

# An Embedding-based Approach to Inconsistency-tolerant Reasoning with Inconsistent Ontologies

Keyu Wang  
School of Computer Science and  
Engineering, Southeast University  
Nanjing, China

Site Li  
School of Mathematics, Southeast  
University  
Nanjing, China

Jiaye Li  
School of Mathematics, Southeast  
University  
Nanjing, China

Guilin Qi  
School of Computer Science and  
Engineering, Southeast University  
Nanjing, China

Qiu Ji  
School of Modern Posts & Institute of  
Modern Posts, Nanjing University of  
Posts and Telecommunications  
Nanjing, China

## ABSTRACT

Inconsistency handling is an important issue in knowledge management. Especially in ontology engineering, logical inconsistencies may occur during ontology construction. A natural way to reason with an inconsistent ontology is to utilize the maximal consistent subsets of the ontology. However, previous studies on selecting maximal consistent subsets have rarely considered the semantics of the axioms, which may result in irrational inference. In this paper, we present a novel approach to reasoning with inconsistent ontologies in description logics based on the embeddings of axioms. We first propose a sentence-embedding-based method and a knowledge-graph-embedding-based method for translating axioms into semantic vectors to calculate semantic similarities among axioms. We then define an embedding-based approach for selecting the maximal consistent subsets of the inconsistent ontology and use it to define an inconsistency-tolerant inference relation. We show the rationality of our inference relation by considering some logical properties. Finally, we conduct experiments on several ontologies to evaluate the effectiveness and efficiency of our proposed method. The experimental results show that our embedding-based method can outperform existing inconsistency-tolerant reasoning methods based on maximal consistent subsets.

## 1 INTRODUCTION

Ontologies are widely used in knowledge management and are critical for the success of the Semantic Web because they provide formal representation of knowledge shared within the Semantic Web applications. The development of the Semantic Web is further accelerated with the proposal of Knowledge Graph, which provides users with more intelligent services, such as more accurate recommendation and search [10]. Ontologies also have a critical impact on the performance of Knowledge Graph reasoning [28]. However, conflicting knowledge in ontologies is unavoidable. For example, ontology fusion [15], ontology evolution [14] and ontology migration [39] may result in inconsistent ontologies. Therefore, inconsistency handling is an essential issue in ontology engineering.

A natural way to reason with inconsistent ontologies is to select maximal consistent subsets of an inconsistent ontology [36]. An elementary method is called skeptical inference [27], i.e., an axiom can be inferred if it can be inferred from every maximal consistent

subset of the inconsistent ontology. A well-known refinement of skeptical inference is to utilize the cardinality-maximal consistent subsets of the ontology for inference [3]. However, in [26], the shortcomings of these two methods are pointed out. They fail to give fine consideration to the difference in the reliability of axioms, which results in weak reasoning power and poor answering quality. [26] gives a general class of monotonic selection relations for comparing maximal consistent subsets. Each monotonic selection relation corresponds to a rational inference relation. However, the approaches given in [26] are limited to propositional logic and it is not trivial to apply them to description logics. Axioms in an ontology contain semantic information, which can be used to define a rational inconsistency-tolerant reasoning method. We use an example to illustrate this.

**Example.** We consider an example of an inconsistent ontology that contains four axioms selected from the widely used ontology OpenCyc<sup>1</sup>:

$\varphi_1 : \text{ArtifactualFeatureType}(\text{Monument})$   
 $\varphi_2 : \text{ExistingStuffType}(\text{Monument})$   
 $\varphi_3 : \text{DisjointWith}(\text{ExistingObjectType}, \text{ExistingStuffType})$   
 $\varphi_4 : \text{SubClassOf}(\text{ArtifactualFeatureType}, \text{ExistingObjectType})$

The existing methods based on propositional logic would assume that these axioms have the same status as shown in the example (continue) in Section 2, regardless of the semantics of axioms. However, these axioms contain different semantic information, which can be exploited to compute the reliability of axioms and select the maximal consistent subset of the ontology.

Pre-trained transformer-based language models for embedding such as BERT [11] and Knowledge Graph Embedding models such as TransE [9] have been successfully applied in many natural language processing tasks and achieved good performance. Recently, they have been applied to ontology matching [18] and instance matching [19] to encode the semantics of instances or concepts in an ontology. They were also used to learn ontologies from knowledge graphs [37]. This motivates us to use embedding-based models to encode the semantic information of an axiom in a description logics ontology and apply them to inconsistency-tolerant reasoning.

In this paper, we propose a novel approach to reasoning with inconsistent OWL ontologies based on the embeddings of axioms. We

<sup>1</sup><https://sourceforge.net/projects/opencyc/>

first introduce two methods, a sentence-embedding-based method and a knowledge-graph-embedding-based method, for translating axioms into semantic vectors to compute semantic similarities among them. By considering the number of occurrences in the maximal consistent subsets and the degree of semantic relationships with other axioms, each axiom could be associated with a degree of reliability. We then propose an embedding-based approach to scoring the maximal consistent subsets and select the maximal consistent subset with the highest score. The selected maximal consistent subset of the ontology can be used to infer consistent and reasonable results. We show the rationality of our proposed reasoning method by considering some logical properties. Finally, we conduct experiments on several ontologies to evaluate the effectiveness and efficiency of our proposed method. The experimental results show that our embedding-based method can outperform existing inconsistency-tolerant reasoning methods based on maximal consistent subsets. The data, source codes and technical report including proofs are provided in the link: <https://github.com/skyfish23/Embedding-based-infer>.

## 2 RELATED WORK

There are mainly two classes of methods for inconsistency handling in DL-based ontologies. One deals with inconsistencies by repairing them and the other tolerates inconsistencies and changes the semantics of description logics. Our work falls into the latter class. In this part, we mainly discuss existing approaches to reasoning with inconsistent OWL DL ontologies based on consistent subsets. [4, 5] provide surveys on this topic. To find consistent subsets, [23] proposes a linear extension strategy for checking whether an entailment could be inferred or not by defining syntactic relevance functions. This work is extended in [20] by defining semantic relevance functions with Google distances. However, such an approach may not find maximal consistent subsets and it may result in many unknown answers to queries.

To achieve some kind of minimal change in the calculation of the consistent subsets, researchers have proposed various methods to retain information as much as possible. The work in [12] focuses on *SHIQ* ontologies, and assumes that an ontology is composed of a consistent TBox, and the instance assertions in the ABox are associated with weights. It answers a conjunctive query upon any subset of the set including the TBox and a weight-maximally consistent subset from the ABox. Similarly, [7] computes four kinds of maximal subsets by exploiting additional information in the ABox, which are maximal with respect to cardinality, weights, prioritized set inclusion or prioritized cardinality. They focus on DL-Lite, which is a sub-language of OWL DL. Furthermore, [40] presents a sound and complete method for DL-Lite ontologies and an approximation method for more expressive DLs, so that the query answering systems can scale up to billions of data. [8] provides a practical approach by considering three well-known semantics for DL-Lite ontologies and defines explanations for answers. Recently, [6] proposes propositional encoding of maximality and then develops several SAT-based algorithms to calculate answers.

There are also some other methods to perform reasoning over an inconsistent ontology which do not rely on (maximal) consistent subsets of the ontology. For instance, various non-classical semantics could be adopted. The works in [30, 31] adopt four-valued and

three-valued semantics for weakening an interpretation in DL from two truth values to four and three values respectively. To infer more useful information from an inconsistent ontology, [45] defines a novel description logics based on the quasi-classical logic.

Different from all these existing methods, our approach does not rely on weight information to select maximal consistent subsets of an ontology as many real ontologies do not have weight information. Instead, we propose a new approach for selecting maximal consistent subsets of an ontology by considering axiom embedding.

## 3 PRELIMINARIES

An ontology is a formal representation of knowledge that defines a set of concepts and relationships between them, and OWL (Web Ontology Language) is a standard language recommended by W3C<sup>2</sup> used to represent ontologies. Description logics (DLs), as the logic foundation of OWL, provide reasoning support for OWL ontologies and we refer to the DL Handbook [2] for a detailed introduction.

A DL-based ontology describes the characteristics of some properties and the relationships between entities through various axioms. Such axioms can be categorized into TBox (terminological axioms) and ABox (assertional axioms). A TBox establishes a conceptualization of a knowledge domain such as concept inclusion in the form  $C \sqsubseteq D$ , where  $C$  and  $D$  are concepts. An ABox describes particular individuals, such as concept assertions in the form of  $C(a)$ , where  $C$  is a concept and  $a$  is an individual.

An ontology is inconsistent if it has no model. To reason with an inconsistent ontology, we consider its maximal consistent sub-ontologies, which are maximal subsets of the ontology that are consistent.

**Definition 1 (MCS).** *Given an ontology  $\Sigma$ , a maximal consistent sub-ontology (MCS)  $\Sigma'$  of  $\Sigma$  satisfies:*

- $\Sigma' \subseteq \Sigma$
- $\Sigma'$  is consistent
- If  $\Sigma' \subset \Sigma'' \subseteq \Sigma$ , then  $\Sigma''$  is not consistent.

We use  $\text{mcs}(\Sigma)$  to represent the set composed of all the maximal consistent sub-ontologies of a given ontology  $\Sigma$ .

[26] first defines mappings that attach a score to each axiom of an ontology and then aggregate those scores to rank each MCS. The MCS with the highest score is selected and used to define the inconsistency-tolerant reasoning method. Since only axioms are used when computing scores for axioms or MCSs, we use the term ontology to refer to a (finite) set of axioms for simplification in the following definitions. In this paper, when we mention  $\alpha \in \Sigma$  given an ontology  $\Sigma$ , we assume  $\alpha$  is an axiom and  $|\Sigma|$  denotes the number of axioms in the ontology  $\Sigma$ . We adapt the notion of a scoring function given in [26] to description logics.

**Definition 2** (scoring function). *A scoring function  $s$  associates with an ontology  $\Sigma$  and an axiom  $\alpha \in \Sigma$  a non-negative real number  $s(\Sigma, \alpha)$  which is equal to 0 if and only if  $\alpha$  is trivial (i.e., such that  $\alpha$  is a tautology or a contradiction).*

Different from [26], in DL ontologies in this paper, we exclude the cases where the axioms are trivial. [26] designs several scoring functions, including the #mc function mentioned in the following Definition 3.

**Definition 3** (#mc). *Let  $\Sigma$  be an ontology and an axiom  $\alpha \in \Sigma$ .*

<sup>2</sup><https://www.w3.org/TR/owl-overview/>

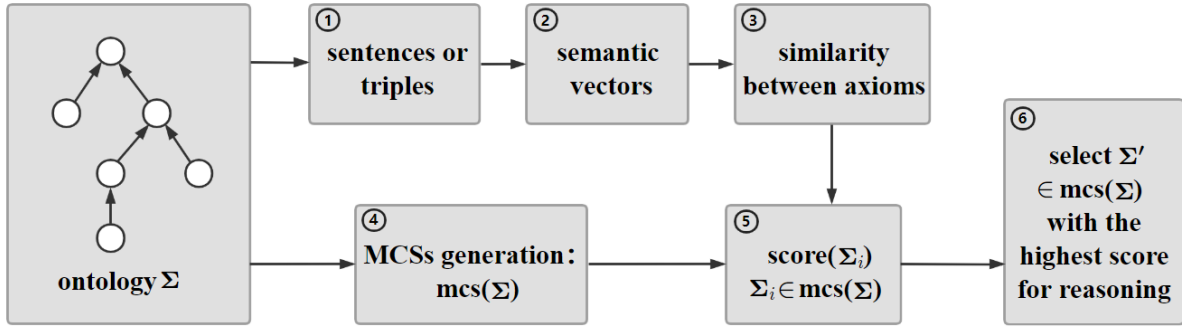


Figure 1: An embedding-based approach to reasoning with an inconsistent ontology.

Define:

$$\#mc(\Sigma, \alpha) = |\{\Sigma_i \in mcs(\Sigma) \mid \alpha \in \Sigma_i\}|$$

**Definition 4** ( $score_{\Sigma, sum}^{\#mc}$ ). Let  $\Sigma$  be an ontology and  $\Sigma_i \in mcs(\Sigma)$ .

Define:

$$score_{\Sigma, sum}^{\#mc}(\Sigma_i) = \sum_{\alpha \in \Sigma_i} \#mc(\Sigma, \alpha)$$

**Example(continue)** Ontology  $\Sigma = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ .  $\Sigma$  has 4 MCSs namely:  $\{\varphi_2, \varphi_3, \varphi_4\}$ ,  $\{\varphi_1, \varphi_3, \varphi_4\}$ ,  $\{\varphi_1, \varphi_2, \varphi_4\}$ ,  $\{\varphi_1, \varphi_2, \varphi_3\}$ . We have  $\#mc(\Sigma, \varphi_1) = \#mc(\Sigma, \varphi_2) = \#mc(\Sigma, \varphi_3) = \#mc(\Sigma, \varphi_4) = 3$ , so the score of  $\Sigma_1$  (resp.  $\Sigma_2, \Sigma_3, \Sigma_4$ ) w.r.t  $score_{\Sigma, sum}^{\#mc}$  is equal to 9 (resp. 9, 9, 9). These four MCSs all have the same value and are selected for reasoning, i.e., an axiom can be inferred if it can be inferred from each of these four MCSs.

Through the definitions above, the MCS  $\Sigma'$  of the original inconsistent  $\Sigma$  with the highest score is selected, and the selected MCS can be used for reasoning. In the example above, all the four MCSs are selected because they all have the highest score, though this situation is rare in our proposed approach. In this work, we also define a reasoning method by attaching a score to each axiom in an ontology and scoring the MCSs. But different from [26], we propose an embedding-based novel method to score each axiom considering not only the syntax but also the semantics of the axioms.

In this work, a sentence-embedding-based method and a knowledge-graph-embedding-based method are proposed for translating axioms into semantic vectors to calculate semantic similarities among axioms. We introduce some background knowledge about embedding models as follows.

**Sentence Embedding.** A sentence embedding is a mathematical representation of a sentence in a fixed-length vector space that encodes the semantic meaning and contextual information of the sentence[33]. Generally, the more similar the semantics of two sentences are, the closer their corresponding vectors are in the semantic space. In this work, we leverage Sentence-BERT[35] and CoSENT[43] to get the semantic vectors of axioms. Sentence-BERT trains the upper classification function by supervised learning and makes balances between performance and efficiency, while CoSENT takes advantage of a ranking loss function, which makes the training process closer to prediction.

**Knowledge Graph Embedding.** Knowledge Graph Embedding

(KGE) is a task for learning low-dimensional representation, typically called embeddings, of a knowledge graph's entities and relations while preserving their semantics. A knowledge graph can be assumed as the ABox of an ontology while KGE can also be applied to embedding ontologies. For example, [17] uses KGE models such as TransE[9], DisMult[44] and HolE[34] to learn ontologies by jointly embedding instances and concepts. There are various KGE models and we refer to [42] for a comprehensive survey. In this work, we use TransE [9] and RDF2Vec[38]. TransE is based on the assumption that  $\mathbf{v}_s + \mathbf{v}_r$  is close to  $\mathbf{v}_o$ , where  $\mathbf{v}_s, \mathbf{v}_r, \mathbf{v}_o$  are the embeddings of subject entity, relation and object entity respectively, while RDF2Vec utilizes random walks on the RDF graph to create sequences of RDF nodes, which are then used as input for the word2vec[32] algorithm.

## 4 AN EMBEDDING-BASED METHOD TO REASON WITH INCONSISTENT ONTOLOGIES

The overall framework of our method is shown in Figure 1. Our method can be divided into six parts. The first five parts can be assumed as offline preparation, including translating axioms into sentences or triples, using a sentence-embedding-based method or a knowledge-graph-embedding-based method to map the axioms into semantic vectors in a continuous space, calculating similarity between axioms based on the embeddings, generating and scoring the MCSs. Finally, we select the MCS with the highest score for reasoning.

### 4.1 Semantic Representation & Embedding

We introduce two methods, a sentence-embedding-based method and a knowledge-graph-embedding-based method in this part, and either of the two methods can be used to convert axioms into semantic vectors.

**Sentence-embedding-based method.** To represent the semantics of classes, individuals and properties in an ontology, we first use NaturalOWL [1] to translate axioms from OWL statements to natural language sentences. For example, the axiom "ClassAssertion(ObjectMaxCardinality(1 madeFromGrape) product145)" is translated to "product145 is a made from at most one Grape". Table 1 shows most of the rules to translate OWL concepts or axioms into

Ontology Elements	OWL Representation	Phrases or Sentences	Triples
Concepts	ObjectSomeValuesFrom(op A)	op at least one A	-
	ObjectAllValuesFrom(op A)	op only A	-
	ObjectHasValue(op a)	op a	-
	ObjectIntersectionOf(A B)	A and B	-
	ObjectUnionOf(A B)	A or B	-
	ObjectExactCardinality(n op A)	op exactly n A	-
	ObjectMinCardinality(n op A)	op at least n A	-
	ObjectMaxCardinality(n op A)	op at most n A	-
Axioms	SubClassOf(A B)	A is a kind of B	<A, SubClassOf, B>
	DisjointClasses(A B)	A isn't a kind of B	<A, Disjointness, B>
	EquivalentClasses(A B)	A is a kind of B	<A, EquivalentClasses, B>
	ClassAssertion(A a)	a is a A	<a, isInstanceOf, C>
	ObjectPropertyAssertion(op a b)	a op b	<a, op, b>
	DataPropertyAssertion(dp a v)	a dp v	<a, dp, v>

**Table 1: Rules to translate OWL concepts or axioms into sentences or triples, respectively, where  $A$  and  $B$  are concepts,  $a$  and  $b$  are individual names,  $n$  indicates an integer,  $op$  and  $dp$  indicate an object property and a data property separately, and  $v$  is a value.**

natural language sentences implemented by NaturalOWL. Despite the fact that NaturalOWL does not translate all kinds of axioms or entities such as property inclusion axioms and transitive properties, the system can handle OWL statements in most ontologies, including various ontologies in our experiment.

Each axiom is transformed into a sentence in natural language form by NaturalOWL[1], and then the sentence is input into a certain sentence embedding model to obtain the corresponding semantic vector representation. Considering that BERT-based pre-trained language models have achieved good performance in text representation and text matching tasks, we use Sentence-BERT [35] and CoSERT [43] as sentence embedding models to compute the embeddings of the axioms in natural language sentence form. We refer to the official website of PyPi<sup>3</sup> for more details about the implementation and experimental performance of these models.

**Knowledge-graph-embedding-based method.** The method based on sentence embedding can make use of the textual information of ontology, but ignores other semantic information of ontology, such as logical expression and class hierarchies. We propose another method using KG embedding, which is used to learn the relational information in the ontology. We implement the system “TripleOWL” of transforming axioms into triples and its transforming mechanism is similar to NaturalOWL. For example, an axiom “SubClassOf(IceCream Food)” is translated into “<IceCream, SubClassOf, Food>”. The last column of Table 1 shows most of the rules to translate axioms to triples. Nonetheless, different from translating axioms to triples, we fail to deal with complex concepts when translating axioms to sentences, that is, we regard the complex concept in an axiom as a whole, which may cause worse performances for when the ontology has many axioms with complex concepts. We leave how to deal with complex concepts in KGE-based method to future work. The three parts of a triple are called subject entity,

relation and object entity respectively for simplicity. Then the three components of the triples can be embedded into vectors using KG embedding models, and the vectors corresponding to the subject entity, relation and object entity are concatenated together to obtain the semantic vector representation of this axiom. We use TransE [9], RDF2Vec [38] in our experiments, which are implemented by OpenKE platform [16] and pyRDF2Vec[41] respectively.

## 4.2 Semantic Similarity of Embedding

We compute the semantic similarities among axioms. We denote the similarity of the axioms  $\alpha, \beta$  as  $Sim(\alpha, \beta)$  and the embedding of the axiom  $\alpha$  as  $Emb(\alpha)$ . The similarity calculation method is given as follows:

$$Sim(\alpha, \beta) = Similarity(Emb(\alpha), Emb(\beta))$$

*Similarity* represents certain similarity calculation method. The most common ones are based on Cosine Distance and Euclidean Distance:

$$Similarity_{Cos}(\mathbf{v}_1, \mathbf{v}_2) = \frac{1}{2} \left( 1 + \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \times \|\mathbf{v}_2\|} \right)$$

$$Similarity_{Euc}(\mathbf{v}_1, \mathbf{v}_2) = \frac{1}{1 + \sqrt{\|\mathbf{v}_1 - \mathbf{v}_2\|}}$$

where  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are vectors of the same arity.

$Similarity_{Cos}$  and  $Similarity_{Euc}$  satisfy the following three properties. For simplicity, we use  $Sim(\phi, \varphi)$  to denote either  $Similarity_{Cos}(Emb(\phi), Emb(\varphi))$  or  $Similarity_{Euc}(Emb(\phi), Emb(\varphi))$ .

**Range.** The semantic similarity is a real number between 0 and 1:  $0 \leq Sim(\phi, \varphi) \leq 1$  for any  $\phi$  and  $\varphi$ . The higher the similarity between  $\phi$  and  $\varphi$ , the closer the  $Sim(\phi, \varphi)$  is to 1, and the opposite is to 0.

**Grammatical Reflexivity.** Any axiom is always semantically closest to itself:  $Sim(\phi, \phi) = 1$  for any  $\phi$ .

**Symmetry.** The semantic similarity between two axioms is symmetric:  $Sim(\phi, \varphi) = Sim(\varphi, \phi)$  for any  $\varphi$  and  $\phi$ .

<sup>3</sup><https://pypi.org/project/text2vec/>

However, the similarity functions may not satisfy the semantic reflexivity defined below, because two semantically equivalent axioms may be converted into different sentences. This does not affect that our proposed method satisfies the properties in Section 5 so our method still has the logical rationality. And in our work, we use a unique set of transformation rules proposed by NaturalOWL[1], and the experimental part shows the significant performance of our embedding-based method under this set of rules.

**Semantic Reflexivity.** Two semantically identical axioms should be closest to each other: If  $\models \phi \leftrightarrow \varphi$ , then  $Sim(\phi, \varphi) = 1$ .

**Example (continue)** By applying the cosine similarity metric and Sentence-BERT, we obtain  $Sim_{Cos}(\varphi_1, \varphi_2) = 0.41$  and  $Sim_{Cos}(\varphi_1, \varphi_4) = 0.62$ . This may be because  $\varphi_4$  has a same concept *ArtifactualFeatureType* with  $\varphi_1$  while  $\varphi_3$  doesn't, so  $\varphi_4$  has higher semantic association with  $\varphi_1$ .

### 4.3 Semantic Selection Functions

We first define the degree of aggregation of each axiom in the MCS. Then we score each axiom to represent the reliability of the axioms. Finally we aggregate the scores of the axioms to get the score of each MCS.

We use the degree of aggregation to express how closely an axiom relates to other axioms in a MCS. The greater the aggregation degree of an axiom in a MCS, the closer the axiom is to the semantics of other axioms in this MCS, which indicates more semantic information it contains about this MCS.

**Definition 5 (agg).** Given an ontology  $\Sigma$  and  $\Sigma_i \in mcs(\Sigma)$ , we define the aggregation of an axiom  $\alpha \in \Sigma_i$  as follows:

$$agg(\Sigma_i, \alpha) = \frac{1}{|\Sigma_i|} \sum_{\beta \in \Sigma_i} Sim(\alpha, \beta)$$

We then calculate the score for each axiom. If an axiom exists in more MCSs and it has a higher degree of aggregation in the MCSs it appears in which means it is more likely a true axiom in these MCSs, this axiom is considered to be more reliable. Below, we define scoring function for axioms.

**Definition 6 (mc).** Given an ontology  $\Sigma$  and an axiom  $\alpha \in \Sigma$ , we define the score of an axiom  $\alpha \in \Sigma$  as follows:

$$mc(\Sigma, \alpha) = \sum_{\{\Sigma_i \in mcs(\Sigma) | \alpha \in \Sigma_i\}} agg(\Sigma_i, \alpha)$$

Finally we accumulate the scores of axioms in each MCS to obtain the scores of the MCSs.

**Definition 7 (scoring function)** Given an ontology  $\Sigma$  and  $\Sigma_i \in mcs(\Sigma)$ , the scoring function for the maximal consistent sub-ontologies are defined as follows:

$$score(\Sigma_i) = \sum_{\alpha \in \Sigma_i} mc(\Sigma, \alpha)$$

Each MCS is assigned a score by the scoring function in Definition 7. Then we select the MCS with the highest score for reasoning.

**Example (continue).** For the example mentioned above, we summarize the above calculations: MCS  $\Sigma_3 = \{\varphi_1, \varphi_2, \varphi_4\}$  gets the highest score. In our method, we conclude  $\varphi_1, \varphi_2, \varphi_4$  and use these axioms for reasoning.

## 5 LOGICAL PROPERTIES

We consider the logical properties of the inference relation between the axioms in  $\Sigma$  and reasoning results using our method. An inference relation is rational when it satisfies the six properties in the minimal set of expected properties of preferential inference relations (also called *system P*) and one of the rational inference relations (also called *system R*) [27].

<b>Ref</b>	$\alpha \sim \alpha$	<b>Cut</b>	$\frac{\alpha \wedge \beta \sim \gamma, \alpha \sim \beta}{\alpha \sim \gamma}$
<b>LLE</b>	$\frac{\models \alpha \leftrightarrow \beta, \alpha \sim \gamma}{\beta \sim \gamma}$	<b>Or</b>	$\frac{\alpha \sim \gamma, \beta \sim \gamma}{\alpha \vee \beta \sim \gamma}$
<b>RW</b>	$\frac{\models \alpha \rightarrow \beta, \gamma \sim \alpha}{\gamma \sim \beta}$	<b>CM</b>	$\frac{\alpha \sim \beta, \alpha \sim \gamma}{\alpha \wedge \beta \sim \gamma}$
<b>RM</b>	$\frac{\alpha \not\sim \neg\beta, \alpha \sim \gamma}{\alpha \wedge \beta \sim \gamma}$		

For example, **Cut** expresses the fact that one may, in his way towards a plausible conclusion, first add a hypothesis to the facts he knows to be true and prove the plausibility of his conclusion from this enlarged set of facts, and then deduce (plausibly) this added hypothesis from the facts. We refer to the technical report<sup>4</sup> for detailed explanations about these properties.

To investigate whether our inference relation satisfies the logical properties above, we present some significant definitions in [27] and theorems as follows:

**Definition 8 (Aggregation function).**  $\oplus$  is an aggregation function if for every positive integer  $n$ , for every non-negative real number  $x_1, \dots, x_n$ ,  $\oplus(x_1, \dots, x_n)$  is a non-negative real number.

For example, this paper chooses *sum* as the aggregation function.

**Definition 9 (score $_{\Sigma, \oplus}^s$ ).** Let  $s$  be a scoring function defined in Definition 2 and let  $\oplus$  be an aggregation function. Let  $\Sigma$  be an ontology and  $\Sigma_i = \{\alpha_1 \dots \alpha_n\} \subseteq \Sigma$ . We define  $score_{\Sigma, \oplus}^s(\Sigma_i) = \oplus_{\alpha \in \Sigma_i} s(\Sigma, \alpha)$ .

In Definition 7, we denote  $score_{\Sigma, sum}^{mc}(\Sigma_i)$  as  $score(\Sigma_i)$  for simplicity. On the foundation of Definition 9, we can define the monotonic selection relation according to our method. Before that, we propose a more general one.

**Definition 10 (Monotonic selection relation).** Given an ontology  $\Sigma$ , let  $\geq_{\Sigma} \subseteq 2^{\Sigma} \times 2^{\Sigma}$  be a reflexive, transitive and total relation over the powerset of  $\Sigma$ .  $\geq_{\Sigma}$  is said to be a monotonic selection relation if for every consistent set  $\Sigma_i \subseteq \Sigma$ , for every non-trivial axiom  $\alpha \in \Sigma \setminus \Sigma_i$ ,  $\Sigma_i \cup \{\alpha\} >_{\Sigma} \Sigma_i$ .

For instance, the relation  $\geq_{card}$  defined over  $P(\Sigma)$  is a monotonic selection relation, in which  $\Sigma_i \geq_{card} \Sigma_j$  if and only if  $|\Sigma_i| \geq |\Sigma_j|$ . To be specific, a selection relation and an inference relation in [26] are given as follows:

**Definition 11 ( $\geq_{\Sigma, \oplus}^s$ ).** Let  $s$  be a scoring function and  $\oplus$  an aggregation function. Let  $\Sigma$  be an ontology,  $\Sigma_i, \Sigma_j \subseteq \Sigma$ . We state that  $\Sigma_i \geq_{\Sigma, \oplus}^s \Sigma_j$  if and only if  $score_{\Sigma, \oplus}^s(\Sigma_i) \geq score_{\Sigma, \oplus}^s(\Sigma_j)$ .

Based on Definition 11, we can compare the subsets of an ontology by the scores of them. We adopt the following notation for convenience:  $mcs(\Sigma, \alpha) = \{\Sigma_i \subseteq \Sigma \mid \Sigma_i \cup \{\alpha\} \in mcs(\Sigma \cup \{\alpha\})\}$ .

**Definition 12 (mcs $_{\geq_{\Sigma}}$ ).** Given an ontology  $\Sigma$ , an axiom  $\alpha \in \Sigma$ , and a monotonic selection relation  $\geq_{\Sigma}$ , we define  $mcs_{\geq_{\Sigma}}(\Sigma, \alpha) = \{\Sigma_i \in mcs(\Sigma, \alpha) \mid \text{there exists no } \Sigma'_i \in mcs(\Sigma, \alpha) \text{ such that } \Sigma'_i >_{\Sigma} \Sigma_i\}$ .

<sup>4</sup><https://github.com/sky-fish23/Embedding-based-infer/blob/main/technical%20report.pdf>

Ontology Name	#class	#prop.	#indi.	#axiom	#MCS	Expressivity
AUTO.-cocus-eda	158	86	114	907	6	$\mathcal{ALCOIN}$
bioportal-metadata	98	183	61	822	14	$\mathcal{ALUHN}$
UOBM-lite-10-35	52	40	43	162	16	$\mathcal{SHOIN}$
UOBM-lite-10-36	52	40	43	163	40	$\mathcal{SHOIN}$

**Table 2: Inconsistent ontologies used in the evaluation.**

With respect to a monotonic selection relation, we define a selection mechanism consisting in keeping only the best subsets.

**Definition 13** [26] ( $\sim_{\Sigma}^{mcs \geq \Sigma}$ , a binary inference relation based on best subsets w.r.t  $\geq_{\Sigma}$ ). Given an ontology  $\Sigma$ , two axioms  $\alpha$  and  $\beta$ , and a monotonic selection relation  $\geq_{\Sigma}$ , we state that  $\alpha \sim_{\Sigma}^{mcs \geq \Sigma} \beta$  if and only if for every  $\Sigma_i \in mcs_{\geq \Sigma}(\Sigma, \alpha)$ , we have  $\Sigma_i \cup \{\alpha\} \models \beta$ .

We propose the inference relation to show how conclusions are inferred from an ontology.

**Theorem 1.** Let  $\Sigma$  be an ontology and let  $\geq_{\Sigma}$  be a selection relation over  $\Sigma$ . Then  $\geq_{\Sigma}$  is monotonic if and only if there exists a scoring function  $s$  for  $\Sigma$  such that  $s(\Sigma, \alpha) > 0$  for every axiom  $\alpha \in \Sigma$ .

**Proof sketch of Theorem 1.** From the definition of monotonic selection relation and  $\geq_{\Phi, \Sigma}$ , this result can be easily induced. Especially, for necessity, to ensure the arbitrariness of  $\alpha$ , we have to show that for every axiom  $\alpha \in \Sigma$ , there exists a consistent subset  $\Sigma_i \subseteq \Sigma$ , such that  $\alpha \notin \Sigma_i$ .

Based on Theorem 1, we show that the given selection relation is monotonic. Furthermore, Theorem 2 shows the equivalence of the rationality of an inference relation and the monotonicity of its corresponding selection relation.

**Theorem 2.** A relation  $\sim$  is rational if and only if there exists an ontology  $\Sigma$  and a monotonic selection relation  $\geq_{\Sigma}$  such that  $\sim_{\Sigma}^{mcs \geq \Sigma} = \sim$ .

**Proof sketch of Theorem 2.** Theorem 2 is a rewriting of Theorem 5.18 in [27], so the proof is the same as that.

**Theorem 3.** The relation  $\sim$  defined by the selection relation which takes our proposed scoring function is rational.

**Proof sketch of Theorem 3.** According to Theorem 2, we know that the rationality of the inference relation is equivalent to the monotonicity of corresponding selection relation. So we have to verify the positivity of our scoring functions to ensure every axiom  $\alpha \in \Sigma$ ,  $s(\Sigma, \alpha) > 0$  (We exclude the trivial axioms).

In conclusion, the selection relation based on our proposed scoring function is monotonic relation according to Theorem 1. And Theorem 2 shows the rationality of the corresponding inference relation. Due to the rationality of our proposed method, the reasoning satisfies all the seven logical properties mentioned above. Detailed proofs of these theorems and explanation of the properties can be found in the technical report <sup>5</sup>.

## 6 EXPERIMENT AND EVALUATION

In this section, we first introduce the experimental dataset and settings. Then we conduct some experiments to show the performance of our proposed method.

### 6.1 Data

In order to fairly evaluate the quality of our proposed method in answering queries, the test ontology dataset should be diverse. We select four inconsistent ontologies, which vary in sources, size, expressiveness, number of axioms and MCSs. Table 2 provides details of the dataset, where #class, #prop., #indi., #axiom and #MCS represent the number of classes, properties, individuals, axioms and MCSs for the ontology respectively. Half of the ontologies are existing inconsistent ontologies crafted by others, and others are created by us based on existing ontologies.

AUTO.-cocus-eda is constructed in [24] by merging two source ontologies *cocus* and *edas* with their mapping generated by the mapping system AUTOMSV2 which has participated in the famous contest of ontology alignment evaluation initiative<sup>6</sup>. Its inconsistency is caused by multiple sources, lying on two similar concepts ‘cocus#person’ and ‘edas#person’. Bioportal-metadata<sup>7</sup> is a real-life ontology from the world’s most comprehensive repository of biomedical ontologies [24]. Some inconsistencies come from multiple sources of certain individual. For instance, these four axioms are inconsistent:

```

 $\varphi_1$  : SubClassOf(KnowledgeRepresentationParadigm,
      documentation max 1 rdfs : Literal)
 $\varphi_2$  : Domain(documentation, KnowledgeRepresentationParadigm)
 $\varphi_3$  : documentation(Jena-ARQ, "http://jena.sourceforge.net/ARQ/")
 $\varphi_4$  : documentation(Jena-ARQ, "http://jena.sourceforge.net/")

```

The two UOBM ontologies are constructed by us based on UOBM-lite-10 with an increasing number of axioms. UOBM [13] indicates University Benchmark and is enriched from the famous Lehigh University Benchmark (LUMB) [29]. UOBM-lite-10 means that it is enriched from LUMB by adding OWL Lite constructors and contains individuals from 10 universities. To construct inconsistent ontologies, we borrowed the idea from [12] and develop the tool *Injector*<sup>8</sup> to insert different numbers of conflicts. The mechanism to generate conflicts in this work is the same as [12] and details about the implementation of *Injector* are shown in section 5 of [12]. A conflict is a set of instance assertions violating a functional role restriction or a disjointness constraint. UOBM-lite-10-35 and UOBM-lite-10-36 are obtained by inserting 35 and 36 conflicts into ontology UOBM-lite-10 respectively.

### 6.2 Experimental Settings

The experiment includes four parts: generating the MCSs, transforming axioms with NaturalOWL or TripleOWL, scoring the MCSs and testing the queries. We perform the MCS generation system, NaturalOWL system and TripleOWL system in Java 1.8.0 environment, on a laptop with 1.80 GHz Intel core CPU, 16 GB RAM and a 64-bit Windows 10 operating system. The MCS generation system can be applied to any DLs. We implement the scoring function for selecting MCSs using Python 3.9.7 and PyTorch 1.11.0 with the same computer requirements as MCS generation system. For the query test experiment, we used the widely used ontology editor Protégé<sup>9</sup> with version 5.5.0 and max memory set to 444MB. All

<sup>6</sup><http://oaei.ontologymatching.org/2012/conference/>

<sup>7</sup><https://bioportal.bioontology.org/>

<sup>8</sup><https://anonymous.4open.science/r/Embedding-based-infer/data/Injector/>

<sup>9</sup><https://protege.stanford.edu/>

<sup>5</sup><https://github.com/sky-fish23/Embedding-based-infer/blob/main/technical%20report.pdf>

Method	Similarity	Queries	IA	CA	RA	CIA	IA Rate	ICR Rate
AUT.-cocus-edas								
Skeptical inference		124	6	118	0	0	4.84%	100%
CMCS		124	8	116	0	0	6.45%	100%
#mc		124	8	116	0	0	6.45%	100%
Sentence-BERT	Cosine	124	122	0	2	0	98.39%	100%
	Euclid	124	122	0	2	0	98.39%	100%
ConSERT	Cosine	124	119	3	1	1	95.97%	99.21%
	Euclid	124	122	0	3	0	98.39%	100%
TransE	Cosine	124	8	116	0	0	6.45%	100%
	Euclid	124	8	116	0	0	6.45%	100%
RDF2Vec	Cosine	124	122	0	2	0	98.39%	100%
	Euclid	124	8	116	0	0	6.45%	100%
Biportal metadata								
Skeptical inference		41	22	19	0	0	53.66%	100%
CMCS		41	22	19	0	0	53.66%	100%
#mc		41	22	19	0	0	53.66%	100%
Sentence-BERT	Cosine	41	36	0	3	2	87.80%	95.12%
	Euclid	41	38	0	3	0	92.68%	100%
ConSERT	Cosine	41	38	0	3	0	92.68%	100%
	Euclid	41	38	0	3	0	92.68%	100%
TransE	Cosine	41	38	0	3	0	92.68%	100%
	Euclid	41	38	0	3	0	92.68%	100%
RDF2Vec	Cosine	41	36	2	2	1	87.80%	97.56%
	Euclid	41	22	19	0	0	53.66%	100%
UOBM-lite-10-35								
Skeptical inference		90	74	16	0	0	82.22%	100%
CMCS		90	74	16	0	0	82.22%	100%
#mc		90	74	16	0	0	82.22%	100%
Sentence-BERT	Cosine	90	80	0	4	6	88.89%	93.33%
	Euclid	90	89	0	0	1	98.89%	98.89%
ConSERT	Cosine	90	84	0	4	2	93.33%	97.78%
	Euclid	90	89	0	0	1	98.89%	98.89%
TransE	Cosine	90	83	1	4	2	92.22%	97.78%
	Euclid	90	89	0	0	1	98.89%	98.89%
RDF2Vec	Cosine	90	78	3	4	5	87.67%	94.44%
	Euclid	90	74	16	0	1	82.22%	100.00%
UOBM-lite-10-36								
Skep.		92	61	31	0	0	63.30%	100%
CMCS		92	71	12	5	4	77.17%	95.65%
#mc		92	71	12	5	4	77.17%	95.65%
Sentence-BERT	Cosine	92	82	0	5	5	89.13%	94.57%
	Euclid	92	86	0	5	1	93.48%	98.91%
ConSERT	Cosine	92	82	0	5	5	89.13%	94.57%
	Euclid	92	86	0	5	1	93.48%	98.91%
TransE	Cosine	92	82	0	5	5	89.13%	94.57%
	Euclid	92	86	0	5	1	93.48%	98.91%
RDF2Vec	Cosine	92	86	0	5	1	93.48%	98.91%
	Euclid	92	71	12	5	4	77.17%	95.65%

Table 3: IA = Intended Answers, CA = Cautious Answers, RA = Reckless Answers, CIA = Counter-Intuitive Answers, IA Rate = Intended Answers(%), ICR Rate = IA+CA+RA(%)

the data and codes including MCS generator, NaturalOWL system, TripleOWL system, *Injector* and scoring function are available at <https://anonymous.4open.science/r/Embedding-based-infer/>.

We selected three existing methods based on MCSs as baselines as follows.

- Skeptical inference [27].
- CMCS: It is based on selecting those cardinality-maximal consistent subsets, known as a refinement on skeptical inference [3].
- #mc: It is proposed by [26] considering the reliability of the information carried by the axioms as shown in Definitions 3, 4.

Our proposed method is according to Definition 5, 6, 7 and we have implemented different variances corresponding to difference embedding models.

[26] proposed a series of reasonable reasoning approaches based on MCS which satisfy a set of logical properties, but there is a lack of experiment to evaluate the performance of the reasoning methods. In this section, each algorithm is evaluated with respect to its effectiveness and efficiency. When reasoning with an inconsistent ontology, the effectiveness of one algorithm indicates the quality of the answers to the test queries. The efficiency of the algorithm is the time to generate answers to the test queries. We use the same evaluation method as [21, 22]. We compare answers generated by the method to be evaluated with hand-crafted Gold Standard that contains the humanly-judged correct answer which is supposed by human, including ontology experts and postgraduate students. For a query, there might exist the following differences between an answer generated by the method to be evaluated and its intuitive answer.

- Intended Answer: the method's answer is the same as the intuitive answer.
- Counter-intuitive Answer: the method's answer is opposite to the intuitive answer. Namely, the intuitive answer is 'accepted' whereas the method's answer is 'rejected', or vice versa.
- Cautious Answer: the intuitive answer is 'accepted' or 'rejected', but the method's answer is 'undetermined'.
- Reckless Answer: the method's answer is 'accepted' or 'rejected' whereas the intuitive answer is 'undetermined'. Under this situation the method to be evaluated returns just one of the possible answers without seeking other possibly opposite answers, which may lead to 'undetermined'.

We randomly select classes, individuals and properties to form the queries such as 'Denmark(individual) Type country(concept) ?', 'Administrator(concept) SubclassOf Person(concept) ?'. Especially, we give greater weight to classes, individuals and properties that are more likely to cause inconsistency when randomly forming queries. We compared these answers against a hand-crafted Gold Standard that contained the humanly-judged correct answer for all of these queries.

### 6.3 Evaluation Results

Table 3 presents the evaluation results. Baselines cannot give an effective MCS selection method because the semantic information of the ontology is ignored, which leads to poor reasoning quality. Compared with baselines, our method has a significant improvement in the intended answer (IA) rate. For AUT.cocus-edas, the IA rate of the baselines is lower than 7%, but our proposed method based on

sentence embedding can be above 96%. The method based on knowledge graph embedding doesn't perform well here, because there is rich and important textual information in this ontology, which is not utilized in the knowledge-graph-embedding-based method, and simply concatenating parts of the triple can't represent the meaning of an axiom well especially for axioms with complex concepts. For bioportal metadata and the two UOBM ontologies, our method has improved the IA rate by more than 10% and 20% respectively with comparison to the baselines. The baselines achieve high ICR rate at the cost of low intended answer (IA) rate. However, our proposed method can achieve both high ICR rate and IA rate. For AUT.cocus-edas and bioportal metadata, most results for the ICR rate of our method are 100%. For the two UOBM ontologies, our method can achieve more than 93% and the majority of the results are above 98%. These reflect the fact that the proposed method could provide promising results when reasoning with inconsistent ontologies. It owes to the semantic information of the axioms considered in our embedding-based method.

We also evaluate the efficiency of our method with respect to selecting MCSs. For the four ontologies in our experiment, the consumed time of the proposed method is within 10 minutes. Although the time consumed by the baselines is within several seconds and our proposed method spends more time than the baselines to score and select MCSs, it is efficient enough in practice as the procedure of scoring and selection only needs to be performed once for each ontology and this process can be done offline. Suppose the selection is done, we also evaluate the time to execute each query. Our proposed approach is very efficient and a query can be answered within about half a second.

## 7 CONCLUSION AND DISCUSSION

In this paper, we introduced a new approach to reasoning with inconsistent ontologies based on maximal consistent subsets. As far as we know, this is the first work that applies embedding techniques to inconsistency-tolerant reasoning. In our work, we first proposed two methods for turning axioms into semantic vectors and computed the semantic similarity between axioms. We then proposed an approach for selecting the maximal consistent subsets and defined an inconsistency-tolerant inference relation. We showed the logical properties of proposed inconsistency-tolerant inference relation. We have proved that the inference relation we proposed satisfies the logic properties, which showed the rationality of the inference relation. We conducted extensive experiments on four ontologies and the experimental results show that our embedding-based method can outperform existing methods based on maximal consistent subsets.

As for future work, we will give more careful consideration to dealing with complex axioms. We also plan to extend our method to inconsistency-tolerant inference with weighted ontologies and consider applying ChatGPT for translating axiom into sentences. Finally, we can explore applying embedding techniques to ontology repair by defining some relevant relations like [25].

## 8 ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments. This work is partially supported by National Nature Science Foundation of China under No. U21A20488.



## REFERENCES

- [1] Ion Androutsopoulos, Gerasimos Lampouras, and Dimitrios Galanis. 2013. Generating natural language descriptions for OWL ontologies: the NaturalOWL system. *Journal of Artificial Intelligence Research* 48 (2013), 671–715.
- [2] Franz Baader, Diego Calvanese, Deborah McGuinness, Peter Patel-Schneider, Daniele Nardi, et al. 2003. *The description logic handbook: Theory, implementation and applications*. Cambridge university press.
- [3] Salem Benferhat, Claudette Cayrol, Didier Dubois, Jérôme Lang, and Henri Prade. 1993. Inconsistency management and prioritized syntax-based entailment. In *IJCAI*, Vol. 93. 640–645.
- [4] Meghyn Bienvenu. 2020. A short survey on inconsistency handling in ontology-mediated query answering. *KI-Künstliche Intelligenz* 34, 4 (2020), 443–451.
- [5] Meghyn Bienvenu and Camille Bourgaux. 2016. Inconsistency-tolerant querying of description logic knowledge bases. In *Reasoning Web International Summer School*. Springer, 156–202.
- [6] Meghyn Bienvenu and Camille Bourgaux. 2022. Querying Inconsistent Prioritized Data with ORBITS: Algorithms, Implementation, and Experiments. *arXiv preprint arXiv:2202.07980* (2022).
- [7] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. 2014. Querying inconsistent description logic knowledge bases under preferred repair semantics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.
- [8] Meghyn Bienvenu, Camille Bourgaux, and François Goasdoué. 2019. Computing and explaining query answers over inconsistent DL-Lite knowledge bases. *Journal of Artificial Intelligence Research* 64 (2019), 563–644.
- [9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).
- [10] Xiaojun Chen, Shengbin Jia, and Yang Xiang. 2020. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications* 141 (2020), 112948.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [12] Jianfeng Du, Guilin Qi, and Yi-Dong Shen. 2013. Weight-based consistent query answering over inconsistent SHIQ knowledge bases. *Knowledge and Information Systems* 34, 2 (2013), 335–371.
- [13] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. 2005. LUBM: A benchmark for OWL knowledge base systems. *Journal of Web Semantics* 3, 2-3 (2005), 158–182.
- [14] Peter Haase, Frank van Harmelen, Zhisheng Huang, Heiner Stuckenschmidt, and York Sure. 2005. A framework for handling inconsistency in changing ontologies. In *International semantic web conference*. Springer, 353–367.
- [15] Adil Hameed, Alun Preece, and Derek Sleeman. 2004. Ontology reconciliation. In *Handbook on ontologies*. Springer, 231–250.
- [16] Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An Open Toolkit for Knowledge Embedding. In *Proceedings of EMNLP*.
- [17] Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1709–1719.
- [18] Yuan He, Jiaoyan Chen, Denvar Antonyrajah, and Ian Horrocks. 2022. BERTMap: a BERT-based ontology alignment system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 5684–5691.
- [19] Sven Hertling, Jan Portisch, and Heiko Paulheim. 2020. Supervised ontology and instance matching with MELT. *arXiv preprint arXiv:2009.11102* (2020).
- [20] Zhisheng Huang and Frank van Harmelen. 2008. Using semantic distances for reasoning with inconsistent ontologies. In *International Semantic Web Conference*. Springer, 178–194.
- [21] Zhisheng Huang, F. V. Harmelen, and Annette ten Teije. 2005. Reasoning with Inconsistent Ontologies. In *International Joint Conference on Artificial Intelligence*.
- [22] Zhisheng Huang and Frank van Harmelen. 2008. Using semantic distances for reasoning with inconsistent ontologies. In *The Semantic Web-ISWC 2008: 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings 7*. Springer, 178–194.
- [23] Zhisheng Huang, Frank Van Harmelen, and Annette Ten Teije. 2005. Reasoning with inconsistent ontologies. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. 454–459.
- [24] Qiu Ji, Zhiqiang Gao, Zhisheng Huang, and Man Zhu. 2014. Measuring effectiveness of ontology debugging systems. *Knowledge-Based Systems* 71 (2014), 169–186.
- [25] Qiu Ji, Guilin Qi, and Peter Haase. 2009. A relevance-directed algorithm for finding justifications of DL entailments. In *Asian Semantic Web Conference*. Springer, 306–320.
- [26] Sébastien Konieczny, Pierre Marquis, and Srdjan Vesic. 2019. Rational Inference Relations from Maximal Consistent Subsets Selection. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 1749–1755. <https://doi.org/10.24963/ijcai.2019/242>
- [27] Sarit Kraus, Daniel Lehmann, and Menachem Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence* 44, 1-2 (1990), 167–207.
- [28] Jin Liu, Xin Zhang, Yunhui Li, Jin Wang, and Hye-Jin Kim. 2019. Deep learning-based reasoning with multi-ontology for IoT applications. *IEEE Access* 7 (2019), 124688–124701.
- [29] Li Ma, Yang Yang, Zhaoming Qiu, Guotong Xie, Yue Pan, and Shengping Liu. 2006. Towards a complete OWL ontology benchmark. In *European Semantic Web Conference*. Springer, 125–139.
- [30] Yue Ma, Pascal Hitzler, and Zuoquan Lin. 2007. Algorithms for paraconsistent reasoning with OWL. In *European Semantic Web Conference*. Springer, 399–413.
- [31] Frederick Maier, Yue Ma, and Pascal Hitzler. 2013. Paraconsistent OWL and related logics. *Semantic Web* 4, 4 (2013), 395–427.
- [32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. [arXiv:arXiv:1301.3781](https://arxiv.org/abs/1301.3781)
- [33] Mridul K Mishra and Jaydeep Viradiya. 2019. Survey of sentence embedding methods. *International Journal of Applied Science and Computations* 6, 3 (2019), 592–592.
- [34] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.
- [35] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).
- [36] Nicholas Rescher and Ruth Manor. 1970. On inference from inconsistent premisses. *Theory and decision* 1, 2 (1970), 179–217.
- [37] Petar Ristoski, Stefano Faralli, Simone Paolo Ponzetto, and Heiko Paulheim. 2017. Large-scale taxonomy induction using entity and word embeddings. In *Proceedings of the International Conference on Web Intelligence*. 81–87.
- [38] Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*. Springer, 498–514.
- [39] Stefan Schlobach, Ronald Cornet, et al. 2003. Non-standard reasoning services for the debugging of description logic terminologies. In *Ijcai*, Vol. 3. 355–362.
- [40] Eleni Tsalapati, Giorgos Stoilos, Giorgos B Stamou, and George Koletsos. 2016. Efficient Query Answering over Expressive Inconsistent Description Logics.. In *IJCAI*. 1279–1285.
- [41] Gilles Vandewiele, Bram Steenwinkel, Terencio Agozzino, and Femke Ongenae. 2022. pyRDF2Vec: A Python Implementation and Extension of RDF2Vec. (2022). <https://doi.org/10.48550/ARXIV.2205.02283>
- [42] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743. <https://doi.org/10.1109/TKDE.2017.2754499>
- [43] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741* (2021).
- [44] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).
- [45] Xiaowang Zhang, Guohui Xiao, Zuoquan Lin, and Jan Van den Bussche. 2014. Inconsistency-tolerant reasoning with OWL DL. *International Journal of Approximate Reasoning* 55, 2 (2014), 557–584.