from the candidate list by ontology atomic graphs. The design of this prompt comprises a clear task description, the reference text, and the enumerated top k axiom sentences. The template of the prompt based on the top k candidates $\mathcal{L}[1:k]$ and the input reference text ref is as follows:

```
### Input:
Could you please find the most relevant setences from the following
candidate sentences with respect to the reference sentence? Please provide
the ranking in the format:
"The ranking of candidates is: [ranked list of IDs]."
 Please answer briefly using candidate IDs, separated by commas.
Here is the reference sentence: {Reference text}
Candidates:
ID {ID of α_i}: - {V(α_i)}
...
(list until k candidates)
### Response:
```

This structured approach aims to harness the zero-shot learning capabilities of LLMs, where no additional fine-tuning or task-specific training data is provided. The instruction sequence is followed by a delineated marker ("### Response"), signaling the point at which the LLM's generated response is anticipated to commence.

5 Evaluation

5.1 Creation of Benchmark Datasets

To evaluate the performance of our framework, we developed three benchmark datasets across diverse domains: geology, food, and medicine. These datasets are based on extractive summaries derived from three specific ontologies: GeoFault [25], and two branches of the SNOMED CT [3] ontology that focuses on diseases and anatomy.

The datasets were constructed with two primary use cases in mind: ontology modeling and semantic search. For ontology modeling, we created GeoFault and Disease datasets. Initially, our methodology involved extracting definition annotations [4], which were previously added by ontologists during the modeling phase of each ontology. These annotations provided essential contextual foundations for our datasets. We identified and extracted entities related to these annotations along with their corresponding axioms. A meticulous manual review of these annotations and axioms was conducted. This critical step was essential to ensure the relevance and integrity of the data included in our datasets. Axioms identified as irrelevant or extraneous were systematically excluded. To maintain the quality of the dataset, any data points that raised doubts regarding their relevance to the axioms were also removed. Although this may result in datasets that are less challenging, this approach significantly simplified the subsequent validation process. In particular, for the GeoFault ontology, we sought additional validation by consulting with the original ontologists who developed GeoFault. Their expert insights were crucial for confirming the accuracy and relevance of the data extracted for the GeoFault benchmark dataset. Consider the second use case, which is focused on semantic search, specifically, extracting axioms that are semantically pertinent to reference texts. We leveraged the principle of justifications, a foundational notion that is proposed in the field of knowledge representation and reasoning. Justifications, also known as axiom pinpointing, are concerned with identifying a minimal subset of an ontology that preserves a specified logical consequence that follows the given ontology. Formally, a *justification* [1, 24] for $\mathcal{O} \models \beta$ is defined as a subset $\mathcal{M} \subseteq \mathcal{O}$, such that $\mathcal{M} \models \beta$ and for any $\mathcal{M}' \subsetneq \mathcal{M}$, $\mathcal{M}' \not\models \beta$. There may exist multiple justifications for a single logical consequence β. For this use case, the dataset was constructed by generating reference texts as $\mathbb{V}(\beta)$, with β on the form of $A \sqsubseteq B$, where A and B are any concept names in the ontology such that $\mathcal{O} \models A \sqsubseteq B$. The verbalization of the logical statement β, along with the corresponding axioms, are defined as $\{\alpha \mid \alpha \in J, J \in \text{Just}(\mathcal{O}, \beta)\}$. For example, if $\mathcal{O} \models \beta$ and $\beta \coloneqq \text{Abscess} \sqsubseteq \text{Clinical\_Finding}$, $\alpha_1 \coloneqq \text{Abscess} \sqsubseteq \text{Disease}$, $\alpha_2 \coloneqq \text{Disease} \sqsubseteq \text{Clinical\_Finding}$, where $\alpha_1, \alpha_2 \in \mathcal{O}$. There exists

[1] The ceiling function denotes the smallest integer greater than or equal to $k/2$. We select the top $\lceil k/2 \rceil$ candidates predicted by the BERT models for their linguistic strengths and then enrich them with $\lceil k/2 \rceil$ candidates from an Ontology Atomic Graph, ensuring a balanced integration of natural language processing and ontology-based enhancements.

[2] As k is typically much smaller than the number of nodes reachable from the initially selected axiom nodes, we assume Algorithm 1 terminates to simplify the presentation.

[3] https://termbrowser.nhs.uk/

[4] http://www.w3.org/2004/02/skos/core#definition

only one justification, $\text{Just}(\mathcal{O}, \beta) = \{\{\alpha_1, \alpha_2\}\}$, we can use justification to create the benchmark. That is, the most relevant axioms to the reference text $\mathbb{V}(\beta)$, "Every abscess is a Disease", are $\{\alpha_1, \alpha_2\}$.

This approach ensures that the axioms included are those from all justifications for β in $\mathcal{O}$, utilizing tools detailed in [6]. This approach ensures that the axioms included are those from all justifications for β in $\mathcal{O}$, utilizing tools detailed in [6]. This results in an absolutely minimal set of axioms that are relevant to the reference text $\mathbb{V}(\beta)$, thereby optimizing the relevance of the dataset. The datasets GeoFault, Disease, and Anatomy contain 102, 2147, and 976 reference texts along with their corresponding axioms, respectively.

5.2 Evaluation Metrics

To evaluate the performance of our framework, we adapt traditional metrics, specifically mHits@k and Mean Reciprocal Rank (MRR), to accommodate scenarios commonly encountered in our dataset where multiple valid axioms correspond to a single input reference text.

Mean Hits@k. The mean Hits@k metric, denoted as mHits@k, measures the frequency with which correct axioms appear within the top k predicted axioms. Unlike the traditional Hits@k, which typically accounts for a single correct response, mHits@k is designed to handle multiple valid axioms for each input reference text. It thus calculates the proportion of correct answers found among the top-k positions, normalized by the smaller of k or the total number of correct answers. The metric is defined as follows:

$$\text{mHits@}k = \frac{\sum_{\alpha \in \mathcal{C}} |\mathcal{L}_k \cap \{\alpha\}|}{\min\{k, |\mathcal{C}|\}},$$

where $\mathcal{C}$ represents the set of all correct axioms for a given input reference text ref, and $\mathcal{L}_k$ is the set of top k predicted axioms. It ensures that the metric accurately reflects the density of correct axioms in the top k predictions, adjusted for the number of correct axioms.

Mean Reciprocal Rank (MRR). The Mean Reciprocal Rank (MRR) is adapted to handle multiple correct axioms per input reference text. This adaptation is crucial for ensuring that the metric fairly evaluates the earliest correct axiom prediction within the context of multiple valid responses. The adapted MRR is calculated as:

$$\text{MRR} = \frac{1}{|\mathcal{C}|} \sum_{\alpha \in \mathcal{C}} \frac{1}{\text{rank}(\alpha, \mathcal{L}_k)},$$

where $\text{rank}(\alpha, \mathcal{L})$ denotes the ranking position of the axiom α in the prediction axiom list $\mathcal{L}$.

5.3 Experimental Settings and Baselines

Implementation Details. We utilize the DeepOnto framework[5] to manage ontology operations and verbalize concepts. We employ several BERT-based models for token embeddings, including the basic BERT model, SBERT [26], and SapBERT [19]. SBERT is selected for its ability to efficiently generate semantically meaningful sentence embeddings, which are crucial for accurate and rapid similarity assessments across extensive datasets. This model provides an optimal balance between computational efficiency and performance. Additionally, SapBERT is incorporated specifically for its expertise in the biomedical domain, where it excels due to specialized training on large-scale biomedical corpora, enhancing semantic accuracy in medical text processing. To improve the efficiency of our similarity computations between axiom sentence embeddings and reference texts, we incorporate the FAISS library [14]. This tool enables efficient indexing of axioms and rapid similarity calculations. By focusing on the top 10% closest axiom vectors, we significantly reduce computational overhead without sacrificing accuracy, as confirmed by sensitivity analysis conducted on the GeoFaults dataset. This analysis demonstrated negligible statistical variation in results with or without this targeted selection approach[6]. For atom decomposition, we employ the Fact++ reasoner [34]. Our computational framework integrates the LLaMA 2 [7] 7b and 13b models among LLMs due to their compatibility with our computational resources. We also utilize the GPT-3.5 and GPT-4 models [8]. However, due to the significantly higher computational cost of GPT-4 (which is 60 times greater than GPT-3.5), we limit its application to a subset of the datasets, but still obtain insights into its performance. To ensure statistical rigor in our evaluation, we calculate the necessary sample size to achieve a 95% confidence level with a 5% margin of error, assuming a population characteristic proportion of 50%. This approach is based on the Central Limit Theorem [11], which asserts that the sample proportion $\hat{p}$ is normally distributed around the true population proportion p with a variance of $\frac{p(1-p)}{n}$. The calculated random sample size of the Disease and Anatomy datasets are 326 and 214, respectively.[9]

Baseline Comparisons. As baselines, we utilize TF-IDF tokenizers [32] and Word2Vec [21] to generate embeddings for tokens, which are then aggregated via mean pooling to represent reference texts and axiom sentences. The similarity between these embeddings is quantified via cosine similarity, in alignment with our framework.

5.4 Analysis of Experimental Results

To generate prompts for LLMs, we selected the top 20 ranked axiom candidates, in order to balance computational efficiency with a broad capture of potential alignments. The overall experiment results, as delineated in Table 2, provide a comprehensive view of the performance landscape. A critical element of our experiment involved dissecting the impact of each system component, which offered insights into the additive value of each feature. In scenarios where only BERT models were employed, denoted as the '{BERT model}', e.g., 'SBERT' system setting, no semantic enhancement or involvement of LLMs occurred—the performance rested solely on the BERT model's cosine similarity calculations. Comparatively, a setting like '{BERT model}+{LLM model}', e.g.,'SBERT+LLaMA 7B' excluded semantic enhancement, focusing on the direct contributions of LLaMA 7B. The setting 'OAG+{BERT model}+{LLM model}', represents the full capabilities of our framework. It employs the Ontology Atomic Graph (OAG) to refine and enhance the axiom ranking process, illustrating the synergy between structured ontology data and advanced language model analytics.

Analysis of the results indicates a clear hierarchical pattern in performance, with simple models like TF-IDF and Word2Vec providing a baseline and more complex models incorporating LLMs and semantic enhancement strategies outperforming the baseline. While TF-IDF delivers moderate outcomes, particularly in the GeoFault dataset, its performance lags in more complex semantic tasks, which

[5] https://github.com/KRR-Oxford/DeepOnto

[6] Results detailed to three decimal places showed no statistical variation, ensuring reliability in computational reduction strategies.

[7] https://huggingface.co/blog/llama2

[8] https://openai.com/

[9] The implementation and datasets are available at: https://github.com/JieyingChenChen/ontoTextAlignment.

	GeoFault				Disease				Anatomy			
System Settings	mHits@1	mHits@5	mHits@10	MRR	mHits@1	mHits@5	mHits@10	MRR	mHits@1	mHits@5	mHits@10	MRR
TF-IDF	0.500	0.350	0.300	0.138	0.197	0.194	0.231	0.185	0.069	0.247	0.301	0.142
Word2Vec	0.588	0.417	0.413	0.279	0.049	0.059	0.073	0.047	0.099	0.177	0.216	0.106
BERT	0.078	0.063	0.061	0.050	0.011	0.020	0.028	0.012	0.193	0.154	0.191	0.107
SBERT	0.676	0.465	0.485	0.325	0.216	0.303	0.343	0.196	0.591	0.296	0.317	0.255
SapBERT	0.657	0.474	0.478	0.313	0.256	0.366	**0.423**	0.234	0.297	0.277	0.306	0.180
BERT+LLaMA 7b	0.029	0.029	0.038	0.016	0.001	0.002	0.002	0.002	0.052	0.063	0.08	0.035
SBERT+LLaMA 7b	0.088	0.050	0.062	0.045	0.012	0.031	0.037	0.016	0.050	0.048	0.066	0.033
SapBERT+LLaMA 7b	0.078	0.033	0.045	0.034	0.007	0.031	0.040	0.014	0.035	0.034	0.044	0.021
BERT+LLaMA 13b	0.020	0.023	0.032	0.017	0.007	0.023	0.028	0.010	0.116	0.142	0.160	0.082
SBERT+LLaMA 13b	0.069	0.070	0.066	0.047	0.097	0.259	0.299	0.128	0.213	0.238	0.263	0.143
SapBERT+LLaMA 13b	0.039	0.055	0.070	0.031	0.099	0.285	0.350	0.141	0.117	0.207	0.242	0.109
BERT+GPT 3.5	0.137	0.061	0.059	0.051	0.037	0.040	0.040	0.021	0.394	0.191	0.190	0.154
SBERT+GPT 3.5	0.627	0.440	0.436	0.287	0.313	**0.380**	0.405	0.214	0.645	0.291	0.289	0.245
SapBERT+GPT 3.5	0.569	0.409	0.399	0.263	0.294	0.377	0.408	0.209	0.600	0.278	0.278	0.226
BERT+GPT 4	0.245	0.093	0.087	0.085	0.037*	0.040*	0.040*	0.021*	0.547*	0.237*	0.243*	0.209*
SBERT+GPT 4	0.765	0.472	0.469	0.344	0.313*	**0.380***	0.405*	0.214*	0.776*	0.323*	**0.335***	0.305*
SapBERT+GPT 4	0.419	0.458	0.468	0.344	0.294*	0.377*	0.408*	0.209*	0.593*	0.288*	0.289*	0.228*
OAG+BERT+LLaMA 13b	0.108	0.065	0.068	0.050	0.004	0.021	0.029	0.008	0.059	0.141	0.166	0.075
OAG+SBERT+LLaMA 13b	0.471	0.434	0.449	0.245	0.071	0.182	0.214	0.120	0.126	0.242	0.269	0.134
OAG+SapBERT+LLaMA 13b	0.520	0.456	0.456	0.255	0.091	0.224	0.267	0.140	0.076	0.222	0.261	0.11
OAG+BERT+GPT 3.5	0.186	0.084	0.079	0.067	0.029	0.030	0.031	0.018	0.444	0.203	0.202	0.177
OAG+SBERT+GPT 3.5	0.676	0.43	0.418	0.293	0.264	0.223	0.227	0.198	0.674	0.297	0.296	0.254
OAG+SapBERT+GPT 3.5	0.686	0.446	0.433	0.298	0.292	0.275	0.279	0.222	0.602	0.285	0.285	0.239
OAG+BERT+GPT 4	0.196	0.088	0.087	0.083	0.040*	0.029*	0.031*	0.023*	0.547*	0.239*	0.239*	0.221*
OAG+SBERT+GPT 4	0.676	0.430	0.418	0.293	0.356*	0.269*	0.285*	0.259*	**0.804***	**0.324***	**0.335***	**0.320***
OAG+SapBERT+GPT 4	**0.804**	**0.490**	**0.485**	**0.359**	**0.396***	0.300*	0.316*	**0.275***	0.654*	0.316*	0.324*	0.281*

Table 2: Comparative performance of various system configurations across benchmarks. Values marked with an asterisk (*) are estimated from a statistical model using randomly selected samples, achieving a 95% confidence level with a 5% margin of error, assuming a population characteristic proportion of 50% based on the Central Limit Theorem [11]. The highest values in each column are highlighted in bold.

require deeper linguistic and semantic understanding. This is indicative of the limitations inherent in non-contextual models when tasked with discerning finer semantic distinctions.

The incremental gains from semantic enhancement based on OAG are significant. Together with LLMs, such as 'GPT 4', this is evident by the fact that'SBERT+GPT 4' or 'SapBERT+GPT 4' achieve the highest scores in most categories. By adding the semantic enhancement (OAG) component to the setting, most system configurations achieve higher scores compared to their original settings without using semantic enhancement. It is not surprising, as semantic enhancement adds rich semantic information to refine the candidate lists, give LLMs additional context to consider.

The integration of LLMs like GPT significantly enhances the performance of our framework, particularly for configurations such as SBERT and SapBERT. Upgrading to version 4 from 3.5, for instance, yields marked improvements, demonstrating the efficacy of these advanced models in semantic tasks. Conversely, the addition of components like LLaMA 7B clearly demonstrates how each element within our system can specifically affect performance. Not only do these components fail to consistently enhance performance across all metrics, but they can also significantly worsen outcomes, as evidenced by the comparisons between 'SBERT' and 'SBERT+LLaMA 7B', or 'SapBERT' and 'SapBERT+LLaMA 7B'. This variability highlights the complex and unique requirements of different ontology-specific tasks, emphasizing the need for careful component selection based on the targeted dataset. The GPT models improve the performance significantly for SBERT and SapBERT, especially in versions 4 as opposed to 3.5, showing the effectiveness of integrating generative LLMs into the task. However, the incremental gains from adding components such as LLaMA 7B point to the nuanced role that each system element plays. While these additions may not always result in stark improvements across all metrics, their influence is more pronounced in certain datasets, underscoring the idiosyncratic nature of ontology-specific tasks, sometimes, it will make the result significantly worst, for example comparing the setting 'SBERT' with 'SBERT+LLaMA 7b', or 'SapBERT' with 'SapBERT+LLaMA 7b'.

Limitation. Our framework are constrained by the BERT model's inherent limitation of processing only up to 512 tokens. For axiom sentences that surpass the token limit, a potential solution involves decomposing a long axiom into several logically equivalent, shorter axioms. A significant bottleneck in our methodology is in the axiom retrieval process. Exploring the development of specialized BERT models tailored to distinct datasets represents a promising avenue for enhancing our ability to retrieve relevant axioms more accurately. Additionally, our datasets are created through the annotation of axioms and supplemented by manual validation. Further refinement and development, particularly with the involvement of domain experts, are crucial for enhancing the quality of our datasets. This improvement is essential for more accurately assessing the tools.

6 Conclusion and Future Work

In this paper, we introduced the problem of ontology text alignment and focused on extracting terminological axioms from textual data. We developed a comprehensive framework that integrates knowledge representation techniques with BERT and LLMs to enhance solution implementation. We also constructed three distinct datasets for evaluation. Future work will focus on enriching these datasets with domain expert validation and advancing our framework to embed deeper semantic meanings into the axiom retrieval process.

Acknowledgements

This work was supported by the EPSRC ConCur project (EP/V050869/1), Samsung Research UK, and the NWO Zorro project (KICH1.ST02.21.003). We extend our sincere thanks to Dr. Yuanwei Qu for providing feedback on the annotation of the GeoFault dataset, Dr. Dirk Walther for his contributions to the implementation of atomic decomposition as detailed in [20], and Dr. Yuan He for assistance with the DeepOnto tool.

References

[1] F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic EL+. In J. Hertzberg, M. Beetz, and R. Englert, editors, *Proceedings of the 30th Annual German Conference on AI, KI 2007*, volume 4667 of *Lecture Notes in Computer Science*, pages 52–67. Springer, 2007. doi: 10.1007/978-3-540-74565-5_7.

[2] J. Chen, M. Ludwig, Y. Ma, and D. Walther. Towards extracting ontology excerpts. In *Proc. of KSEM'15: the 8th International Conference on Knowledge Science, Engineering and Management*, pages 78–89, 2015.

[3] J. Chen, M. Ludwig, Y. Ma, and D. Walther. Zooming in on ontologies: Minimal modules and best excerpts. In *Proc. of ISWC'17*, pages 173–189, 2017.

[4] J. Chen, M. Ludwig, and D. Walther. Computing minimal subsumption modules of ontologies. In *Proc. of GCAI'18*, pages 41–53, 2018.

[5] J. Chen, M. Ludwig, Y. Ma, and D. Walther. Computing minimal projection modules for *ELH*ˆr -terminologies. In F. Calimeri, N. Leone, and M. Manna, editors, *Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings*, volume 11468 of *Lecture Notes in Computer Science*, pages 355–370. Springer, 2019. doi: 10.1007/978-3-030-19570-0_23.

[6] J. Chen, Y. Ma, R. Peñaloza, and H. Yang. Union and intersection of all justifications. In P. Groth, M. Vidal, F. M. Suchanek, P. A. Szekely, P. Kapanipathi, C. Pesquita, H. Skaf-Molli, and M. Tamper, editors, *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 - June 2, 2022, Proceedings*, volume 13261 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2022. doi: 10.1007/978-3-031-06981-9_4.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

[8] H. Dong, J. Chen, Y. He, and I. Horrocks. Ontology enrichment from texts: A biomedical dataset for concept discovery and placement. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 5316–5320, 2023.

[9] H. Dong, J. Chen, Y. He, Y. Liu, and I. Horrocks. Reveal the unknown: Out-of-knowledge-base mention discovery with entity linking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 452–462, 2023.

[10] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007. doi: 10.1007/978-3-540-49612-0.

[11] W. Feller. *An Introduction to Probability Theory and Its Applications, Volume 1*. An Introduction to Probability Theory and Its Applications. Wiley, 1968. ISBN 9780471257080.

[12] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.*, 31(1):273–318, 2008.

[13] Y. He, J. Chen, E. Jiménez-Ruiz, H. Dong, and I. Horrocks. Language model analysis for ontology subsumption inference. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3439–3453. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.213.

[14] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021. doi: 10.1109/TBDATA.2019.2921572.

[15] B. Konev, C. Lutz, D. Walther, and F. Wolter. Semantic modularity and module extraction in description logics. In *Proc. of ECAI'08: the 18th European Conference on Artificial Intelligence*, pages 55–59. IOS Press, 2008.

[16] B. Konev, C. Lutz, D. Walther, and F. Wolter. Model-theoretic inseparability and modularity of description logic ontologies. *Artif. Intell.*, 203: 66–103, 2013.

[17] R. Kontchakov, F. Wolter, and M. Zakharyaschev. Logic-based ontology comparison and module extraction, with an application to dl-lite. *Artif. Intell.*, 174(15):1093–1141, 2010. ISSN 0004-3702.

[18] P. Koopmann and J. Chen. Deductive module extraction for expressive description logics. In C. Bessiere, editor, *Proceedings of IJCAI'20*, pages 1636–1643. ijcai.org, 2020.

[19] F. Liu, A. Vlachos, and T. Cohn. Self-alignment pretraining for biomedical entity representations. *Nature Machine Intelligence*, 3(4):316–325, 2021.

[20] F. Martín-Recuerda and D. Walther. Fast modularisation and atomic decomposition of ontologies using axiom dependency hypergraphs. In *Proc. of ISWC'14: the 13th International Semantic Web Conference*, volume 8797 of *Lecture Notes in Computer Science*, pages 49–64, 2014.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[22] T. Mossakowski, M. Codescu, F. Neuhaus, and O. Kutz. The distributed ontology, modeling and specification language – dol. In A. Koslow and A. Buchsbaum, editors, *The Road to Universal Logic*, volume 2, pages 489–520. Birkhäuser, 2015.

[23] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–20, 2024. doi: 10.1109/TKDE.2024.3352100.

[24] R. Peñaloza. Axiom pinpointing. In G. Cota, M. Daquino, and G. L. Pozzato, editors, *Applications and Practices in Ontology Design, Extraction, and Reasoning*, volume 49 of *Studies on the Semantic Web*, pages 162–177. IOS Press, 2020. doi: 10.3233/SSW200042.

[25] Y. Qu, M. Perrin, A. Torabi, M. Abel, and M. Giese. Geofault: A well-founded fault ontology for interoperability in geological modeling. *Comput. Geosci.*, 182:105478, 2024. doi: 10.1016/J.CAGEO.2023.105478.

[26] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992, 2019.

[27] S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3 (4):333–389, 2009.

[28] U. Sattler, T. Schneider, and M. Zakharyaschev. Which kind of module should I extract? In *Proc. of DL'09*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[29] Ö. Sevgili, A. Shelmanov, M. Arkhipov, A. Panchenko, and C. Biemann. Neural entity linking: A survey of models based on deep learning. *Semantic Web*, 13(3):527–570, 2022.

[30] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2014.

[31] W. Shen, Y. Li, Y. Liu, J. Han, J. Wang, and X. Yuan. Entity linking meets deep learning: Techniques and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):2556–2578, 2021.

[32] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[33] H. Stuckenschmidt, C. Parent, and S. Spaccapietra. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *Lecture Notes in Computer Science*. Springer Verlag, 01 2009.

[34] D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. In U. Furbach and N. Shankar, editors, *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4130 of *Lecture Notes in Computer Science*, pages 292–297. Springer, 2006. doi: 10.1007/11814771_26.

[35] C. D. Vescovo, B. Parsia, U. Sattler, and T. Schneider. The modular structure of an ontology: Atomic decomposition. In *Proc. of IJCAI'11: the 22nd International Joint Conference on Artificial Intelligence*, pages 2232–2237. IJCAI/AAAI, 2011.

[36] Q. Wang, Z. Gao, and R. Xu. Exploring the in-context learning ability of large language model for biomedical concept linking. *arXiv preprint arXiv:2307.01137*, 2023.