Online Reasoning for Ontology-Based Error Detection in Text

Fernando Gutiererz[1], Dejing Dou[1], Stephen Fickas[1], and Gina Griffiths[2]

[1] Computer and Information Science Department
University of Oregon
Eugene, Oregon 97403, USA
{fernando,dou,fickas}@cs.uoregon.edu
[2] Special Education and Clinical Sciences Department
University of Oregon
Eugene, Oregon 97403, USA
ginag@uoregon.edu

Abstract. Detecting error in text is a difficult task. Current methods use a domain ontology to identify elements in the text that contradicts domain knowledge. Yet, these methods require manually defining the type of errors that are expected to be found in the text before applying them. In this paper we propose a new approach that uses logic reasoning to detect errors in a statement from text online. Such approach applies Information Extraction to transform text into a set of logic clauses. The logic clauses are incorporated into the domain ontology to determine if it contradicts the ontology or not. If the statement contradicts the domain ontology, then the statement is incorrect with respect to the domain knowledge. We have evaluated our proposed method by applying it to a set of written summaries from the domain of Ecosystems. We have found that this approach, although depending on the quality of the Information Extraction output, can identify a significant amount of errors. We have also found that modeling elements of the ontology (i.e., property domain and range) likewise affect the capability of detecting errors.

Keywords: Information Extraction, Ontology, Consistency Checking.

1 Introduction

As Information Extraction, the task of automatically identifying entities and relationships in text, moves from the analysis of scientific documents to Internet textual content, we cannot rely completely on the assumption that the content of the text is correct. Indeed, in contrast to scientific documents, which are peer reviewed, Internet content is not verified for the quality and correctness. So, it seems reasonable to consider, as part of the process of extracting textual information, mechanisms and techniques that allow to differentiate between correct and incorrect information.

However, it is not easy to determine the correctness of a sentence; hence, it has been addressed only indirectly. Research from the educational oriented field

R. Meersman et al. (Eds.): OTM 2014, LNCS 8841, pp. 562–579, 2014.

such as automatic text grading has mainly treated incorrectness as low similarity to a gold standard. Automatic grading systems based on Latent Semantic Analysis [13] and *n-gram* co-occurrence [14] try to determine how similar a student's summary or essay is with respect to a *perfect* summary or essay. If the student's writings has low similarity to the gold standard, it means that it is *less correct*. However, low similarity can still be obtained in the process of a correct text, such as if the text was written in an unexpected fashion, or if the text content is broader than the gold standard. On the other hand, incorrectness can be identified in the presence of contradiction. The research area of Contradiction Detection [20], an extension of Textual Entailment, intends to identify in text pair of sentences that cannot be true at the same time (i.e., logic contradiction). By identifying specific lexical and syntactical elements, the semantics of the sentences are captured and compared. However, since Contradiction Detection is limited to the content of the text itself in order to support the correctness of the contradicting sentences, it cannot determine with certainty which sentence of the pair is false.

Following a different approach, we have proposed an ontology-based error detection method using pre-defined errors for extraction rules [5] and machine learning based patterns [6]. An ontology provides a formal knowledge representation of a domain through concepts, relationships, and axiomatic constraints. By including a domain ontology into our approach, we have a formal model that allows the use of logic reasoning services, such as contradiction detection, plus a set of correct facts from the domain (e.g., relationships between concepts of the domain). So, if a sentence contradicts the domain ontology, then the 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 incorrect sentences based on the domain, it is limited by the set of expected errors defined in our heuristic algorithm. Such heuristic algorithm has a set of manually defined rules that generate axioms by violating constraints in the domain ontology. For example, if the domain ontology defines that concept A and B cannot share elements (i.e., they are disjoint), then the heuristic algorithm will generate an axiom of elements that result from the union of concepts A and B. This inconsistent axioms is encoded into an extraction pattern that identifies incorrect sentences that state that elements of concept A are also elements of concept B. In this way our previous method was able to identify errors (i.e., incorrect statements). However, this is also its main limitation. Our previous approach could only recognize incorrect sentences if they were part of the training set or very similar to a sentence in the training set. New sentences could not be judged correctly.

In this paper, we propose a new method to detect incorrectness by online inference. This method considers incorrectness as inconsistency with respect to the domain, but instead of defining a set of expected error rules, we incorporate a reasoner to determine if a sentence is inconsistent with the domain ontology. To this end, we apply a variation of an Ontology-based Information Extraction

(OBIE) process. OBIE is a subfield of Information Extraction that incorporates a domain ontology to guide the extraction process. In our approach, instead of having the ontology guiding the extraction process, the information extraction is performed based on structural elements from the text, while the ontology is used to validate the correctness of the extracted statements. Although this approach differs from the definition of OBIE [24], we argue that it is still an OBIE process since the approach relies on the domain ontology to determine the correctness of each statement. The main goal of this approach is to provide the most complete analysis of the correctness of a text. In that sense, the extraction process intends to extract every possible relationship in the text, while the inconsistency analysis use the complete ontology to infer the correctness or incorrectness of the extracted relations. In the evaluation of our proposed method, we found that our method can identify accurately the classification of a significant amount of sentences. We also found that the quality of the IE process affects the overall performance of our proposed method.

The rest of the paper is organized as follows. We provide a brief review of some related work in Section 2. Then we demonstrate our ontology-based error detection method in Section 3. We report our experimental results in Section 4, and discuss some observations from our case study in Section 5. We conclude the paper by summarizing our contributions and future goals in Section 6.

2 Background

In this work, we propose a method to identify incorrectness in text by combining logic reasoning and domain knowledge. This has led us to consider research regarding Information Extraction, Consistency Checking and Ontology Debugging to provide us with ideas that can help us to reach our goal.

2.1 Information Extraction

Information Extraction (IE) is the task of automatically acquiring knowledge by transforming natural language text into structured information, such as a knowledge base [24]. In the process of extracting, IE attempts to retrieve specific elements from text such as concepts, relationships, or instances. For example, from the sentence "Albert Einstein was born in Ulm," an IE process will identify *Albert Einstein* and *Ulm* as relevant instances that are connected by the relationship *born_in*. This leads to the extraction of the relationship *born_in*(*Albert Einstein, Ulm*). However, this transformation of information is in general not a trivial process because of the inherent ambiguity of natural language text. A fact can be stated in many ways, and a statement can have more that one interpretation. The complexity of extracting information from text has kept IE from being more widely adopted, with most IE systems being implemented only for specific domains.

The complexity of extracting information from text can be mitigated by the inclusion of domain knowledge in the process of capturing the semantic elements

from the text. The subfield of Ontology-based Information Extraction (OBIE) uses a formal representation of the domain (i.e., ontology) to guide the extraction process [24]. The guide offered by the domain ontology allows OBIE to focus on the extraction of specific concepts and relationships of the domain.

By considering the amount of human intervention in the preparation required for the system's deployment, three different strategies have been proposed to accomplish this transformation: *supervised*, *semi-supervised*, and *unsupervised*. A *supervised* IE systems require significant amount of intervention before being utilized. This intervention comes either in the form of labeled data in the case of IE system that are based on machine learning, or handcrafted patterns in the case of extraction rules [5]. Since the extraction process is tuned to focus on specific entities and relationships, most OBIE systems follow a supervised strategy. Ontological elements are included in the creation of extraction patterns and in the labeling of data. Because of this close labeling process, the uncertainty about the extracted relationship and what it represents is small. The close labeling process also allows supervised systems to extract implicit relationships, which in most cases cannot be identified by semi-supervised or unsupervised systems.

In the case of *semi-supervised* IE systems, the extraction process uses the connection between sentences and hand built knowledge bases to learn information extraction models and patterns for specific relationships. In contrast with supervised systems which have an explicit link between text and instances, semi-supervised systems have to discover this connection. In some cases the connection is clear, for example *Kylin* system exploits the relation between Wikipedia's *infoboxes* and articles [25]. On other cases, where the corpus of text is the Web [17], or the text is in a different language than that of the knowledge base [2], the connection is not that evident.

Semi-supervised system will consider a known relationship between a pair of entities (e.g., *place_of_birth(Immanuel Kant, Königsberg)*) or a known instance of a concept (e.g., *country(United States)*). This known tuple can be obtained from a knowledge based (e.g., DBpedia) or similar (e.g., Wikipedia's infoboxes). The system then will search in the text for sentences containing the entities from the relationship, such as "Immanuel Kant was born in 1724 in Königsberg, Prussia, as the fourth of nine children (four of them reached adulthood)." The selected sentences will then be transformed into vectors and enriched with grammatical information (e.g, *part-of-speech*). The transformed sentences will be used as training data for machine learning methods, such as Conditional Random Fields [25]. The learned machine learning models are then used to extract new instances and relationships.

Finally, *unsupervised* IE system can perform extractions without requiring labeled data or specific pattern construction. They used general patterns based on lexical and syntactical features from the text to identify relationship between entities. Initially these patterns could only identify *hyponymy* relationships [9]. However, continuous extensions to these patterns by the inclusion of more

complex features (e.g., syntactic dependencies [15]) has led to unsupervised IE systems to extract a wide variety of relationships [3].

One of the unsupervised extraction patterns proposed by Hearst [9] is "NP_0, ${NP_i}^{i=1...n}$ (*and/or*) *other* NP," which allows the extraction of the relation $[hyponymy(NP, NP_i)]^{i=0...n}$. For example, when the previous pattern is applied to the sentence "France, Italy, and other European countries..." we obtain the relationships *hyponymy*(*France, European_country*) and *hyponymy*(*Italy, European_country*).

2.2 Consistency Checking

Through concepts, relationships, and constraints, an ontology provides a vocabulary and a model of the domain it represents. In this work, we consider Description Logic based ontologies, as those described through the Web Ontology Language (OWL) [1]. OWL is the standard ontology language proposed by the World Wide Web Consortium (W3C).

Description Logic (DL) is a fragment of first-order logic that is decidable, and it has sound and complete reasoners, such as HermiT [18]. DL employs Boolean constructors, such as conjunction ($\sqcap$), disjunction ($\sqcup$), existential ($\exists$) and value ($\forall$) restrictions, to describe concepts and roles (i.e., relationships). DL consists of a *TBox* $\mathcal{T}$, which is a set of *general concept inclusions* (GCI) of the form $C \sqsubseteq D$, for concepts C and D. It also has a set $\mathcal{A}$, called *ABox*, of concept and role assertions of the form $C(a)$ and $R(a, b)$, for concept C, role R and individuals a and b. In some DL languages, there is an *RBox* $\mathcal{R}$ that consists of complex role constructions such as role inclusion ($R_1 \sqsubseteq R_2$), role disjointess , reflexivity, and others. A knowledge base $\mathcal{K}$ in DL is conformed as the tuple $(\mathcal{R}, \mathcal{T}, \mathcal{A})$. In the case of DL languages that do not have *RBox*, such as $\mathcal{ALC}$, the tuple takes the form $(\emptyset, \mathcal{T}, \mathcal{A})$ or $(\mathcal{T}, \mathcal{A})$.

The semantics of a DL knowledge base $\mathcal{K}$ is defined by an interpretation $I = (\Delta^I, \cdot^I)$. The interpretation consists of a domain (Δ^I) and a mapping function ($\cdot^I$). The function maps the concepts, roles and assertions of the knowledge base to the domain. If I satisfies all axioms of $\mathcal{K}$, then I is a model of $\mathcal{K}$, which makes $\mathcal{K}$ consistent. A basic service of a DL reasoner is to determine if a knowledge base $\mathcal{K}$ is satisfiable. Determining satisfiability is fundamental since other types of inferences can be reduced to unsatisfiability, such as entailment [11]. A DL reasoner demonstrate the satisfiability of a knowledge base by constructing an abstract model through a tableau algorithm. The algorithm creates a sequence of *ABoxes*, where each new *ABox* ($\mathcal{A}_i$) is the result of the application of derivation rules to a previous *ABox* ($\mathcal{A}_{i-1}$) [18]. 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 $C(t)$.

Through the derivation rules, the consistency checking algorithm can terminate by reaching one of two possible cases. In the first case, if the derivation leads to a contradiction, the knowledge base $\mathcal{K}$ is determined to be inconsistent, and the algorithms terminates. The second case, if no derivation rule can be applied, then the algorithm terminates, and the derived *ABox* ($\mathcal{A}_i$) represents a model for the knowledge base $\mathcal{K}$ (i.e., $\mathcal{K}$ is consistent).

2.3 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 the inconsistency. There are three main approaches when dealing with inconsistency [8]: preventing ontology inconsistency [7], fixing ontology inconsistency [10,21], or reasoning with inconsistency [12]. We believe that research in fixing ontology inconsistency can provide some insight in incorrectness detection in text because both tasks focus on determining logical contradiction and its origin.

The process of fixing an inconsistent ontology is known as *Ontology Debugging* [4]. The task of Ontology Debugging has been addressed using properties of the underlying description logic (DL) language. These methods try to identify the minimal inconsistent sub-ontology, which is the set of axioms that make the ontology inconsistent. Horridge et al. [10] 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*).

In order to find the set of all the inconsistent sub ontologies, Schlobach and Cornet [21] proposed to determine the *minimal unsatisfiability-preserving sub-TBoxes* (MUPS) of unsatisfiable concepts of the ontology. The *MUPS* of a concept is the set of axioms that cause the concept to be unsatisfiable. In their original work, Schlobach and Cornet [21] obtained the MUPS of each concept through a modified $\mathcal{ALC}$ reasoner that inserts traceable labels in the axioms when performing consistency checking. But because this approach does not scale well to more expressive DL languages, Schlobach et al. [22] offer an alternative mechanism to identify each concept MUPS. Based on Huang et al. selection function [12] for reasoning with inconsistent ontologies, Schlobach et al. used an informed search to find concept-relevant axioms. The set produced by the selection function is then pruned by removing axioms that do not affect the unsatisfiability of the concept MUPS.

On the other hand, in Horridge et al.'s approach [10], the inconsistent subsets of the ontology are obtained by dividing the ontology and testing the consistency of the partition. The intuition is that the cause of inconsistency will be in the inconsistent part of the ontology, and not in the consistent part. It is important to note that it is possible to remove accidentally the inconsistency when dividing the ontology. To avoid missing an inconsistent subset, Horridge et al. approach also analyzes the recombination of the divided parts.

Once the set of all inconsistent sub ontologies are obtained, the minimal inconsistent sub ontology can be identified. In the case of Schlobach and Cornet's approach, from the *MUPS* the *minimal incoherence-preserving sub-TBoxes* (MIPS) are determined, which are unsatisfiable sub-TBoxes that can become satisfiable if one atomic concept is removed. Because each element of the MIPS set comes from some MUPS, the MIPS set is contained in the union of all MUPS of the original *TBox*. Although the *MIPS* already identifies the minimal inconsistent set of the ontology, Schlobach and Cornet's approach offers an even more fine grained solution. Because inconsistencies can be interpreted as the effect of overspecified concepts [21], it is possible to identify the actual concepts that are clashing by generalizing the axioms of the MIPS. This new set is obtained by the *generalized incoherence-preserving terminology* (GIT), where all elements in an axiom of the MIPS which do not affect its unsatisfiability are removed.

On the other hand, Horridge et al. used the Hitting Set Tree algorithm [19] to identify the minimal inconsistent sub ontology from the set of inconsistent sub ontologies. Reiter proposed the Hitting Set Tree (HST) [19] as a form to determine the diagnosis a faulty system. In a faulty system there can be multiple reasons that explain the actual error (i.e., *conflict sets*). Yet in order to correct or fix the system, it is necessary to identify the minimal conflict set (i.e., diagnosis). Reiter's HST finds the diagnosis by learning how the conflict sets intersect. The HST algorithm iteratively searches or accesses the set of conflict sets. It constructs a tree where each node indicates a conflict set and the edges indicate an element of the conflict set. The set formed by the labels on the edges along a branch of the HST corresponds to one diagnosis. In the case of ontology inconsistency, the HST identifies the minimal inconsistent sub ontology by constructing a tree that has as nodes the inconsistent sub ontologies.

These methods can help in the task of incorrectness detection in text by providing mechanisms to determine the reason a sentence is incorrect with respect to the domain ontology.

3 Methodology

In the present work we propose online incorrectness inference, a method to detect logic errors of text content by incorporating logic reasoning and domain knowledge. The error detection comes from determining if the text is logically consistent with respect to the modeled domain knowledge (i.e., ontology). The consistency of the text can be checked by adding sentences, as extracted entities and relationships, to the domain ontology and verifying its consistency.

The online incorrectness inference approach consists of three steps. In the first step, statements are extracted from the text through an Information Extraction process. The extraction process provides a transformation of the statement from its original textual form into a logical bound form. As a second step, the extracted statements are formalized and added to the domain ontology. The formalization intends to transform the extracted tuples into a form compatible with the ontology. The third and final step is to determine the correctness of the statement

through a logic-based service provided by an ontology: consistency checking. If the domain ontology becomes inconsistent after the extracted sentence is added into it, then the sentence is incorrect with respect to the domain.

In the following sections we provide details of each one of the steps of our proposed method.

3.1 Information Extraction

In order to evaluate the correctness of a sentence, we first transform unstructured text into structured information. As previously mentioned, there are three main strategies to IE depending on the level of human intervention (i.e., data preparation). However, because our approach intends to determine the correctness of each statement 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* [15]. Open Information Extraction systems intend to extract binary relationships without using any training data (or handcraft 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 [3]. 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 Open Information Extraction system *ReVerb* [3], which uses a 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 in 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 [15]. 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., the relation is in between the two entities in the sentence), a pure syntactic pattern can be learned from the sentence (e.g., 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.

Because we focus on determining the correctness of the text's content, we will consider *OLLIE* as a blackbox component of our system. *OLLIE* will produce a set of binary relationships from the text, which will be passed to the reasoner.

For example, from the sentence "Scavengers feed from dead organisms," OLLIE will produce the tuple *feed(Scavengers, dead organism)*.

3.2 Transformation of Extracted Relationships

Before adding the extracted relationships to the ontology to determine their correctness, we need to solve first a lexical gap that might exist between the text and the 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 will need a mapping mechanism that can allow us pass from the vocabulary of the extracted entities and relationships to the vocabulary of the ontology.

In the case of relationships at the terminology level of the domain (i.e., *TBox*), the mapping can be accomplished with simple mechanisms, such as gazetteers (i.e., list of words) of ontological terms, or through *WordNet* to find synonyms of terms [16]. However, in the case of extracted relationships from the assertion level of the domain (i.e., *ABox*), the solutions might not be so trivial. Most domain ontologies have very few individuals, if any at all. Because the domain ontology provides a general representation of the domain, it is very unlikely that it contains a large number of specific examples. We believe that the case of managing extracted relationships from the assertion level will require more sophisticated techniques, such as reasoning in the presence of uncertainty.

In the present work, because we are dealing with the writings on the generalization of domain knowledge (i.e., summaries), it seems reasonable to consider gazetteers for mapping vocabularies. We have defined two gazetteers: one for managing concepts, and another for managing relationships. In the case of the gazetteer for managing concepts, an extracted entity will lead to the equivalent ontological concept. For example, 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 *Carnivore* and the relation *feed* will lead to the property *feed_from_herbivore*, while concept *Herbivore* and relation *feed* will lead to the property *feed_from_producer*. Both gazetteers are generated by considering a subset of extracted relationships from the data set.

3.3 Determining Correctness of a Sentence

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 statements by using consistency checking. We propose the use of a logic reasoner to determine the correctness of text.

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. Although the analysis of multiple sentences leads to a higher computational complexity, it allows us analyze the correctness between sentences. There are cases where sentences can be consistent when considered independently, but become inconsistent when analyzed as a set.

In this work, we focus on perform 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. Multiple sentence analysis will be discussed in Section 6.

Type of Sentence. In the previous work, we identify three type of sentences [5]: correct, incorrect, and incomplete. In the present work, we offer a more precise definition of them.

A sentence is considered to be *correct* if it can be entailed from the domain. We argue that consistency is not a sufficient requirement to define correctness since a sentence could be undefined in the domain but still being consistent with it. On the other hand, if a sentence is entailed by the domain, then there is a subset of the domain from which we can derive the semantic content of the sentence. For example, given the domain ontology $\mathcal{O}$ and the axioms $a_1 : Producer \equiv Autotroph$ and $a_2 : Producer \sqsubseteq \exists produce.Food$, with $a_1, a_2 \in \mathcal{O}$, the sentence "Autotrophs produce their food" (i.e., $Autotroph \sqsubseteq \exists produce.Food$) is correct because $\mathcal{O} \models (Autotroph \sqsubseteq \exists produce.Food)$.

A sentence is considered to be *incorrect* if it is inconsistent with respect to the domain ontology. In other words, a sentence is incorrect if it violates a domain constraint. In order to determine the consistency of a sentence, we must add it to the domain ontology before doing consistency checking. For example, given the domain ontology $\mathcal{O}$ and the axioms $a_1 : Producer \sqsubseteq \neg Carnivores$ and $a_2 : Producer \sqsubseteq \exists produce.Food$, with $a_1, a_2 \in \mathcal{O}$, the sentence "Carnivores produce their food" (i.e., $Carnivores \sqsubseteq \exists produce.Food$) is incorrect because $\mathcal{O} \cup Carnivores \sqsubseteq \exists produce.Food \models \bot$.

Finally, if a sentence is consistent with the domain ontology, but it cannot be entailed, then it is considered an *incomplete* sentence. By incomplete, we mean that it is not possible to determine the sentence is correct or not. Under the open world assumption, we do not know the truth value of entities and relationships that are not defined in the ontology. Let us consider the case where the axioms $a_1 : Tree \sqsubseteq Producer$ and $a_2 : Producer \sqsubseteq \exists produce.Food$, with $a_1 \notin \mathcal{O}$ and $a_2 \in \mathcal{O}$, the sentence "Trees produce their food" (i.e., $Tree \sqsubseteq \exists produce.Food$) is incomplete because $\mathcal{O} \cup (Tree \sqsubseteq \exists produce.Food) \not\models \bot$ and $\mathcal{O} \not\models (Tree \sqsubseteq \exists produce.Food)$.

However, to be able to determine if a sentence is consistent or entailed by the domain, the domain ontology needs to meet two requirements: consistency and completeness. The first requirement, consistency of the domain ontology, is fundamental for determining the correctness of a sentence. As stated by Haase and Völker [8], an inconsistent ontology is meaningless since anything can be inferred from a set of contradicting axioms. This means that all sentences can be entailed by an inconsistent domain ontology, making all of them correct. In the case of the second requirement, completeness of the domain ontology, it has more practical implication. If the domain ontology does not have all the axioms (e.g., concepts, relationships) required to analyze the text, it is most likely that a sentence will be labeled as incomplete rather than correct or incorrect.

In this work we have selected HermiT as the reasoner because of its higher efficiency (i.e., hypertableau reasoning algorithm) [18].

Explanation of Incorrectness. We expect that in the case of incorrect sentence, our method should provide an explanation of why the sentence contradicts the domain ontology. For that purpose, we have considered the ontology debugging solution proposed by Horridge et al. [10]. As previously mentioned, Horridge et al. *explanation* approach integrates Reiter's Hitting Set Tree (HST) [19] to identify the minimal inconsistent sub-ontology, i.e., subset of axioms from the ontology that cause the inconsistency.

Horridge et al.'s approach first determines a set of inconsistent sub ontologies (e.g., $\mathcal{O}_1 ... \mathcal{O}_n \subseteq \mathcal{O}$ with $\mathcal{O}_1 \models \bot \wedge ... \wedge \mathcal{O}_n \models \bot$). These sub ontologies are obtained by dividing the ontology and checking their consistency. The intuition behind this step is that the cause of inconsistency of the ontology will be located in a section of the ontology (i.e., sub ontology), and not across the whole ontology. From the set of sub ontologies, the approach constructs a Hitting Set Tree (HST) [19]. The Hitting Set Tree algorithm provides a mechanism to identify a minimal set of elements that can provide *minimal coverage* of over a set of clauses. In the case of inconsistent ontology, HST identifies the minimal set of axioms that participate in all the inconsistent sub ontologies. Reiter has offered a set of optimization and pruning mechanisms to reduce the computational cost of HST [19]; yet, since HST requires multiple consistency checks, it is a very costly method.

Horridge et al.'s approach has been incorporated into popular DL reasoners, such as HermiT [18].

4 Experiments

4.1 Data Set

In this work we will use a set of summaries collected on an earlier study by Sohlberg et al. [23] that looked at the use of electronic strategies (eStrategies) for reading comprehension for college students. As part of the study, students were asked to provide oral summaries of each article they had read, where each

article is roughly 3 pages in length (4 articles were used). The oral summaries were manually transcribed into text form. From the Sohlberg et al.'s collection, 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 Example 1.

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.
...

Example 1: Part of a summary from the Ecosystems set.

Because the summaries are originally *oral summaries*, they slightly differ from written ones (as it can be seen in Example 1). The transcribed summaries contain repeated statements, and in some cases there are contradictions when considering the summary as a whole. However, because we focus on resolving the correctness of the text content one sentence at a time, these cohesion issues do not affect our analysis.

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.

4.2 Ontology

In this work, we constructed an ontology for the domain of Ecosystems. We used the introductory article that the student used for their summaries. 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 intend that the ontology will better cover concepts and relationships from the students summaries, which are also solely based on the article.

Table 1. Statistical information about the ontology

Element type	Number of element
Concepts	55
Relationships	30
Axioms	224

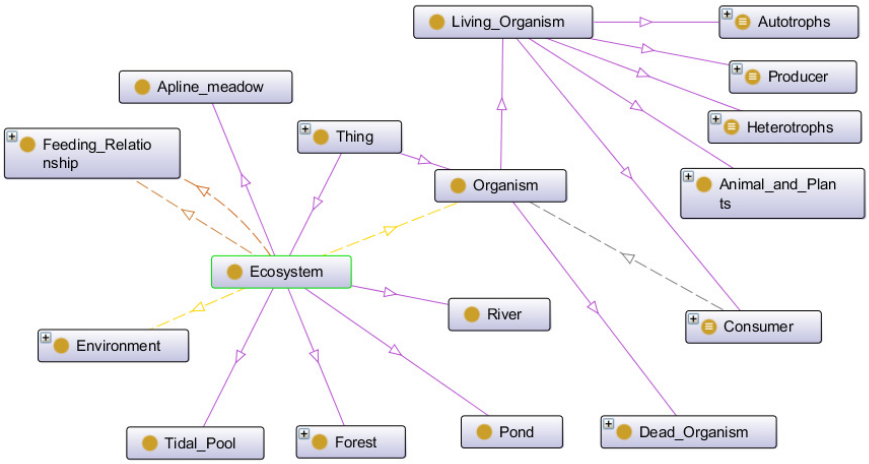

Fig. 1. Graphical representation of a section of the Ecosystems ontology

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. This is originated by the nature of the Ecosystems article. Because the article is an introduction to the domain, a broad set of concepts and relationships of the topic are presented rather than details, such as specific examples. In Figure 1 we presents a graphical representation of a part of the Ecosystems ontology.

It must be mentioned that, although the ontology used in our present approach is similar to the one used in our previous work [5], there is a significant difference in the number of axioms of each ontology. In order to determine incorrectness based on logic contradiction, the ontology for the present work incorporates a large set of constraints, such as disjointness between classes, and strictly defines domain and range for each property.

4.3 Comparison Method

To obtain a better understanding of how well our proposed method performs, we have considered two comparison methods. The first method is our previous work [5], which is, to the best of our knowledge, the only method for error detection in text based on ontologies. As previously mentioned, our previous approach manually defines domain inconsistent axioms by violating ontological constraints. These inconsistent axioms are encoded into extraction patterns that can detect incorrect sentences before the extraction process begins (i.e., precomputed approach).

For comparison, we have used the same set of rules manually defined before. We created the extraction rules by using the domain ontology and considering the content documents. Because it is possible to generate a large set of inconsistent axioms from the domain ontology, we use the content of the four documents to limit the number of extraction rules that need to be generated. This led to 31 extraction rules to identify correct sentences, 16 extraction rules to identify incorrect sentences, and five extraction rules to identify incomplete sentences.

The second comparison method is a variation to our proposed method that replace the IE process with *manual extraction*. This variation can provide us with insight of 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. The use of *manual extractions* can lead to an overall performance depending on directly the performance of the mapping and reasoning steps of our approach. We have constructed a data set formed by binary relationships manually extracted from the 18 summaries. These manually extracted relationships are then analyzed by our approach to determine their correctness.

For the mapping step, we use the same gazetteers for both proposed approach (i.e., automatic extraction) and the *manual extraction* method. The gazetteers were constructed by observing extracted relationships from 40 sentences taken from four of the 18 summaries.

5 Results and Discussion

We use the traditional IE metrics (precision, recall, F1 measure) to measure the quality of the performance of our proposed new method and compare it with our previous method using precomputed errors. Precision measures the correctness of the labeling process, i.e., what percentage of the sentences that are labeled correct are actually correct. Precision 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* ($tp + fp$).

$$P = \frac{tp}{tp + fp}$$

Recall measures the completeness of the labeling process, i.e., what percentage of all correct sentences are labeled as such. Recall 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 negative* ($tp + fn$).

$$R = \frac{tp}{tp + fn}$$

Finally, F1 is the simplest and most commonly used form of F-measure in OBIE, and it is the harmonic mean between precision and recall.

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

From Table 2, we can say that in the case of automatic extraction approach it is possible to determine with high precision the correctness of a sentence with respect to the domain by logic reasoning. However, there is a significant amount of sentences that, although contained in the domain, are considered to be unrelated to it. There are a significant amount of cases where 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 correct and incorrect sentences were captured, the sentences that were labeled as correct are all semantically correct sentences. The same goes with the incorrect sentences.

The perfect precision (i.e., 100%) obtained by both automatic and manual extraction approaches in the case of 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 was labeled as correct when it was actually incorrect, it would mean that reasoning process used to determine the label 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 ontology, the reasoner can accurately determine if 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 automatic extraction approach, with the same level of precision. This result indicates that the quality of the extraction process has a significant effect in the detection of correctness, 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 where the mapping procedure failed to connect extraction entities with ontological concepts.

Table 2. Precision, recall, and F1 measure for the proposed method (automatic and manually extraction) and for the *precomputed* approach [5]

	Automatic Extraction			Manual Extraction			Precomputed approach		
Sentence	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Correct	100%	40.9%	58.1%	100%	80.23%	89.0%	91.9%	83.3%	87.4%
Incorrect	100%	41.3%	58.4%	100%	88.63%	93.97%	97.4%	88.6%	92.8%
Incomplete	89.5% %	100%	94.4%	74.71%	100%	85.52%	66.7%	80.0%	72.7%

When compared with our previous approach, precomputed 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 pre-defined 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 Conclusions and Future Work

In this work, we propose a new approach to detect errors in text. By applying logic reasoning over a domain ontology, we can determine the correctness (and incorrectness) of text content. Our method applies Information Extraction to transform text into a set of logic clauses. These logic clauses are incorporated into the domain ontology to determine if it contradicts the ontology or not. Our experiments show that it is possible to determine the correctness of sentences with higher precision but lower recall compared with our previous work. Our experiments also show that the performance is dependent on the quality of the extractions of the underlying IE process.

As future goals, we have identified four topics in our work that require improvement. The first topic is the improvement in the identification of entities in the extraction process. In many cases, the entities extracted by the IE process are phrases which cannot be clearly mapped to the concepts of the domain ontology because the extraction includes other elements as part of the entity (e.g., a property). This aspect of IE can have a significant impact in our method because a relationship might become unrelated to the domain if its elements cannot be recognized as part of it. A 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. The third topic refers to determining the reason why a sentence is incorrect. Although current ontology debugging methods can provide tentative solutions to this problem, they have both different focus and different parameters to find the origin of inconsistency. For example, because in ontology debugging the origin of the inconsistency is not known, a search mechanism must be defined as part of the debugging process. In the case error detection in text the origin of the inconsistency is the analyzed text and the ontological axioms that are related to the analyzed text, so there is no need for a search mechanism.

Finally, the fourth topic refers to the simultaneous analysis of multiple sentences. Because analyzing a set of sentences at once has a significant computational cost, we are considering an incremental approach. Iteratively, we add statements into the ontology and perform consistency checking. If there is an inconsistency, we try to identify the origin. In this approach, a key element is the order of the statements that are being added into the ontology. For example, from a text we can produce the set of statements $S = s_1, ..., s_i, ..., s_j, ..., s_n$ (with i much smaller than j). 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 later after s_i. If we sort the statements with a selection function such as the one in [12], the analysis with both statements can be performed earlier. Although 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.

Acknowledgments. This research is partially supported by the National Science Foundation grant IIS-1118050 and grant IIS-1013054. 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.

References

1. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., Ant Peter, D.L.M., Patel-Schneider, F., Stein, L.A.: OWL Web Ontology Language, `http://www.w3.org/TR/owl-ref/`
2. Blessing, A., Schütze, H.: Crosslingual distant supervision for extracting relations of different complexity. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM 2012, pp. 1123–1132 (2012)
3. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 1535–1545 (2011)
4. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: Classification and survey. The Knowledge Engineering Review 23(02), 117–152 (2008)
5. Gutierrez, F., Dou, D., Fickas, S., Griffiths, G.: 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 2012, pp. 1722–1726 (2012)
6. Gutierrez, F., Dou, D., Fickas, S., Martini, A., Zong, H.: Hybrid Ontology-based Information Extraction for Automated Text Grading. In: Proceedings of the 12th IEEE International Conference on Machine Learning and Applications, ICMLA 2013, pp. 359–364 (2013)
7. Haase, P., Stojanovic, L.: Consistent evolution of OWL ontologies. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 182–197. Springer, Heidelberg (2005)
8. Haase, P., Völker, J.: Ontology learning and reasoning — dealing with uncertainty and inconsistency. In: da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2005 - 2007. LNCS (LNAI), vol. 5327, pp. 366–384. Springer, Heidelberg (2008)
9. Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proceedings of the Fourteenth Conference on Computational Linguistics, COLING 1992, pp. 539–545 (1992)
10. Horridge, M., Parsia, B., Sattler, U.: Explaining inconsistencies in OWL ontologies. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS, vol. 5785, pp. 124–137. Springer, Heidelberg (2009)
11. Horrocks, I., Patel-Schneider, P.F.: Reducing OWL entailment to description logic satisfiability. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) ISWC 2003. LNCS, vol. 2870, pp. 17–29. Springer, Heidelberg (2003)
12. Huang, Z., van Harmelen, F., Teije, A.t.: Reasoning with inconsistent ontologies. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI 2005, pp. 454–459 (2005)

13. Landauer, T.K., Laham, D., Foltz, P.W.: Learning human-like knowledge by singular value decomposition: a progress report. In: Proceedings of the 1997 Conference on Advances in Neural Information Processing Systems 10, NIPS 1997, pp. 45–51 (1998)
14. Lin, C.-Y.: Rouge: a package for automatic evaluation of summaries. In: Workshop on Text Summarization Branches Out, pp. 25–26 (2004)
15. Mausam, Schmitz, M., Soderland, S., Bart, R., Etzioni, O.: Open language learning for information extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, pp. 523–534 (2012)
16. Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM 38, 39–41 (1995)
17. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, ACL 2009, pp. 1003–1011 (2009)
18. Motik, B., Shearer, R., Horrocks, I.: Hypertableau Reasoning for Description Logics. Journal of Artificial Intelligence Research 36, 165–228 (2009)
19. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32(1), 57–95 (1987)
20. Ritter, A., Downey, D., Soderland, S., Etzioni, O.: 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 2008, pp. 11–20. Association for Computational Linguistics, Stroudsburg (2008)
21. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI 2003, pp. 355–362 (2003)
22. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. Journal of Automated Reasoning 39(3), 317–349 (2007)
23. Sohlberg, M., Griffiths, G., Fickas, S.: The effect of electronically delivered strategies on reading after mild-moderate acquired brain injury. American Journal of Speech-Language Pathology (November 2011) (in review)
24. Wimalasuriya, D.C., Dou, D.: Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science 36, 306–323 (2010)
25. Wu, F., Weld, D.S.: Autonomously semantifying wikipedia. In: Proceedings of the 16th ACM Conference on Information and Knowledge Management, CIKM 2007, pp. 41–50 (2007)