

# Online Reasoning for Semantic Error Detection in Text

**Fernando Gutierrez, Dejing Dou,  
Nisansa de Silva & Stephen Fickas**

**Journal on Data Semantics**  
Concepts and Ideas for Building  
Knowledgeable Systems

ISSN 1861-2032

J Data Semant  
DOI 10.1007/s13740-017-0079-6



```
<owl:Class rdf:about="&myCamera"
  <rdfs:subClassOf rdf:resource="
</owl:Class>

<owl:Class rdf:about="&myCamera;DSLR">
  <rdfs:subClassOf rdf:resource="&myCamera;Digita
</owl:Class>

<owl:Class rdf:about="&myCamera;PointAndShoot">
  <rdfs:subClassOf rdf:resource="&myCamera;Digita
</owl:Class>

<owl:Class rdf:about="http://xmlns.com/foaf/0.1/Person">
  <owl:interClassOfPropertyOf rdf:resource="http://xmlns.com/foaf/0.1/Person"
  <owl:Class
    rdf:about="http://xmlns.com/foaf/0.1/Person"
  <owl:Class>
  <owl:hasKey rdf:parseType="Collection"
  <owl:DatatypeProperty
    rdf:about="&myCamera
```

**Journal on  
Data Semantics**

Editor-in-Chief  
Esteban Zimányi

Co-Editors  
Sudha Ram  
Heiner Stuckenschmidt

Springer

Springer

**Your article is protected by copyright and all rights are held exclusively by Springer-Verlag GmbH Germany. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at [link.springer.com](http://link.springer.com)".**

# Online Reasoning for Semantic Error Detection in Text

Fernando Gutierrez<sup>1</sup> · Dejing Dou<sup>1</sup>  · Nisansa de Silva<sup>1</sup> · Stephen Fickas<sup>1</sup>Received: 13 May 2016 / Revised: 19 May 2017 / Accepted: 22 May 2017  
© Springer-Verlag GmbH Germany 2017

**Abstract** Identifying incorrect content (i.e., semantic error) in text is a difficult task because of the ambiguous nature of written natural language and the many factors that can make a statement semantically erroneous. Current methods identify semantic errors in a sentence by determining whether it contradicts the domain to which the sentence belongs. However, because these methods are constructed on expected logic contradictions, they cannot handle new or unexpected semantic errors. In this paper, we propose a new method for detecting semantic errors that is based on logic reasoning. Our proposed method converts text into logic clauses, which are later analyzed against a domain ontology by an automatic reasoner to determine its consistency. This approach can provide a complete analysis of the text, since it can analyze a single sentence or sets of multiple sentences. When there are multiple sentences to analyze, in order to avoid the high complexity of reasoning over a large set of logic clauses, we propose rules that reduce the set of sentences to analyze, based on the logic relationships between sentences. In our evaluation, we have found that our proposed method can identify a significant percentage of semantic errors and, in the case of multiple sentences, it does so without significant computational cost. We have also found that both the quality of the information extraction output and modeling elements

of the ontology (i.e., property domain and range) affect the capability of detecting errors.

**Keywords** Information extraction · Ontology · Semantic error detection

## 1 Introduction

Information extraction (IE) is the task of automatically transforming unstructured data (e.g., text) into structured information (e.g., a knowledge base). Because IE has traditionally used scientific documents as its data source, it assumes that the content of the text is correct. However, as IE moves to the analysis of non-curated domains, such as the Internet, this guarantee of quality and correctness does not hold. This lack of quality has become evident in the form of *fake news* articles that are shared through social media. Therefore, it seems reasonable to incorporate into IE methods and mechanisms to determine the semantic correctness of the text being analyzed.

However, semantic error in text has only been addressed indirectly. Research in automatic text grading has approached semantic error as a deviation from a gold standard. Grading systems based on methods such as latent semantic analysis [23] try to measure how similar a student's writing is to a *perfect* version of a summary or an essay. The greater the difference (i.e., the smaller the similarity) between the student's writings and the gold standard, the more *incorrect* the student's writings. However, a semantically correct text can have low similarity if its content is broader than that of the gold standard, or if it is written in an unexpected form. On the other hand, semantic error can be seen as existence of logic contradiction. Contradiction detection [31] is the subfield of textual entailment that focuses on identifying logic contra-

---

✉ Dejing Dou  
dou@cs.uoregon.edu

Fernando Gutierrez  
fernando@cs.uoregon.edu

Nisansa de Silva  
nisansa@cs.uoregon.edu

Stephen Fickas  
fickas@cs.uoregon.edu

<sup>1</sup> Department of Computer and Information Science, 1202  
University of Oregon, Eugene, OR 97403, USA

diction between bodies of text. By capturing the semantics of the sentences through lexical and syntactical elements, contradiction detection tries to determine whether a pair of sentences cannot be true at the same time (i.e., whether they contradict each other). However, because this approach is limited to determining contradiction within the text itself, it cannot determine with certainty, from a pair of contradicting sentences, which one is false or incorrect.

Following a different approach, we have proposed an ontology-based semantic error detection method using predefined errors for extraction rules [10] and machine learning-based patterns [12]. An ontology provides a formal knowledge representation of a domain, through concepts and relationships. By including a domain ontology into our approach, we have a set of correct facts from the domain (e.g., relationships among concepts of the domain). So, if a sentence contradicts the domain ontology, then the said sentence can be considered to be incorrect with respect to the domain knowledge. Our approach incorporates a heuristic algorithm to generate domain-inconsistent facts that are encoded into an ontology-based information extraction system. Although this approach can identify semantic errors in sentences based on the domain, it is limited by the set of expected semantic errors defined in our heuristic algorithm. As such, the heuristic algorithm has a set of manually defined rules that generate axioms by violating constraints in the domain ontology. However, this is also its main limitation. Our previous approach could only recognize incorrect sentences if they were part of the expected semantic error set. New sentences could not be judged correctly.

In this paper, we propose *online reasoning for semantic error detection*, a new method for identifying semantically incorrect content based on logic reasoning (i.e., inference) and domain knowledge. Our proposed reasoning-based approach consists of two steps. In the first step, sentences are transformed into logic clauses through a combination of IE and *vocabulary mapping*. This step intends to take written natural language (i.e., sentences) to a formalized representation that is compatible with the domain ontology (i.e., ontological axioms). In the second step, the transformed sentence or sentences are included into the ontology to determine their consistency with the domain. This process, which is performed by a reasoner, is known as *consistency checking*. If the domain ontology becomes inconsistent after the inclusion of the extracted text, then the sentence or sentences are semantically incorrect with respect to the domain.

Our proposed method extends semantic error detection from considering only single-sentence analysis to considering multiple-sentence analysis. In single-sentence analysis, we intend to determine the semantic correctness of text by considering one sentence at a time [11]. Under this approach, the semantic content of each sentence is considered, independent from the rest of the text. In the case of multiple-sentence

analysis, we intend to identify sentences that, only when considered as a set, are inconsistent. Because the multiple-sentence analysis leads to a higher computational complexity, we have proposed *reduction rules* as a mechanism to keep the complexity as low as possible. Because our proposed method intends to extract all possible relationships from the text and then performs an analysis against the whole ontology, it is possible to offer a broader analysis than our previous methods.

The remainder of this paper is organized as follows. We introduce some related work in Sect. 2. In Sect. 3, we present how our method transforms text into logic clauses. Then we present, in Sect. 4, the single-sentence analysis of our proposed method, while in Sect. 5, we present the extension to multiple-sentence analysis. We report our experimental results and discuss some observations from our case study in Sect. 6. We conclude the paper by summarizing our contributions and future goals in Sect. 7.

## 2 Background and Related Work

Our proposed semantic error detection approach builds on research done on information extraction, consistency checking, and ontology debugging.

### 2.1 Information Extraction

Because words can have more than one meaning, and because ideas can be stated in multiple ways, text analysis is a very difficult task. This analysis can be simplified, for example, through a *controlled natural language*, which avoids ambiguity by using a subset of a full natural language. Each controlled language, such as Attempto Controlled English (ACE), intends to provide an expressive representation language that is easy for non-experts to use. What facilitates use by non-experts is that a controlled language conforms to standard English grammar and syntax. Yet, its thorough formality allows ACE to describe the logical content of an OWL ontology [20] and to perform automatic logic reasoning [8], among other tasks [22].

However, we currently have large sets of documents in which vast amounts of information are expressed in *non-controlled* languages. Information extraction is the process of automatically identifying relevant semantic elements (entities and relationships) from text and transforming them into structured information (e.g., instances in a knowledge base). For example, from the sentence “Albert Einstein is a scientist of the 20th century,” an IE system can identify that the entities, *Albert Einstein* and *the 20th century*, are connected by the relationship, *is\_a\_scientist\_of*.

An approach to mitigating the complexity of IE is to incorporate domain knowledge (i.e., domain ontology) into the extraction process. Ontology-based information extraction

(OBIE), a subfield of IE, uses an ontology to guide the extraction process by focusing on the extraction of specific concepts and relationships of the domain [38]. Because the domain ontology can provide a vocabulary of the domain, the extraction process is narrowed to the concepts and relationships of the domain.

The current trend of applying IE to a large corpus (e.g., the Internet) has led to considering the amount of human intervention required for a system's deployment. We can distinguish three main strategies based on the level of preparation required: *supervised*, *semi-supervised*, and *unsupervised*. A *supervised* IE system relies on labeled training sets or hand-crafted extraction patterns to produce high-quality extraction from text, which leads to a significant amount of preparation [10]. However, the close processes, labeling data and handcrafting extraction patterns, lead supervised IE systems to produce very accurate extractions. Most OBIE systems follow a supervised extraction strategy.

In the case of *semi-supervised* IE systems, information extraction models and patterns are learned based on known instances of hand-built knowledge bases. In contrast to the supervised systems, which have an explicit link between text and instances, semi-supervised systems have to discover this connection. Initially, semi-supervised IE methods follow the assumption that if a sentence mentions the entities of a known relationship, then the sentence is referring to this relationship [26]. Currently, semi-supervised IE methods have relaxed this assumption by taking into account (a) that a pair of entities can participate in multiple relationships and (b) that a sentence that contains a pair of known entities might not be referring to them [37].

Finally, *unsupervised* IE systems can perform extractions without requiring labeled data or specific pattern construction. They use general patterns based on lexical and syntactical features from the text to identify relationships between entities. Initially, these patterns could only identify *hyponymy* relationships [15]. One of the unsupervised extraction patterns proposed by Hearst [15] is “ $NP_0, NP_i^{i=1..n}$  (and/or) other  $NP$ ,” which allows the extraction of the relation  $[hyponymy(NP, NP_i)]^{i=0..n}$ . New patterns, such as  $NP_0$  Verb  $NP_1$ , have allowed systems such as *ReVerb* [6] to perform the extraction of a wide variety of relationship instances, which can cover a significant quantity of binary relationships [6].

It must be mentioned that semantic parsers can also be considered as IE systems. For example, the semantic parser *Boxer* [2,3] produces a deep analysis of a text by generating discourse representation structures (DRS). Through these structures, *Boxer* can provide a mental representation of the text by considering different levels of abstraction. Although IE outputs, such as the tuples generated by *OLLIE*, can be more easily transformed into relationships and concepts, *Boxer* can also produce clauses that can be used for tasks such as ontology population [29].

## 2.2 Consistency Checking

An ontology is an explicit specification of a conceptualization [9]. Usually used to represent domain knowledge, an ontology can provide a formal model and vocabulary by categorizing the entities of the domain and their properties. In this work, we will focus on ontologies that are based on Description Logic, such as those described by the Web Ontology Language (OWL) [1] proposed by the World Wide Web Consortium (W3C).

Description logic (DL) is a fragment of first-order logic with sound and complete reasoners such as *Hermit* [27]. In DL, a knowledge base  $\mathcal{K}$  consists of a tuple  $(\mathcal{R}, \mathcal{T}, \mathcal{A})$ . The *TBox*  $\mathcal{T}$  is a set of *general concept inclusions* (GCI) of the form  $C \sqsubseteq D$ , for concepts  $C$  and  $D$ . The *ABox* has concepts and role assertions of the form  $C(a)$  and  $R(a, b)$ . Finally, the *RBox*  $\mathcal{R}$  consists of complex role constructions such as role inclusion  $(R_1 \sqsubseteq R_2)$ . However, not all DL knowledge bases define an *RBox*, such as  $\mathcal{ALC}$ , because they do not have role construction.

The semantics of a DL knowledge base  $\mathcal{K}$  are defined by a function  $\cdot^I$  that maps the concepts, roles, and assertions of the knowledge base  $\mathcal{K}$  to the domain  $\Delta^I$ . If  $\mathcal{I}$  satisfies all axioms of  $\mathcal{K}$ , then  $\mathcal{I}$  is a model of  $\mathcal{K}$ , which makes  $\mathcal{K}$  consistent. Determining the satisfiability of a knowledge base is a basic service of a DL reasoner. Its importance comes from the fact that other types of inferences, such as entailment, can be reduced to satisfiability [18]. Satisfiability can be proven by a decision procedure such as a *semantic tableau* (i.e., a tableau algorithm). This method creates a sequence  $1 \dots n$  of *ABoxes*, where the application of derivation rules on *ABox*  $(\mathcal{A}_{i-1})$  results in a new *ABox*  $(\mathcal{A}_i)$  [27]. Following are commonly used tableau derivation rules for DL:

- Given  $C \sqsubseteq D$  and an individual  $s$ , derive  $(\neg C \sqcup D)(s)$ .
- Given  $(C \sqcup D)(s)$ , derive  $C(s)$  or  $D(s)$ .
- Given  $(C \sqcap D)(s)$ , derive  $C(s)$  and  $D(s)$ .
- Given  $(\exists R.C)(s)$ , derive  $R(s, t)$  and  $C(t)$  for a new individual  $t$ .
- Given  $(\forall R.C)(s)$  and  $R(s, t)$ , derive  $C(t)$ .

The tableau algorithm terminates if there are no more derivation rules that can be applied to *ABox*  $(\mathcal{A}_n)$ , or if we reach a contradiction. In the case of a contradiction, the algorithm backtracks to the last OR  $((C \sqcup D)(s))$  derivation and *chooses* a different *path*. If all choices lead to contradiction,  $\mathcal{K}$  is unsatisfiable.

### 2.2.1 Ontology Debugging

Since we consider incorrectness as an inconsistency with respect to the domain ontology, it seems reasonable to consider research regarding ontology inconsistency. However,

DL reasoners cannot determine the origin of an inconsistency. The process of fixing an inconsistent ontology is known as *Ontology Debugging* [7]. The task of Ontology Debugging has been addressed using properties of the underlying description logic (DL) language [34]. These methods try to identify the minimal inconsistent sub-ontology, which is the set of axioms that make the ontology inconsistent. Horridge et al. [17] have identified two main stages in the process of determining the origin of inconsistency in an ontology. In the first stage, they determine the set of all inconsistent sub-ontologies (i.e., inconsistency justification), while in the second stage they construct the minimal inconsistent sub-ontology from the previous set (i.e., ontology repair).

### 2.3 Semantic Error Detection

Recently, we have developed an approach to identifying semantically incorrect information in text by incorporating into the process domain knowledge, i.e., a domain ontology [10–13]. The inclusion of a domain ontology, which provides a formal knowledge representation of a domain through concepts and relationships, yields a set of domain facts (i.e., true statements) in a formal model that allows the use of automatic logic reasoning services, such as consistency checking. Our ontology-based approach led to two methods for semantic error detection in text.

The first method uses a set of predefined semantic errors. This set of semantic errors is obtained by a heuristic algorithm that generates domain-inconsistent statements (i.e., axioms) from the ontology. The generated domain-inconsistent axioms are then encoded into an IE system in the form of pattern-based extraction rules [10, 13] and machine learning models [12, 13].

The second method improves semantic error detection by incorporating a *logic reasoner* for the analysis of the text (i.e., consistency checking) [11]. For the reasoner to be able to analyze the consistency of the text with respect to domain ontology, the text needs to be transformed into logic clauses. This transformation is performed by an IE that uses lexical and syntactical elements from the text to perform the extraction (i.e., unsupervised IE). After sentences have been transformed into logic clauses, they are incorporated into the ontology to determine their consistency with respect to the domain. Our reasoner-based method can identify semantic errors more accurately than our previous method, because it analyzes the text against the whole ontology.

Both semantic error detection methods we have proposed consider each sentence as an independent unit of information. Each sentence in a text can be seen as a set of information. However, sentences usually share information (e.g., anaphora). Even more, this connection between sentences is what provides a document with coherence. In this paper, we present an extension to semantic error detection based on

reasoning that can determine the semantic correctness of sets of sentences.

## 3 Transforming Text into Logic Clauses

Our proposed method for semantic error detection determines the logic consistency of the text against the domain ontology. This *online reasoning* analysis is performed in two steps. In the first step of our proposed approach, sentences need to be transformed from their written form into logic clauses. We achieve this transformation through an IE process and a *mapping* mechanism.

### 3.1 Information Extraction

As previously mentioned in Sect. 2.1, there are three main strategies to IE depending on the level of human intervention (i.e., depending on the level of data preparation). However, because our approach intends to determine the correctness of each sentence presented in the text, not all three strategies are suited for our approach. Supervised IE cannot provide a complete extraction from the text, since the process is guided by known labeled data and predefined patterns. Similarly, semi-supervised IE systems are guided to extract relationships based on sets of known individuals. Plus, in order to provide quality extraction, semi-supervised IE requires a significant set of training individuals.

For the present work, we have chosen the unsupervised strategy followed by the open information extraction system *OLLIE* [24]. Open information extraction systems intend to extract binary relationships, while using neither training data nor handcrafted patterns. The main goal behind this approach is to offer an IE system that can scale to the Web. To do this, open information extraction follows a set of general patterns to extract every possible relationship from a text [6, 24]. In the case of *OLLIE*, the patterns are built by generalizing extractions with high confidence (i.e., high-quality extraction). The set of high-quality extractions is obtained from the IE system *ReVerb* [6], which uses verb-based patterns to identify relations in text. These extractions (e.g., tuples) have two constraints: They contain solely proper nouns as entities participating in the extracted relation and they have a high confidence value. Then, similar to semi-supervised IE systems, *OLLIE* gathers a set of sentences that contain the entities and relations from the extracted tuples. To avoid collecting sentences that might introduce errors, *OLLIE* only gathers sentences with a structure that is centered on the elements of the extracted tuple, i.e., elements of the relation must be in a linear path of at most size four in the dependency parse [24]. From the selected sentences, *OLLIE* learns a set of general extraction patterns. If the structure of a sentence meets a set of requirements (e.g., if the relation is between

the two entities in the sentence), a pure syntactic pattern can be learned from the sentence (e.g., the most general pattern). If the structure of the sentence does not meet the requirements, lexical aspects of the sentence are considered in order to produce a general pattern. These generalized patterns are used to extract new tuples from text. For example, from the sentence, “Scavengers feed from dead organisms,” *OLLIE* will produce the tuple, *feed(Scavengers, dead organism)*.

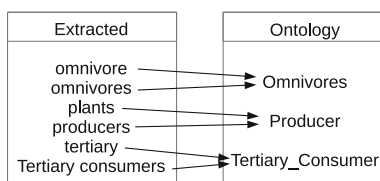
Because we are focused on determining the correctness of the text content, we considered *OLLIE* as a *black box* component of our system. This approach to the extraction component of our method allows us to change to other unsupervised IE systems, such as *ReVerb* [6], in the future without needing to redesign our method.

### 3.2 Mapping Extractions to Ontology

Although the text and the ontology belong to the same domain, it is very possible that the selection of words to represent concepts and relationships might differ. So, to be able to use the domain ontology to evaluate the correctness of the text’s semantics, we need to first solve the lexical gap that might exist between the text and the ontology. In other words, we will need a mapping mechanism that can allow us to pass from the vocabulary of the extracted entities and relationships to the vocabulary of the ontology.

Because we are focused on semantic error detection, we have opted for a simple and direct solution for the translation (i.e., vocabulary mapping) task. This approach is equivalent to the IE method of *gazetteers* (i.e., dictionaries of possible entities) [4, 16, 32, 35, 38]. The mapping mechanism that we proposed is based on two *dictionaries of terms*: one for managing concepts and another for managing relationships. In the case of the dictionary for managing concepts, an extracted entity will lead to the equivalent ontological concept. For example, in Fig. 1, both *dead organisms* and *dead animals* lead to the concept *Dead Organism*.

In the case of managing relationships, because a relationship might have different meaning depending on other elements in the sentence, we consider both subject entity and relation to determine the ontological property. For example, the concept *Carnivores* and the relation *feed* will lead to the property *feed\_from\_herbivore*, while concept *Herbivore* and



**Fig. 1** Example of mapping between extracted terms and ontology concepts

relation *feed* will lead to the property *feed\_from\_producer*. Both dictionaries are generated by considering a subset of extracted relationships (i.e., a sample) from the data set.

## 4 Single-Sentence Analysis

Once we have extracted all the relations from the text (e.g., “Autotrophs produce their food,” to *produce(Autotrophs, food)*), and the relations have been mapped to the vocabulary of the ontology (e.g., *produce(Autotrophs, food)* to *Autotrophs*  $\sqsubseteq$   $\exists$  *produce.Food*), we proceed to analyze the correctness of the sentences by using consistency checking.

As mentioned, we have identified two approaches when analyzing text extractions: single-sentence analysis and multiple-sentence analysis. In single-sentence analysis, we intend to determine the correctness of text by considering one sentence at a time. Under this approach, the semantic content of each sentence is considered, independent from the rest of the text. In the case of multiple-sentence analysis, a group of sentences from the text are analyzed as set of clauses.

In this section, we focus on single-sentence analysis. Each sentence will be included into the domain ontology independently. After the analysis of the sentence has concluded, the sentence’s relationship will be removed from the domain ontology.

However, to be able to determine the semantic correctness of a sentence, we need to consider some requirements for our approach. First, because our online reasoning approach to semantic error detection uses logic reasoning, we need a more strict definition of sentence types. Second, the domain ontology needs to be consistent and complete. In the following sections, we provide details regarding these requirements.

### 4.1 Defining Sentence Types

In our previous work [10], we presented a classification of sentences based on their relationship with the domain. Although the original definition is necessary for determining the sentence’s type, it is not sufficient when using our *online reasoning* approach. In the following sections, we offer a new definition of sentence types for our reasoning-based semantic error detection method.

#### 4.1.1 Correct Sentences

In our previous work [10], we defined a sentence as semantically correct if it was consistent with respect to the domain. A sentence is consistent if the domain does not prove the sentence to be false. However, although consistency is required, it is not sufficient to prove correctness. Even more, if a sentence is completely unrelated to the domain, it is more

likely that the statement will not violate any constraint of the domain. Let us consider the following example:

Ontology  $Planets \sqsubseteq \exists orbits.Stars$   
 $orbits(Earth, Sun)$   
 $Stars(Sun)$

Axiom 1  $Planets(Earth)$   
 Axiom 2  $Myosin \sqsubseteq Amino\_Acid$

In the example, neither axiom contradicts the domain. We can see that Axiom 1, which states that *Earth* is a *Planet* ( $Planets(Earth)$ ), is logically consistent with the domain ontology because from the domain, we know that *Planets* orbit around *Stars* ( $Planets \sqsubseteq \exists orbits.Stars$ ), *Earth* orbits around the *Sun* ( $orbits(Earth, Sun)$ ), and the Sun is a star ( $Stars(Sun)$ ). Axiom 2, which states that *Myosin* is an *amino acid* ( $Myosin \sqsubseteq Amino\_Acid$ ), also is consistent with the domain because it does not contradict the domain. However, the ontology does not have any information regarding the elements that are being referred in Axiom 2 that would allow us to determine Axiom 2's semantic correctness conclusively.

We have revised our definition of semantic correctness. A sentence is semantically correct if it is a logic consequence of the domain, i.e., a semantically correct sentence *s* can be entailed from the domain  $\mathcal{O}$  ( $\mathcal{O} \models s$ ) [11]. For a sentence to be entailed by the domain, it must express either explicit or implicit (i.e., inferred) facts of the domain. From the previous example, we can see that while Axiom 1 is entailed by the domain, the second sentence, although consistent, cannot be entailed from the domain.

#### 4.1.2 Incorrect Sentences

The definition of semantic incorrect sentence presented in our previous work [10] is still valid for our *online reasoning* approach. A sentence *s* is semantically incorrect if it is inconsistent with respect to domain ontology  $\mathcal{O}$  ( $\mathcal{O} \cup s \models \perp$ ) [11].

Ontology  $Producer \sqsubseteq \neg Carnivore$   
 $Producer \sqsubseteq \exists produce.Food$

Axiom 1  $Carnivores \sqsubseteq \exists produce.Food$

In the example, the domain ontology states that *Producers* can create (i.e., produce) *Food* ( $Producer \sqsubseteq \exists produce.Food$ ) and that *Producers* are not *Carnivores* ( $Producer \sqsubseteq \neg Carnivore$ ). Axiom 1 is semantically incorrect because it defines the relationship *Carnivores* produce their *food* ( $Carnivores \sqsubseteq \exists produce.Food$ ), which contradicts the domain ontology.

In light of the new definition of semantically correct sentence presented previously (Sect. 4.1.1), the definition of semantically incorrect sentence becomes more natural. In general, if a sentence is *not* correct, it is considered to be

incorrect. For example, automatic text grading systems based on LSA follow this approach. In LSA-based systems [23], a text is correct if it is very similar to the gold standard (i.e., a perfect text), while an incorrect text has very low similarity. Our definition of semantic incorrect sentence can be restated as the consequence of a false statement, i.e., a sentence *s* is semantically incorrect if its negation is a consequence of the domain ontology ( $\mathcal{O} \models \neg s$ ).

#### 4.1.3 Unknown Sentences

All those sentences that are neither correct nor incorrect shall be considered in this work as *unknown*. This definition is, in essence, the same as in the original definition of a semantically unknown sentence. However, because of the new definition of semantic correctness (Sect. 4.1.1), a sentence is considered as unknown if it is neither true nor false with respect to the domain ontology ( $\mathcal{O} \not\models s$  and  $\mathcal{O} \not\models \neg s$ ). In other words, a sentence is unknown if its truth value cannot be determined.

Ontology  $Producer \text{ is not a Carnivore. } (Tree \sqsubseteq Producer)$   
 $Producers \text{ create their own food. } (Producer \sqsubseteq \exists produce.Food)$

Axiom 1  $Trees \text{ produce their food. } (Tree \sqsubseteq \exists produce.Food)$

In the example, we can see that Axiom 1 is a semantically unknown sentence because it states *Trees* produce their *food* ( $Tree \sqsubseteq \exists produce.Food$ ), and the domain ontology only mentions that *Producers* produce their own *food* ( $Producer \sqsubseteq \exists produce.Food$ ). From the ontology, we cannot determine whether the sentence is true or false.

In our previous work on a *precomputed* approach [10], determining whether a sentence is semantically unknown is not practical, because its implementation leads to an overlap with determining whether a sentence is semantically incorrect. In contrast, under the *online reasoning* approach, identifying a sentence as unknown becomes an effect of verifying if a sentence is semantically correct ( $\mathcal{O} \models s$ ) or if it is semantically incorrect ( $\mathcal{O} \models \neg s$ ).

## 4.2 Ontology Consistency and Completeness

As seen in the preceding section, a sentence type depends on the relationship between a sentence and the domain ontology. In order for the ontology to be able to help us determine the semantic correctness of a sentence, it must meet two requirements: consistency and completeness.

The first requirement, i.e., consistency, is the most important one. In general, a domain ontology is expected to be



consistent (i.e., to have no logical contradictions). A consistent domain ontology provides an unambiguous taxonomic classification of concepts and relationships of a domain. If the ontology is inconsistent, there is a concept or relationship that has two or more irreconcilable interpretations (e.g., disjoint concepts stated in a *ISA* relationship). An inconsistent ontology is not only problematic in terms of utility. From a more theoretical point of view, an inconsistent ontology is seen as useless, since anything can be inferred from a set of contradicting axioms [14]. So, consistency of the domain ontology is fundamental for determining the correctness of a sentence.

Although it can be argued that methods, such as *inconsistent reasoner* by Huang et al. [19], can manage logic contradiction in an ontology, its application to semantic error detection is not straightforward. A sentence might be consistent with part of the ontology and inconsistent with another part of it, or it might represent the fact that creates the contradiction in the ontology. For the present work, the domain ontologies used for semantic error detection are consistent.

On the other hand, the requirement of completeness of the domain ontology has more practical implications. If the domain ontology only reflects a section of the domain, it is more likely that the analysis by our method will label some sentences as unknown, although they are semantically correct or semantically incorrect. The more complete the domain ontology, the more accurate the analysis of the text.

Although it might seem simple to address the issue of completeness of an ontology, e.g., ontology population [5], to make the ontology more complete, change can easily lead to inconsistency (Sect. 2.2.1). Ontology completeness might be difficult to address, since it can lead to a need to redesign an ontology. For the present work, we will make the assumption that the domain ontology is complete regarding the information referenced in the analyzed documents.

### 4.3 Determining the Correctness of a Sentence

After a sentence has been transformed from its written form into its logic representation which is compatible with the domain ontology, we analyze the sentence for semantic correctness.

We start by determining whether an extracted statement  $s$  is correct by entailment (Algorithm 1, line 2). If the extracted statement can be entailed, it is labeled as *correct*. If it cannot be entailed, the statement is added to the ontology to determine its consistency (Algorithm 1, line 3). If the domain ontology becomes inconsistent after an extracted sentence is added to it, then the sentence is *incorrect*. If the extracted statement is not entailed by the domain but consistent with it, the statement is labeled as *unknown* (i.e., incomplete) with respect to the domain ontology. In this work, we have selected *HermiT* as the reasoner, because of its higher effi-

ciency which was obtained by using hypertableau reasoning algorithm [27].

```

1 Input:  $s, \mathcal{O}$ 
2 if  $\mathcal{O} \models s$  then
3   if  $\mathcal{O} \not\models \neg s$  then
4     |  $s$  is unknown
5   else
6     |  $s$  is incorrect
7   end
8 else
9   |  $s$  is correct
10 end

```

**Algorithm 1:** *Online reasoning* approach for semantic error detection in single-sentence analysis.  $s$  represents a sentence after it has been transformed from its written form into its logic representation.

In case of inconsistency (i.e., one or more incorrect sentences), we preferred that the error detection approach could provide an *explanation* of the origin of the inconsistency. For that purpose, we have included into our approach the ontology debugging solution proposed by Horridge et al. [17]. As previously mentioned, the *explanation* approach by Horridge et al. integrates Reiter's Hitting Set Tree (HST) [30] to identify the minimal inconsistent sub-ontology, i.e., the minimal subset of axioms from the ontology that cause the inconsistency. Since the inconsistency is originated by the sentence, the HST-based debugging method can determine which part of the ontology is contradicted by the incorrect sentence (i.e., the explanation). Horridge et al.'s approach has been incorporated into popular DL reasoners, such as *Pellet* [28] and *HermiT* [27].

## 5 Multiple-Sentence Analysis

In our previous approaches for semantic error detection, we analyzed individual sentences of the text [10,11]. Single-sentence analysis is based on the notion that a sentence is the smallest linguistic unit from which an IE system can extract information. However, because sentences are usually used to construct paragraphs and documents to express more complex ideas, they are dependent on each other. Although not all sentences of the same document are semantically *connected*, it is very likely that sets of sentences refer to the same concepts and relationships. Let us consider the following example:

Ontology  $Planet \sqsubseteq \neg DwarfPlanet$   
 Axiom 1  $Planet(Pluto)$   
 Axiom 2  $DwarfPlanet(Pluto)$

From the domain ontology, we only know that a *Planet* cannot be a *DwarfPlanet*. If we state that *Pluto* is a *Planet* (Axiom 1), we cannot label it as a semantically correct or incorrect statement. The same occurs with Axiom 2. In other words, if we apply any of our previous approaches to determining the semantic correctness of these two axioms, we would only discover that both axioms are unknown. However, it is clear that, given the domain ontology, these axioms together would make a document semantically incorrect.

In this work, we extend semantic analysis to consider a broader set of possible semantic errors in a text. We need to consider that sentences are not independent of each other; i.e., semantic errors can occur by combining two or more sentences. As in the example, these semantic errors become evident only when analyzing a set of sentences as a whole, and not as a series of independent sentences.

### 5.1 Analyzing All Sentences Simultaneously

Although it is possible that the multiple-sentence semantic errors affect all sentences of a text, it is more likely that a set of sentences can be domain inconsistent. But, because a set of semantically erroneous sentences can be formed with parts of any section of the text (e.g., domain inconsistency between sentences from different paragraphs), determining which set of sentences needs to be analyzed together becomes a difficult issue.

A simple approach would be to analyze all the sentences of a text together. This approach would avoid the complex task of determining which sentences need to be considered as a set to be analyzed. It also avoids the problematic situation of missing a set of semantically incorrect sentences by splitting them into different analysis sets.

However, by considering all sentences at a time, we lose some of the information that is obtained through consistency checking. As mentioned, consistency checking can only determine the consistency of the ontology and the set of sentences. In the case of the single-sentence analysis, we could determine that the analyzed sentence is semantically incorrect. However, if a set of sentences is inconsistent against the ontology, that means at least one sentence is semantically incorrect. We also cannot differentiate between semantically correct and unknown sentences, since both types are consistent with the domain ontology.

It can be argued that we could reduce the error detection problem to ontology debugging (e.g., apply a method such as Horridge et al. [17]). However, if we consider that the number of sentences to be analyzed could be large (analyzing a large document), this approach becomes impractical. Methods such as Horridge et al. [17], Schlobach and Cornet [33], and Schlobach et al. [34] need to perform multiple consistency checks. We can easily see that this approach becomes impractical when considering that consistency checking has

an exponential complexity in DL, and that the size of the ontology, combined with the extracted statements, is significantly large.

### 5.2 Analyzing an Incremental Set of Sentences

Alternatively, instead of analyzing all sentences at the same time, we can consider a subset of sentences, making the method more practical. However, if we do not generate the subset of sentences carefully, it is possible to partition the set of sentences in a way that could eliminate the actual semantic errors [17].

An option to analyze groups of sentences without overlooking a semantic error is by incrementally analyzing the set of statements. Iteratively, we add sentences into the ontology, and we perform consistency checking. If there is an inconsistency, we try to identify the origin. This incremental approach allows us to keep some control over the complexity of the process while still providing completeness throughout the analysis.

In this approach, a key element is the order in which the sentences are being added to the ontology for analysis. For example, we produce the set  $S = s_1, \dots, s_i, \dots, s_j, \dots, s_n$  (with  $i$  much smaller than  $j$ ) of extractions from sentences of a text. Let us assume that the inclusion into the ontology of statements  $s_i$  and  $s_j$  together makes it inconsistent. Then, since  $i$  is much smaller than  $j$ , in our incremental approach,  $s_j$  will be added many iterations after  $s_i$ . If we order the set of extracted sentences based on some similarity, such as Huang et al.'s *selection function* [19], the analysis with both of  $s_i$  and  $s_j$  can be performed earlier. Even though this efficient ordering of statements does not reduce the complexity of the consistency checking, it can reduce the complexity when trying to find the origin of the inconsistency.

The weakness of this approach is that it can easily degrade into the approach of analyzing all sentences simultaneously. As we iterate, the number of sentences to analyze will prevent us from determining which sentences in the subset are semantically correct, or which sentences from the large set are semantically incorrect.

### 5.3 Reduced Sentence Set

We proposed that for multiple-sentence analysis, the sentences that do not provide new information can be removed from the analysis process without losing information for the analysis. This assumption of reduction without loss of deduction capability is based on *cut-elimination* over entailed elements. *Cut-elimination* is the central inference rule in sequent calculus.

$$\frac{\Gamma \vdash \Delta, A \quad A, \Sigma \vdash \Delta}{\Gamma, \Sigma \vdash \Pi, \Delta}$$

*Cut-elimination* mainly states that if we can entail a logic formula  $A$  from a set of formulas  $\Gamma$ , we can entail other elements  $\Gamma \vdash \Delta$  without  $A$ , since it is already contained in  $\Gamma$ .

Based on *cut-elimination*, we could remove two types of sentences without affecting the completeness of our analysis approach: semantically correct sentences and semantically incorrect sentences. Since semantically correct sentences are consequences of the domain (i.e.,  $\mathcal{O} \models s_i$ ), they do not provide any information that is not already contained in the domain ontology. Similarly, semantically incorrect sentences are *false consequences* of the domain (i.e.,  $\mathcal{O} \models \neg s_i$ ).

### 5.3.1 Determining Sentence Types

```

1 Input:  $S = \{s_1, s_2, \dots, s_n\}, \mathcal{O}$ 
2  $i = 1$ 
3  $U$  set of unknown sentences
4 while  $i \leq |S|$  do
5   if  $\mathcal{O} \not\models s_i$  then
6     if  $\mathcal{O} \not\models \neg s_i$  then
7       if  $\mathcal{O} \cup U \cup s_i \models \perp$  then
8          $U \cup s_i$  is incorrect
9       else
10         $s_i$  is unknown and added to the set of unknown
11        sentences  $U$ 
12      end
13    else
14       $s_i$  is incorrect
15    end
16  else
17     $s_i$  is correct
18  end
19  $i = i + 1$ 
20 end

```

**Algorithm 2:** *Online reasoning* approach for semantic error detection in multiple-sentence analysis.  $s_i$  represents a sentence after it has been transformed from its written form into its logic representation.

As mentioned above, because sentence type can allow us to determine which sentence needs to be considered as part of a set of sentences for analysis, multiple-sentence analysis for online reasoning semantic error detection provides a generalized approach of our reasoning-based approach.

The reduction in the number of sentences occurs, as seen in Algorithm 2, by not including semantically correct and semantically incorrect sentences for the following iteration of the process. As it can be seen in line 6 in Algorithm 2, we only evaluate the consistency between unknown sentences. Since unknown sentences contain information that is not in the ontology, we need to consider them for following iterations.

### 5.3.2 Proof of Lossless Reduction

To use our proposed reduction method without losing information for multiple-sentence analysis, we must prove that in each iteration, some sentences allow the use of *cut-elimination*.

Let us consider the set of extracted relations  $S = s_1, \dots, s_n$  where  $s_i$  is an extracted sentence with  $i \in [1, n]$ ,  $S'$  is a subset of extracted relations that have already been analyzed ( $S' \subsetneq S$ ), and the domain ontology  $\mathcal{O}$ .

- $s_i$  is correct: Let us assume that  $s_i$  is a correct sentence, i.e.,  $\mathcal{O} \cup S' \models s_i$  is true. Then, to analyze  $s_{i+1}$  we must consider  $\mathcal{O} \cup S' \cup s_i \models s_{i+1}$ . Then through *cut-elimination*,  $\mathcal{O} \cup S' \cup s_i \models s_{i+1}$  can be reduced to  $\mathcal{O} \cup S' \models s_{i+1}$ .
- $s_i$  is incorrect: Let us assume that  $s_i$  is an incorrect sentence, i.e.,  $\mathcal{O} \cup S' \models \neg s_i$  is true. Similarly to the case of  $s_i$  being a correct sentence, we do not need  $\neg s_i$  to determine if  $s_{i+1}$  is a logical implication from the domain and the previous sentences. Then through *cut-elimination*,  $\mathcal{O} \cup S' \cup s_i \models s_{i+1}$  can be reduced to  $\mathcal{O} \cup S' \models s_{i+1}$ .
- $s_i$  is unknown: Finally, let us assume that  $s_i$  is an unknown sentence.  $\mathcal{O} \cup S' \models s_i$  and  $\mathcal{O} \cup S' \models \neg s_i$  are false. If  $\mathcal{O} \cup S'$  cannot entail  $s_i$  (previous axiom is true), then we cannot remove  $s_i$  for the analysis of  $s_{i+1}$  because  $s_i$  is not contained in  $\mathcal{O}$ .

We can see that  $S'$  contains all sentences that have been labeled as semantically unknown, because if we determine that a sentence is semantically correct (or incorrect), we do not need to consider it for the following analysis.

## 6 Evaluation

We have evaluated both the single-sentence analysis and the multiple-sentence analysis of our proposed *online reasoning* approach for semantic error detection. The following sections provide details of data sets, ontologies, and comparison methods used for each type of analysis.

### 6.1 Evaluating Single-Sentence Analysis

#### 6.1.1 Overview of the Data Set and Ontology

*Data Set* In this work, we will use a set of summaries collected on an earlier study by Sohlberg et al. [36] that looked at the use of electronic strategies (eStrategies) for reading comprehension of college students. As part of the study, students were asked to provide oral summaries of each of four articles they had read, where each article is roughly three pages in length. The oral summaries were manually transcribed into text form. From the Sohlberg et al. collection,

**Table 1** Statistical information about the ontology

Element type	Number of elements
Concepts	45
Relationships	28
Subclass relationships	7

we will consider for the present work 18 summaries from the Ecosystems article. The summaries vary in length from a pair of sentences to 60 sentences. A section of a summary from the Ecosystem set can be seen in the following example:

*In the ecosystem there are different types of animals.  
Producers make their own food from the environment.  
Consumers eat other consumers and producers.  
The producers are plants, trees, algaes.  
...*

The summaries have been preprocessed in order to simplify the extraction process. The preprocessing has been focused on resolving *anaphoras* and *cataphoras* (e.g., pronouns) and on correcting misspellings. The summaries have also been labeled at the sentence level, according to the correctness of their content. The labeled summaries have been used as the gold standard for the purpose of evaluation.

**Ontology** The ontology used for this evaluation is based on the same article used by the students for summarization. The construction of the ontology is constrained to explicit facts from the domain knowledge defined by the article and does not include facts from the entire domain of Ecosystems. By keeping our ontology centered on the introductory article, we intended that the ontology could better cover concepts and relationships from the students' summaries, which are also solely based on the article. Because of the strict construction criteria, the ontology has many concepts that do not have a membership relationship with another concept, as well as not having instances (Table 1; Fig. 2).

On the other hand, in order to determine semantic errors based on logic contradiction, the ontology for the present evaluation incorporates a large set of constraints, such as disjointness between classes, and strictly defines domain and range for each property. Because an ontology provides a representation of domain knowledge in the form of categorized information, disjointness can be easily identified in most domains, since it allows the separation between individuals that have been categorized in a specific way. For example, in the case of the Ecosystems data set, the article that was used as a guide for constructing the ontology states that individuals from one concept (e.g., Carnivores) differentiate from other individuals (e.g., Herbivores) because of a specific property (e.g., what they eat). This distinction forces that the individuals of a specific concept cannot be members of another concept.

### 6.1.2 Evaluation Metrics and Comparison Methods

**Evaluation Metrics** The performance of an IE system is measured with the metrics Precision, Recall, and F1 measure. Precision ( $P$ ) measures how much of the extraction is correct, and it is calculated by dividing the number of correct extractions or *true positives* ( $tp$ ) by the total number of extractions, which are *true positives* plus *false positives* ( $tp + fp$ ). Recall ( $R$ ) measures how complete the extraction is, and it is calculated by dividing the number of correct extractions ( $tp$ ) by the total number of instances that should be extracted, which are *true positives* plus *false negatives* ( $tp + fn$ ). Finally, the F1 measure ( $F_1$ ) provides a harmonic average of precision and recall.

$$P = \frac{tp}{tp + fp} \quad R = \frac{tp}{tp + fn} \quad F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

Although we could use evaluation metrics that take into account the semantic dependencies of the different elements that conform the domain, such as balanced distance metric (BDM) [25], because we are focused on determining whether a sentence is semantically correct or not, these traditional metrics are better suited for our evaluation.

**Comparison Methods** To obtain a better understanding of how well our *online reasoning* method performs, we are comparing the performance of our method against two comparison methods.

The first comparison method is our previously proposed *precomputed* semantic error detection approach [10], which is, to the best of our knowledge, the only ontology-based semantic error detection method. Our *precomputed* approach defines domain-inconsistent axioms as violating ontological constraints. These domain-inconsistent axioms are encoded into extraction patterns that can detect semantically incorrect sentences before the extraction process begins (i.e., precomputed approach). For comparison, we have used the same set of rules that were manually defined before. We created the extraction rules by using the domain ontology and considering the content documents. This resulted in 31 extraction rules to identify correct sentences, 16 extraction rules to identify incorrect sentences, and five extraction rules to identify unknown sentences.

The second comparison method is a variation of our *online reasoning* approach, in which we replace the IE process with *manual extraction*. This variation can provide us with insight into how the mapping and reasoning steps perform when analyzing correctness. Because currently available IE implementations are not 100% accurate, the overall performance of error detection might be affected by the IE process. We have constructed a data set formed by binary relationships manually extracted from the 18 summaries. These manually

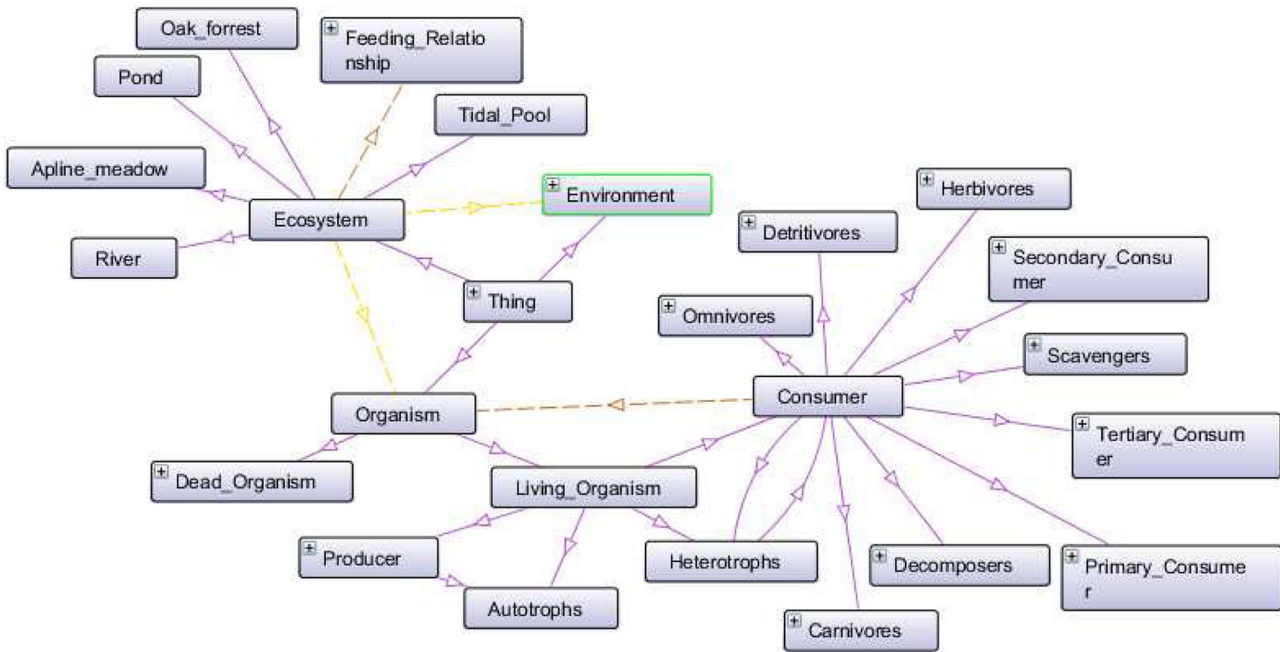


Fig. 2 Graphical representation of a section of the Ecosystems ontology

extracted relationships are then analyzed by our approach to determining their correctness.

### 6.1.3 Results

From Table 2, we can say that in the case of the *online reasoning* approach, it is possible to determine with high precision the semantic correctness of a sentence with respect to the domain by logic reasoning. However, there is a significant quantity of sentences that, although contained in the domain, are considered to be unrelated to the domain. There is a significant quantity of cases in which the IE process extracted phrases as entities. Although this is not strictly incorrect, most of these phrases represented something more than only a domain concept. This leads to a lower recall. On the other hand, although not all semantically correct and incorrect sentences were captured, the sentences that were labeled as correct are all semantically correct sentences. The same goes with the semantically incorrect sentences.

The *perfect precision* (i.e., 100%) observed in Table 2 obtained by both *online reasoning* and the *manual extraction* approaches in the case of semantically correct and incorrect sentences might seem unrealistic. However, it is the natural outcome, given the underlying method used in the process (i.e., reasoning). If one sentence were labeled as correct when it was actually incorrect, it would mean that the reasoning process used to determine the label of the sentence is not accurate. However, as previously mentioned, we are using a DL reasoner (i.e., *HermiT*) which is sound and complete. So, once the semantic elements of a sentence are mapped to the

Table 2 Precision (top), recall (center), and F1 measure (bottom) for the proposed method (automatic and manual extraction) and for the precomputed approach [10]

Sentence	Automatic extraction (%)	Manual extraction (%)	Precomputed approach (%)
Correct	100	100	91.9
Incorrect	100	100	97.4
Unknown	89.5	74.71	–
Correct	40.9	80.23	83.3
Incorrect	41.3	88.63	88.6
Unknown	100	100	–
Correct	58.1	89.0	87.4
Incorrect	58.4	93.97	92.8
Unknown	94.4	74.71	–

ontology, the reasoner can accurately determine whether it contradicts the domain ontology or not.

In the case of *manually extracted* relations, we can observe an increment in the recall with respect to the *online reasoning* approach, with the same level of precision. This result indicates that the quality of the extraction process has a significant effect upon the detection of correctness; yet, it is not the only factor affecting the recall of correct and incorrect sentences. In the case of *manual extractions*, the error in determining the correctness of a sentence can be explained by the mapping between extractions and ontology. The correct (and incorrect) sentences that were labeled as incomplete

are cases in which the mapping procedure failed to connect extraction entities with ontological concepts.

When compared with our previous approach, pre-computed error detection, both our proposed automatic extraction and manual extraction methods are more accurate when identifying incorrect sentences. On the other hand, because our previous approach seeks specific predefined patterns in the text, it has a higher recall. However, the *precomputed error* has higher deployment conditions (i.e., overhead), since the extraction rules need to be created by domain and ontology experts.

### 6.2 Evaluating Multiple-Sentence Analysis

We have also evaluated our *online reasoning* approach for multiple-sentence analysis. However, because multiple-sentence analysis is a new approach to semantic error detection, instead of evaluating the method, we provide some observations from the execution of this new approach over two synthetic data sets.

#### 6.2.1 Data Sets with Manually Inserted Errors

Currently, there are no existing data sets for semantic errors on multiple sentences. For this evaluation, we have manually generated two data sets that contain multiple-sentence semantic errors. We opted for a manual approach to generating the data set, instead of an automatic approach, because of the high complexity of the task, which does not always produce coherent text.

In order to reduce the level of bias in the generation of the data set, we have used an *off-the-shelf* IE system following a *black box* approach (i.e., the IE system has not been altered). We have also considered actual sentences as elements to be added to the data sets. Although these precautions cannot completely eliminate some bias in generation process, the amount of that bias should be limited.

**Ecosystem Data Set** We have also used the Ecosystem data set for evaluating our proposed multiple-sentence analysis. As mentioned, the Ecosystem data set consists of 18 oral summaries, manually transcribed. On the other hand, we have constructed a domain ontology based on the introduction article, and it contains axioms explicitly stated in the introduction article.

We have created ten sets of sentences that are domain inconsistent (i.e., semantically incorrect). However, each sentence of every set is semantically unknown when individually analyzed. These sentences make reference to concepts and relationships from the Ecosystems domain but are not mentioned in the Ecosystems article. Because they do not appear in the article, these concepts and relationships are not included into the domain ontology. Each set of sentences is

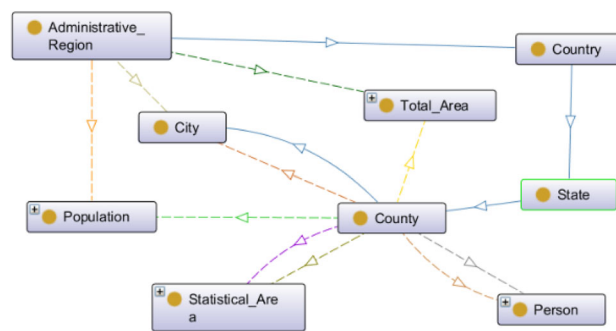


Fig. 3 Graphical representation of a section of the county ontology

Table 3 Statistical information about the ontology

Element type	Number of elements
Concepts	15
Relationships	6
Axioms	40

formed by a sentence that was originally in the text and by a sentence added into the text. We have added each set into a randomly selected summary.

**Wikipedia's County Data Set** It consists of 570 articles from Wikipedia regarding counties of the USA. These articles vary significantly in length, with some articles containing fewer than 10 sentences, while others contain more than 60.

We have designed an ontology by following patterns used for generating the Ecosystems ontology (e.g., explicitly stated relationships). Because the counties' articles had a very limited number of shared topics (e.g., origin of the name of the county), the ontology is small in comparison with other ontologies used for evaluation of semantic error (Fig. 3). However, it still has a large number of constraints (Table 3).

Currently, the county articles seem to be correct in terms of their content. So, to analyze the performance of our approach, we need modify the articles. The semantic errors introduced into each article are based on characteristics of the data set. There are 41 cases in which two or more counties from different states share the same name. The semantic error is introduced by adding sentences from one county to another county that has the same name. Because of constraints such as "a county can have one seat" and "it can belong to only one state," the inclusion of a sentence indicating another seat (or state) than the one in the article creates domain inconsistency across multiple sentences. We have chosen this approach to introducing semantic error because it is very likely that, at some point, these types of semantic errors might have occurred before the content was verified by Wikipedia editors.

**Table 4** Performance of multiple-sentence semantic error detection

	Ecosystem data set (%)	Wikipedia data set (%)
Precision	100	100
Recall	90	53.6
F1 Measure	94.7	69.7

### 6.2.2 Results

As mentioned, because semantic error detection over multiple sentences is a new approach, there are no comparison methods. However, we can still get some insight from the performance of the method. Table 4 presents the precision, recall, and F1 measure for identifying sets of semantically incorrect sentences.

In the case of the Ecosystem data set, the results are mostly a reflection of the performance of the single-sentence analysis (Table 4). If the sentence has been extracted and mapped correctly to the ontology, the multiple-sentence analysis method will accurately identify the semantically incorrect sentences (F1 measure 94.7%). When the transformation from text to logic clause fails, the sentences are labeled as unknown. As mentioned in Sect. 6.1.3, this transformation can fail because the IE process cannot identify the most relevant elements from a sentence, or because the mapping process did not manage to connect an extracted element to an element in the ontology. For example, from the sentence “Albert Einstein is a scientist of the 20th century” an IE system produces the tuple *is\_a* (*Albert Einstein*, *scientist of the 20th century*) instead of *is\_a\_scientist\_of*(*Albert Einstein*, *20th century*). Because the IE process produced a tuple centered on a verb that does not indicate the main relationship in the sentence, the tuple will seem not defined in the domain ontology, which makes the sentence unknown. Similarly, if the dictionaries used in the mapping process do not contain the mapping of an extracted instance (e.g., *was\_a\_scientist\_of*) to an element in the domain ontology (e.g., *is\_a\_scientist\_of*), the instance will be labeled as unknown because it is not defined in the ontology.

In the evaluation, a type of mapping issue that occurred was related to a negation in a sentence. Although information extraction systems can handle negation in most cases, it is not clear to which element in the ontology it should map. Because most DL languages cannot handle complex negation of concepts, we have negation mostly used in ontologies to define disjointness between concepts. Let us consider the concept *Carnivore* from the Ecosystem ontology, which is disjoint with a set of concepts. It is unclear whether the statement *no\_Carnivore* refers to all of the concepts that are disjoint to *Carnivore*, or if it refers to a specific concept like *Herbivore*.

In the case of the Wikipedia data set, the semantic error detection was not as accurate as with the Ecosystem data set

(Table 4). This result was mostly caused by the transformation process, which was affected by the complexity of the text and the broad content of the documents. The first issue refers to how the content is presented. Most articles in the Wikipedia data set present content with elaborated sentences (e.g., compound sentences and anaphora). This made the extraction process produce lower-quality extraction (e.g., incorrectly labeled entity extracted). The second issue refers to the wide range of topics included within each document. All articles from the data set provide general descriptive information, such as the location of the county, and they all provide some demographic information. Some articles provide detailed historic information, while others provide economic or other relevant information from the county. This diversity of content has adversely affected the creation of an adequate mapping for the transformation stage of our process, which led to incorrectly mapped concepts and relationships.

## 7 Conclusions and Future Work

In this work, we propose a new, reasoner-based approach for semantic error detection in text. Our approach transforms sentences into logic clauses through an IE process. A logic reasoner determines whether the extracted logic clauses are consistent with the domain or not. We can analyze the semantic correctness of sentences individually, or as a set, providing a broader understanding of the text. To keep the multiple-sentence analysis at a practical level of complexity, we have proposed a set of *reduction rules*, which minimize the number of sentences to be analyzed by the reasoner. Our experiments have shown that, although it is possible to determine the correctness of a text with high precision, the recall of the analysis can be low. We have also observed that the quality of the underlying IE can affect the detection process.

Based on the results obtained in this work, we have identified three topics that we would like to explore in more detail as future work. The first topic refers to the improvement of IE process regarding identifying entities. To improve the transformation of sentences into logic clauses, we need to obtain a more precise extraction process regarding identifying concepts and individuals. We believe that this aspect of the extraction can be improved by considering semi-supervised IE [21, 37] and Name Entity Linking (NEL) [21]. Of the same nature as the first topic, the second topic refers to finding better mechanisms to define mappings between the vocabulary of the text and the vocabulary of the ontology. We believe that this aspect of our method can be automated by the inclusion of text-processing methods, such as dependency parsing and coreference resolution.

Finally, the third topic refers to identifying a more efficient method to determine the origin of a semantic error. Although current ontology debugging methods can provide

tentative solutions to this problem, they have both different focuses and different parameters for finding the origin of inconsistency. We believe that ontology debugging methods can be improved for semantic error detection by considering the semantic elements of the sentences. The use of sentences can provide a focused search in the debugging process.

**Acknowledgements** This research is partially supported by the National Science Foundation Grant IIS-1118050 and Grant IIS-1013054. This research is also partially supported by the Fondo Nacional De Ciencia Y Tecnologia (FONDECYT), Chile, Grant No. 3170971. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the NSF or FONDECYT.

## References

1. Bechhofer S, van Harmelen F, Hendler J, Horrocks I, Patel-Schneider PF, McGuinness D, Stein L (2004) OWL Web Ontology Language. <http://www.w3.org/TR/owl-ref/>
2. Bos J (2008) Wide-coverage semantic analysis with boxer. In: Proceedings of the 2008 conference on semantics in text processing, STEP '08. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 277–286
3. Bos J (2015) Open-domain semantic parsing with boxer. In: Proceedings of the 20th Nordic conference of computational linguistics, NODALIDA 2015, May 11–13, 2015, Institute of the Lithuanian Language, Vilnius, Lithuania, pp 301–304
4. Buitelaar P, Cimiano P, Racioppa S, Siegel M (2006) Ontology-based information extraction with SOBA. In: Proceedings of the international conference on language resources and evaluation (LREC). ELRA, pp 2321–2324
5. Carlson A, Betteridge J, Hruschka ER, M., M.T (2009) Coupling semi-supervised learning of categories and relations. In: Proceedings of the NAACL HLT 2009 workshop on semi-supervised learning for natural language processing (SemiSupLearn), pp 1–9
6. Fader A, Soderland S, Etzioni O (2011) Identifying relations for open information extraction. In: Conference on empirical methods in natural language processing (EMNLP), pp 1535–1545
7. Flouris G, Manakanatas D, Kondylakis H, Plexousakis D, Antoniou G (2008) Ontology change: classification and survey. *Knowl Eng Rev* 23(02):117–152
8. Fuchs NE, Kaljurand K, Schneider G (2006) Attempto controlled English meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In: Sutcliffe G, Goebel R (eds) FLAIRS conference. AAAI Press, pp 664–669
9. Gruber TR (1993) A translation approach to portable ontology specifications. *Knowl Acquis* 5(2):199–220
10. Gutierrez F, Dou D, Fickas S, Griffiths G (2012) Providing grades and feedback for student summaries by ontology-based information extraction. In: Proceedings of the 21st ACM conference on information and knowledge management (CIKM), pp 1722–1726
11. Gutierrez F, Dou D, Fickas S, Griffiths G (2014) Online reasoning for ontology-based error detection in text. In: Proceedings of the 13th international conference on ontologies, databases and application of semantics (ODBASE), pp 562–579
12. Gutierrez F, Dou D, Fickas S, Martini A, Zong H (2013) Hybrid ontology-based information extraction for automated text grading. In: Proceedings of the 12th IEEE international conference on machine learning and applications (ICMLA), pp 359–364
13. Gutierrez F, Dou D, Fickas S, Wimalasuriya D, Zong H (2015) A hybrid ontology-based information extraction system. *J Inf Sci* 42:798–820
14. Haase P, Völker J (2008) Ontology learning and reasoning—dealing with uncertainty and inconsistency. In: Costa PC, D'Amato C, Fanizzi N, Laskey KB, Laskey KJ, Lukasiewicz T, Nickles M, Pool M (eds) Uncertainty reasoning for the semantic web I, pp 366–384
15. Hearst MA (1992) Automatic acquisition of hyponyms from large text corpora. In: Proceedings 14th conference on computational linguistics (COLING), pp 539–545
16. Hina S, Atwell E, Johnson O (2010) Secure information extraction from clinical documents using snomed ct gazetteer and natural language processing. In: 2010 International conference for internet technology and secured transactions, pp 1–5
17. Horridge M, Parsia B, Sattler U (2009) Explaining inconsistencies in OWL ontologies. *Scalable Uncertain Manag* 5785:124–137
18. Horrocks I, Patel-Schneider P (2004) Reducing owl entailment to description logic satisfiability. *Web Seman Sci Serv Agents World Wide Web* 1(4):345–357
19. Huang Z, van Harmelen F, ten Teije A (2005) Reasoning with inconsistent ontologies. In: Proceedings of the 19th international joint conference on artificial intelligence (IJCAI), pp 454–459
20. Kaljurand K, Fuchs NE (2007) Verbalizing OWL in Attempto Controlled English. In: Proceedings of third international workshop on OWL: experiences and directions, Innsbruck, Austria (6th–7th June 2007), vol 258
21. Koch M, Gilmer J, Soderland S, Weld DS (2014) Type-aware distantly supervised relation extraction with linked arguments. In: Conference on empirical methods in natural language processing (EMNLP)
22. Kuhn T (2007) AceRules: executing rules in controlled natural language. In: Massimo Marchiori CdSM, Pan JZ (eds) Proceedings of the first international conference on web reasoning and rule systems (RR2007), Lecture notes in computer science. Springer
23. Landauer TK, Laham D, Foltz PW (1998) Learning human-like knowledge by singular value decomposition: a progress report. In: Proceedings of the conference on advances in neural information processing systems (NIPS), pp 45–51
24. Mausam Schmitz M, Soderland S, Bart R, Etzioni O (2012) Open language learning for information extraction. In: Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp 523–534
25. Maynard D, Peters W, Li Y (2006) Metrics for evaluation of ontology-based information extraction. In: WWW workshop on evaluation of ontologies for the Web' (EON), Edinburgh, Scotland, UK
26. Mintz M, Bills S, Snow R, Jurafsky D (2009) Distant supervision for relation extraction without labeled data. In: Proceedings of the joint conference of the 47th annual meeting of the association for computational linguistics (ACL) and the 4th international joint conference on natural language processing of the AFNLP, pp 1003–1011
27. Motik B, Shearer R, Horrocks I (2009) Hypertableau reasoning for description logics. *J Artif Intell Res* 36:165–228
28. Parsia B, Sirin E (2004) Pellet: an OWL DL Reasoner. In: 3rd international semantic web conference (ISWC)
29. Presutti V, Draicchio F, Gangemi A (2012) Knowledge extraction based on discourse representation theory and linguistic frames. In: Proceedings of the 18th international conference on knowledge engineering and knowledge management, EKAW' 12, pp 114–129. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-33876-2\_12
30. Reiter R (1987) A theory of diagnosis from first principles. *Artif Intell* 32(1):57–95



31. Ritter A, Downey D, Soderland S, Etzioni O (2008) It's a contradiction—no, it's not: a case study using functional relations. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), pp 11–20
32. Saggion H, Funk A, Maynard D, Bontcheva K (2007) Ontology-based information extraction for business intelligence. *The Semantic Web*, pp 843–856
33. Schlobach S, Cornet R (2003) Non-standard reasoning services for the debugging of description logic terminologies. In: Proceedings of the 19th international joint conference on artificial intelligence (IJCAI), pp 355–362
34. Schlobach S, Huang Z, Cornet R, van Harmelen F (2007) Debugging incoherent terminologies. *J Autom Reason* 39(3):317–349
35. Smith A, Osborne M (2006) Using gazetteers in discriminative information extraction. In: Proceedings of the tenth conference on computational natural language learning, CoNLL-X '06. Association for Computational Linguistics, Stroudsburg, PA, USA, pp 133–140
36. Sohlberg MM, Ehlhardt L, Fickas S, Sutcliffe A (2003) A pilot study exploring electronic mail in users with acquired cognitive-linguistic impairments. *Brain Injury* 17(7):609–629
37. Surdeanu M, Tibshirani J, Nallapati R, Manning CD (2012) Multi-instance multi-label learning for relation extraction. In: Proceedings joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp 455–465
38. Wimalasuriya DC, Dou D (2010) Ontology-based information extraction: an introduction and a survey of current approaches. *J Inf Sci* 36:306–323