ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-491

Logical Ontology Validation Using an Automatic Theorem Prover

Tim vor der Brück[1] and **Holger Stenzhorn**[2]

Abstract. Ontologies are utilized for a wide range of tasks, like information retrieval/extraction or text generation, and in a multitude of domains, such as biology, medicine or business and commerce. To be actually usable in such real-world scenarios, ontologies usually have to encompass a large number of factual statements. However, with increasing size, it becomes very difficult to ensure their complete correctness. This is particularly true in the case when an ontology is not hand-crafted but constructed (semi)automatically through text mining, for example. As a consequence, when inference mechanisms are applied on these ontologies, even minimal inconsistencies oftentimes lead to serious errors and are hard to trace back and find. This paper addresses this issue and describes a method to validate ontologies using an automatic theorem prover and MultiNet axioms. This logic-based approach allows to detect many inconsistencies, which are difficult or even impossible to identify through statistical methods or by manual investigation in reasonable time. To make this approach accessible for ontology developers, a graphical user interface is provided that highlights erroneous axioms directly in the ontology for quicker fixing.

1 Introduction

The application of ontologies is a vital part of a multitude of different tasks, like for information retrieval/extraction or text generation systems, within a multitude of domains, such as biology, medicine or business and commerce.

In order to be truly useful in real-world practice, such systems normally require large ontologies. But with increasing size it becomes quite difficult to ensure their complete correctness, which is especially true, if the ontology has been created (semi)automatically by e.g., text mining. Furthermore, even minimal errors can lead to fatal consequences if logical inferences are applied, e.g., if the knowledge base contains a contradiction then everything can be deduced from it. In addition, one single incorrect and very general factual statement can lead to a vast amount of other incorrect statements. This combined with the fact that an exhaustive manual error inspection of such large ontologies is not possible, automatic validation methods are necessary.

In the following, we present an approach to automatically check ontologies implemented using the semantic network formalism MultiNet [12] by employing an automatic theorem prover in combination with general purpose axioms, which are applicable in any arbitrary domain (cf. Section 3 for details on MultiNet and Section 4 for the applied ontology).

After the step of identifying potential error candidates in the ontology, a disambiguation is applied to identify for each inconsistency problem one or several erroneous relations.

In our understanding, an ontology contains the relations between concepts, i.e., the actual readings and meanings of words but not the relations between (the surface forms of) words. Since this distinction between words and concepts is of utmost importance, we use the following conventions: In the case that a word reading is intended we add the suffix *.x.y* to the associated surface word form, e.g., *house.1.1* refers to the reading *1.1* of the word *house*. For formal (named) entities, they are followed by *.0*. For non-lexicalized concepts, e.g., *Pete's house*, the characters <> surround the associated words to indicate that the concept and not the surface form is referred to, e.g., <Pete's house>[3]. For better readability we omit all suffixes and brackets in the running text.

2 Related Work

To our knowledge, no prior work exists on the logical validation of ontologies implemented in the MultiNet formalism. However, there are various methods to validate knowledge bases in other formalisms, such as OWL [8].

Ontology validation can either deal with estimating the quality of an ontology as a whole (so-called metrics) [19] or with detecting inconsistencies in it, which is the aim of our approach. For the latter, basically two approach "families" exist that are based on either statistical or logical methods:

- Statistical approaches often employ corpus statistics and natural language processing methods. Usually, those methods assign each ontology entry a confidence score expressing the likelihood of this entry being correct or not.
- Logical approaches apply logical rules to detect inconsistencies. Their output is usually "crisp" since either an entry is inconsistent with some other entry or not.[4]

Cimiano et al. [4] introduce an example of a statistical method with focus on hyponym extraction and validation. One of their described validation methods is based on the fact that a hyponym can appear in the same textual context as the hypernym. This method determines a value ranging from zero to one specifying the hyponymy likeliness. Pantel

[1] FernUniversität in Hagen, Hagen, Germany, tim.vorderbrueck@fernuni-hagen.de

[2] Department of Paediatric Oncology and Haematology, Saarland University Hospital, Homburg, holger.stenzhorn@uks.eu

[3] Non-lexicalized entries are not stored in an ontology.

[4] However, some approaches exist combining logic and statistics that assign each inference a probability value

and Pennachiotti [16] focus on arbitrary semantic relations extracted automatically by text mining and the application of a given pattern set. They derive the confidence score from the pattern precision concerning the extracted relations by calculating the pointwise mutual information between patterns and relations.

Arpinar et al. [1] devise a logical validation method which locates inconsistencies in ontologies by applying consistency rules that are defined by the user in RuleML [3]. They also present several domain-specific example rules but do not give any generally applicable and domain-independent rules. Corcho et al. [5] concentrate on detecting cycles, partition errors (e.g., some concept subordinated to *woman* cannot be subordinated to *man* too) and redundancies in taxonomies. To do so, they do not apply any automated theorem prover and thus cannot make use of arbitrary logical consistency axioms which in turn renders this approach quite limited. Schlobach and Cornet [18] investigate how a theorem prover can be adapted to pinpoint logical contradictions. Ji et al. [13] consider networks of ontologies instead of a single ontology. Finally, Kalyanpur et al. [17, 14] describe a graphical user interface for quick identification of inconsistencies.

In contrast to those approaches, our main point is to describe several domain-independent, generally applicable axioms and to show how a logical and a statistical validation approach can be combined.

Furthermore, instead of OWL we use an ontology implemented in the MultiNet knowledge representation formalism. In contrast to OWL which is based on a restricted subset of first order predicate logic, the expressiveness of the MultiNet formalism goes even beyond first order predicate logic, e.g., by containing support for fuzzy quantifiers.

3 MultiNet

The ontology validated for this work is implemented using MultiNet (*Multi*layered Extended Semantic *Net*works) [12], a semantic network formalism that has already been applied for a wide variety of domains and tasks, such as question answering, readability analysis, geographic information retrieval and literature search [7]. It provides the necessary basic formalism and contains more than 140 relations and functions to define ontologies and to describe the complete semantics of natural language, i.e., the meaning of arbitrary sentences, paragraphs or whole texts - but it does not contain itself any concept definitions.

When a MultiNet concept[5] is defined then it is associated with some semantic information stored either in the semantic lexicon HaGenLex (*Ha*gen *Ge*rma*n* *Lex*icon) [11] or dynamically derived by the deep linguistic parser WOCADI (*Wo*rd *C*lass *Di*sambiguation) [10]. The semantic information consists of an ontological sort (of which there exist more than forty different ones, e.g., *d(iscrete) or ab(stract) object*), semantic features, e.g., human: + for concepts denoting human beings, and a set of layer features, e.g., type of extensionality (*etype*) or cardinality [12].

The ontological sorts and semantic features can also become quite useful in ontology validation [20]. However, in some cases a validation employing ontological sorts or semantic features is not possible since either lexicon entries are missing or sorts and features are not specific enough. In such a case, along with several statistical methods, a logical validation based on a theorem prover is applied here.

In work, we employ the layer feature *type of extensionality* classifying nodes on the pre-extensional knowledge representation level (see [12] or [15] for a distinction of intensional and (pre)extensional interpretation).

The type of extensionality can take the following values:

0: Representative of an elementary extensional, which is itself not a set, e.g., *house.1.1*, <Max> (person named Max)
1: Set of elements of type 0, e.g., <several children>, <three cars>, *team.1.1*, *brigade.1.1*
2: Set of elements of type 1, e.g., <three crews>, <many organizations>, <umbrella organization>
3: Set of elements of type 2
4: ...

MultiNet provides the ELMT(element) relation (defined in Section 4) to specify a member-collection relationship, i.e., $\text{ELMT}(a,b) \Rightarrow \mathit{etype}(a) + 1 = \mathit{etype}(b)$ where $\mathit{etype} : \mathit{Concepts} \rightarrow \mathbb{N}_0$ and $\mathit{etype}(c)$ denotes the type of extensionality of concept c.

4 Ontologies

The two most important relations for our task are meronymy and hyponymy which are further differentiated into several subrelations. Meronymy is a part-whole relation where the concept denoting the part is called the meronym and the containing concept the holonym. Winston [21] states the subrelations (with the corresponding MultiNet relation in brackets):

- Component-integral: A relation between an object and one of its components. Important for this relation is the fact that object and component can be perceived separately from each other, e.g., *A car wheel is part of a car.* (PARS)
- Member-collection: This relation represents the membership in a set, e.g., *A soccer player is a member of a soccer team.* (ELMT)
- Portion-mass: Relations which refer to mass units and their parts, e.g., *A meter is part of a kilometer, a slice of the cake is part of a cake.* (PARS, for temporal units TEMP)
- Stuff-object: This relation represents the chemical composition of an object, e.g., *Alcohol is part of wine. Steel is part of a bike.* (PARS or ORIGM^{-1} in the case the holonym denotes a physical object)
- Feature-activity: Activities can usually be divided into several subtasks, e.g., the following subtasks belong to the activity *going out for dinner*: *visiting a restaurant*, *ordering*, *eating* and *payment.* (HSIT)
- Place-area: This relation holds between two objects if one of these objects is geographically part of the other object, e.g., *Germany is part of Europe.* (PARS)

Additionally Helbig [12] defines a further meronymy subrelation for subsets called SUBM, e.g., *brigade* is a subset of *division.* Note that the relationship between brigade and division is not of type member-collection since the elements of a division and a brigade are in both cases soldiers.

MultiNet can also be used to describe instance relations through an attribute value mechanism. For example, the fol-

[5] Note that MultiNet treats instances as concepts as well.

lowing denotes the fact that Germany is a part of Europe:

$$\begin{aligned}&\textsc{attr}(x,y) \wedge \textsc{sub}(y, \mathit{name.1.1}) \wedge \textsc{val}(y, \mathit{germany.0})\wedge\\&\textsc{sub}(x, \mathit{country.1.1}) \wedge \textsc{pars}(x,z)\wedge\\&\textsc{attr}(z,u) \wedge \textsc{sub}(u, \mathit{name.1.1}) \wedge \textsc{val}(u, \mathit{europe}.0)\wedge\\&\textsc{sub}(z, \mathit{continent.1.1})\end{aligned} \tag{1}$$

According to Lyons, an expression x is a hyponym of another expression y if and only if x entails y, e.g., if a concept denotes a dog then it denotes also an animal [15]. MultiNet defines the following hyponymy subrelations:

- SUBS: Relation of conceptual subordination for situations, e.g., the situation *party* is subordinated to *event*
- SUBR: Relation of conceptual subordination for relations, e.g., *equality* is subordinated to *relation*
- SUB: Relation of conceptual subordination not covered by the first two cases, e.g., a *church* is subordinated to *building*

Note that SUB, SUBR, and SUBS are also used to specify *instance of* relations (see Equation 1).

Other important relations defined for ontologies are:

- ANTO: Antonymy relation, e.g., *increase* is an antonym of *decrease*
- SYNO: Synonymy relation, e.g., *kid* is a synonym of *child*

There is a strict differentiation in MultiNet between the cases where a meronymy relation holds directly or where an additional SUB relation needs to be included. For example:

$$\textsc{pars}(\mathit{car\_wheel.1.1}, \mathit{car.1.1}) \text{ but} \tag{2}$$

$$\textsc{sub}(x, \mathit{wheel.1.1}) \wedge \textsc{pars}(x, \mathit{car.1.1}) \tag{3}$$

The second example states that something exists, which is derived from *wheel.1.1* (i.e., *car_wheel.1.1*) and which is part of a car.

5 Search for Contradictions

The automatic theorem prover E-KRHyper[6] [2] is applied to find incorrect entries of the knowledge base by deriving contradictions. E-KRHyper supports full first predicate logic and uses a tableau algorithm for proving. The validation process is done in several steps:

1. A subset *TDB* of the knowledge base *KB* which is to be validated is stored in the theorem prover's fact database.
2. Additionally, a validated knowledge base *VKB* can be specified that contains knowledge which is known to be true.
3. A synonymy normalization is done such that each concept is replaced by the lexicographic smallest element of its synonymy set, e.g., $\mathit{normalize}(\mathit{car.1.1}) = \mathit{auto.1.1}$ if $\mathit{synset}(\mathit{car.1.1}) = \{\mathit{car.1.1}, \mathit{auto.1.1}\}$.
4. The theorem prover is applied on the fact database.
5. All instantiated relations (facts) that are used by E-KRHyper to derive a contradiction and which are not found in the validated knowledge base are considered potentially erroneous. Those relations are marked and removed from the fact database employed by the theorem prover. Afterwards, the entire process is repeated again until no further contradiction can be found (go to Step 4).

The entire process is shown as pseudo-code in Figure 1. For deriving the contradictions a set of MultiNet axioms is used.

```
Input: knowledge base KB and
  validated knowledge base VKB
 Select TDB with TDB$\subset$KB
 Loop:
  facts:=derive_contradiction(TDB$\cup$VKB)
  facts=$\emptyset$ ? $\Rightarrow$ exit loop
  for all f $\in$ facts
   f$\notin$ VKB ? $\Rightarrow$ mark(f,KB)
  TDB:=TDB $\setminus$ facts
 End Loop
```

Figure 1. Pseudocode for recognizing inconsistent relations in the knowledge base

6 Case Study: Important Inconsistencies

To keep the set of axioms small we investigate which of them are needed to derive several types of inconsistencies the theorem prover should be able to identify. A typical inconsistency is the asymmetry of the meronymy and hyponymy relations. Both types of relations can in most cases be expressed by the MultiNet relations SUB and PARS. For instance, if

$$\textsc{pars}(\mathit{car\_wheel.1.1}, \mathit{car.1.1}) \tag{4}$$

then the relation

$$\textsc{pars}(\mathit{car.1.1}, \mathit{car\_wheel.1.1}) \tag{5}$$

cannot hold. Then we investigate what happens if Equation 4 and Equation 5 are modified in such a way that additional SUB relations are involved, like for example Equation 6:

$$\begin{aligned}&\textsc{sub}(x, \mathit{wheel.1.1}) \wedge \textsc{pars}(x, \mathit{car.1.1})\wedge\\&\textsc{sub}(y, \mathit{car.1.1}) \wedge \textsc{pars}(y, \mathit{wheel.1.1})\end{aligned} \tag{6}$$

It can be shown, however, that this example leads to a contradiction as well, which is proven in Theorem 1 by applying several MultiNet axioms.

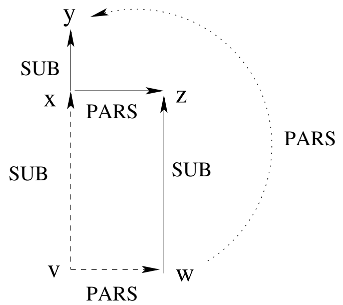


Figure 2. Proof by contradiction: the dashed lines indicate inferred relations, the dotted one the relation to be contradicted.

Theorem 1 *Let us assume:*

$$\begin{aligned}\mathit{KB}(=\text{knowledgebase}) &\models \textsc{sub}(x,y) &(7)\\ \mathit{KB} &\models \textsc{pars}(x,z) &(8)\\ \mathit{KB} &\models \textsc{sub}(w,z) &(9)\end{aligned}$$

Claim: $\mathit{KB} \models \neg\textsc{pars}(w,y)$

[6] E-KRHyper is available as open-source at http://www.uni-koblenz.de/~bpelzer/ekrhyper

Proof by contradiction, assuming $KB \models \textsc{pars}(w, y)$
Note that x does not have to be lexicalized. It often denotes a non-lexicalized concept, which is a subtype (hyponym) of y and a part of z. The proof is illustrated in Figure 2.

$$\begin{aligned}
KB &\models \textsc{pars}(x,z) \wedge \textsc{sub}(w,z) \wedge \textsc{sub}(x,y)\\
\Rightarrow \exists v : KB &\models \textsc{sub}(v,x) \wedge \textsc{pars}(v,w)\\
&\quad (\text{Axiom : Inheritance of Part} - \text{Whole}\\
&\quad \text{Relationships[12]/Modus Ponens})\\
\Rightarrow KB &\models \textsc{sub}(v,y) \text{ and}\\
&\quad (\text{Transitivity of } \textsc{sub}\text{/Modus Ponens})\\
KB &\models \textsc{pars}(v,y)\\
&\quad (\text{Transitivity of } \textsc{pars}\text{/Modus Ponens})
\end{aligned}$$

But $KB \models \textsc{pars}(v,y) \wedge \textsc{sub}(v,y)$ is not possible. Therefore, the assumption must hold. *q.e.d.*

Analogously, a contradiction can be proven if only one of the SUB relations in Equation 6 show up, e.g., $\textsc{sub}(x, \mathit{wheel.1.1}) \wedge \textsc{pars}(x, \mathit{car.1.1}) \wedge \textsc{pars}(\mathit{car.1.1}, \mathit{wheel.1.1})$. Theorem 1 is important for two reasons: First, it states that all relations where this theorem is applied contain an inconsistency. Second, the axioms used for the proof can be very useful because, instead of the theorem, the theorem prover can employ those axioms. This procedure has the advantage of a higher generality which means that additional inconsistencies can be found possibly not detectable by employing only the theorem.

Next we consider the case that instead of the PARS relation the ELMT relation is used, e.g.,

$$\begin{aligned}
&\textsc{sub}(\mathit{soldier.1.1}, \mathit{man.1.1}) \wedge\\
&\textsc{elmt}(\mathit{soldier.1.1}, \mathit{division.1.1}) \wedge\\
&\textsc{sub}(\mathit{airforce\_division.1.1}, \mathit{division.1.1}) \wedge\\
&\textsc{elmt}(\mathit{airforce\_division.1.1}, \mathit{man.1.1})
\end{aligned} \tag{10}$$

Theorem 1 can no longer be applied since the axiom *Inheritance of Part-Whole* is only defined for the relations SUB and PARS. Also, the ELMT relation is not transitive, which is used in the proof. Thus, an additional theorem has to be stated which handles the relations SUB and ELMT:

Theorem 2 *Let us assume:*
$KB \models \textsc{sub}(x,y) \wedge \textsc{elmt}(x,z) \wedge \textsc{sub}(w,z)$
Claim: $KB \models \neg\textsc{elmt}(w,y)$

Proof by contradiction, assuming $KB \models \textsc{elmt}(w,y)$.
Let $\mathit{etype}(x) = n$.

$$\begin{aligned}
&\Rightarrow \mathit{etype}(z) = n+1 (\text{Definition of } \textsc{elmt})\\
&\Rightarrow \mathit{etype}(w) = n+1\\
&(\text{Concepts connected by } \textsc{sub} \text{ have}\\
&\text{identical types of extensionality provided}\\
&\text{that the type of extensionality is not}\\
&\text{underspecified by the hypernym})\\
&\Rightarrow \mathit{etype}(y) = n+2\\
&\Rightarrow \mathit{etype}(x) = n+2 = n\\
&\text{which is a contradiction.}
\end{aligned} \tag{11}$$

q.e.d.

Now consider the case the second SUB relation in Equation 6 is directed in the opposite direction. In order to form a meaningful MultiNet expression the example had to be additionally adjusted in such a way that the anonymous concept y was replaced by *machine.1.1*:

$$\begin{aligned}
&\textsc{sub}(x, \mathit{wheel.1.1}) \wedge \textsc{pars}(x, \mathit{car.1.1}) \wedge\\
&\textsc{sub}(\mathit{car.1.1}, \mathit{machine.1.1}) \wedge\\
&\textsc{pars}(\mathit{machine.1.1}, \mathit{wheel.1.1})
\end{aligned} \tag{12}$$

In this case a logical contraction using MultiNet axioms cannot be reached but it is quite unlikely that the meronymy relation changes its direction if hyponyms of the original concepts are compared. Thus we define the following assumption:

Assumption 1 Let us assume:
$KB \models \textsc{sub}(x,y) \wedge \textsc{pars}(x,z) \wedge \textsc{pars}(w,y)$
Claim: $KB \models \neg\textsc{sub}(z,w)$

Note however, that a contradiction could be derived if the relation PARS is replaced by the ELMT relation which can be shown analogously to the proof of Theorem 2.

Table 1. Logical axioms used to derive contradictions

Axiom ID	Matching formula
N_1	$\textsc{sub}(x,y) \rightarrow \neg\textsc{sub}(y,x)$
N_2	$\textsc{pars}(x,y) \rightarrow \neg\textsc{pars}(y,x)$
N_3	$\textsc{sub}(x,y) \wedge \textsc{sub}(y,z) \rightarrow \textsc{sub}(x,z)$
N_4	$\textsc{pars}(x,y) \wedge \textsc{pars}(y,z) \rightarrow \textsc{pars}(x,z)$
N_5	$\textsc{sub}(x,y) \rightarrow \neg\textsc{pars}(x,y)$
N_6	$\textsc{sub}(x,y) \wedge \textsc{pars}(z,y) \rightarrow$ $\exists u : \textsc{sub}(u,z) \wedge \textsc{pars}(u,x)$
N_7	$\textsc{elmt}(x,y) \rightarrow \mathit{etype}{<}?(x,y)$
N_8	$\mathit{etype}{<}?(x,y) \wedge \mathit{etype}{<}?(y,z) \rightarrow \mathit{etype}{<}?(x,z)$
N_9	$\neg\mathit{etype}{<}?(x,x)$
N_{10}	$\mathit{etype}{<}?(x,y) \wedge \textsc{sub}(z,y) \rightarrow \mathit{etype}{<}?(x,z)$
N_{11}	$\textsc{sub}(x,y) \wedge \textsc{pars}(y,z) \wedge$ $\textsc{pars}(x,w) \rightarrow \neg\textsc{sub}(w,z)$
N_{12}	$\textsc{anto}(x,y) \wedge \textsc{sub}(w,x) \rightarrow \neg\textsc{sub}(w,y)$

Table 1 shows a subset of the axioms we employ. Literal N_1 states the relation SUB to be asymmetric. This axiom can be stated in analogous form for the other hyponymy subrelations SUBS and SUBR. The PARS relation is also asymmetric as stated in axiom N_2. Again, corresponding axioms can be defined for the other meronymy subrelations ELMT, HSIT, $\textsc{origm}^{-1}$, SUBM, and TIME. Axioms N_3 and N_4 reflect the transitivity of SUB and PARS. N_5 specifies that the relations SUB and PARS cannot hold simultaneously. N_3, N_4, N_5, and N_6 are required for the proof of Theorem 1 and are, therefore, necessary to derive contradictions for relations which are inconsistent according to this theorem. In axiom N_7 a predicate $\mathit{etype}{<}?(x,y)$ is introduced which is fulfilled if the type of extensionality of concept x falls below the type of extensionality of concept y. In case the relation ELMT holds between two concepts x and y ($\textsc{elmt}(x,y)$) then the type of extensionality of x is one less than the type of extensionality of y, i.e., $\mathit{etype}{<}?(x,y)$. The predicate $\mathit{etype}{<}?$ is transitive (N_8) and irreflexive (N_9). Furthermore, a hyponym has the identical type of extensionality as the hypernym (if the hypernym is not underspecified, N_{10}). The axioms N_7, N_8, N_9, and N_{10} are required for the proof of Theorem 2 and therefore also to derive a contradiction for relations of the knowledge base where this theorem can be applied. N_{11} states the heuristic illustrated in Equation 12 (Assumption 1). N_{12} is a generalization of MultiNet axiom 133 [12, p.475] and should at least hold prototypically.

7 Error Disambiguation

Usually not all relations employed by the automatic theorem prover to derive a single contradiction are actually erroneous. Actually, in a lot of cases only a single relation is incorrect. But the identification of this particular relation is usually not trivial and thus an automated mechanism to already point in the right direction is very helpful. We followed two approaches here:

First, a validated and trusted knowledge base (HaGenLex which is mainly derived from Wiktionary and GermaNet [9] where the mapping to HaGenLex concept identifiers was done manually) is used in addition. All relations defined in this knowledge base are assumed to be correct. This means that the relations determined by the theorem prover contained in this knowledge base can be discharged as error candidates.

Second, additional features are used to estimate the quality of knowledge base entries. These features are combined to a global quality score [20] and include context comparison of hyponym/hypernym (similar to [4]), taxonomy-based validation for meronyms [6] and exploit the fact that in many cases the relation correctness can be reliably estimated by regarding only the assumed hypernym alone (hyponym/meronym/holonym respectively).

In case one of the relations found by the theorem prover is assigned a quality score significantly lower than the other ones which can be determined by an outlier detection approach then it is assumed to be incorrect.

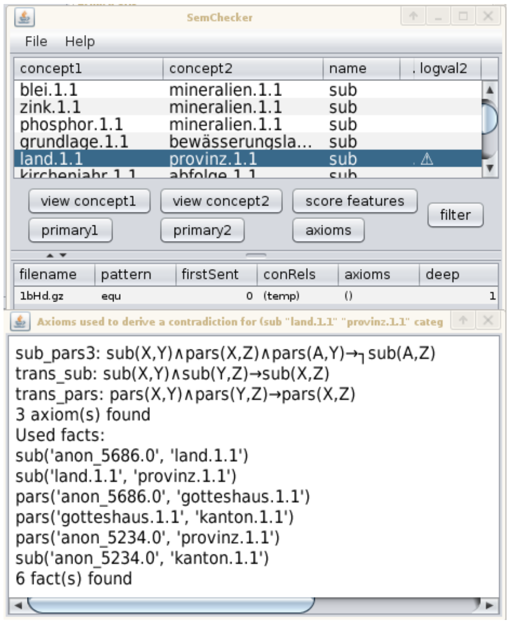
An alternative approach not followed here would be to not remove inconsistent relations one after the other from the knowledge base instead of removing all of them simultaneously. If the contradiction disappears after the removal of some single relation this relation is probably incorrect. However, if the contradiction can still be derived then this relation can be assumed to be correct. Note that such an approach would lead to a serious increase in calculation time and thus strongly decrease performance.

8 Graphical User Interface

We implemented a graphical user interface *SemChecker* to visualize the identified inconsistencies (see Figure 3). It displays all entries of the knowledge base. Furthermore, it states for a selected entry the name of the file from which this entry was extracted, the pattern employed to extract this entry, and also the information whether this pattern is deep (1) or shallow (0). Entries leading to a contradiction are marked by an attention symbol. Additionally, SemChecker displays all axioms applied to derive the contradiction and all facts where those axioms were applied to.

9 Evaluation

We evaluated our approach on an ontology constructed automatically by text mining. For that the German Wikipedia was converted into a semantic network representation following the MultiNet formalism, by employing the deep syntactico-semantic parser WOCADI. We applied a set of patterns given as semantic subnetworks on the Wikipedia sentences in the form of semantic networks and extracted a set of hyponymy and meronymy relations. Additionally a set of shallow patterns were applied on the token information to guarantee a high recall if sentences could not be parsed. Most occurring subrelations of the extracted pairs were PARS (636 711) and SUB (153 459), followed by SUBS (13 236), SUBM (30 375), and ELMT (5 961). All relation pairs (in total 847 727) were assigned a confidence score estimating the likelihood of their correctness [20]. The highest-scored relations were investigated for a possible addition to our knowledge base and were, therefore, checked with the theorem prover.



concept1	concept2	name	logval2
blei.1.1	mineralien.1.1	sub	
zink.1.1	mineralien.1.1	sub	
phosphor.1.1	mineralien.1.1	sub	
grundlage.1.1	bewässerungsla...	sub	
land.1.1	provinz.1.1	sub	⚠
kirchenjahr.1.1	abfolge.1.1	sub	

filename	pattern	firstSent	conRels	axioms	deep
1bHd.gz	equ	0	(temp)	()	1

```
sub_pars3: sub(X,Y)∧pars(X,Z)∧pars(A,Y)→┐sub(A,Z)
trans_sub: sub(X,Y)∧sub(Y,Z)→sub(X,Z)
trans_pars: pars(X,Y)∧pars(Y,Z)→pars(X,Z)
3 axiom(s) found
Used facts:
sub('anon_5686.0', 'land.1.1')
sub('land.1.1', 'provinz.1.1')
pars('anon_5686.0', 'gotteshaus.1.1')
pars('gotteshaus.1.1', 'kanton.1.1')
pars('anon_5234.0', 'provinz.1.1')
sub('anon_5234.0', 'kanton.1.1')
6 fact(s) found
```

Figure 3. Graphical User Interface for the Logical Validation highlighting the entry SUB(*land.1.1*, *provinz.1.1*) (SUB(*country.1.1*, *province.1.1*)) as potentially erroneous

Currently, we perform a theorem prover check on the 30 000 highest scored meronymy and hyponymy relation candidates (TDB, see Section 5). The data set consists of all 6 000 ELMT relations, best 12 000 hyponyms, and 12 000 meronyms. The theorem prover timeout per proof was set to 9000 seconds.

We employ 16 axioms; a selection of them is given in Table 1. In total 164 inconsistent relations were identified. Only 9 of them were identified by asymmetry- and ANTO-axioms (N_1, N_2, and N_{12} in Table 1 and Table 2) which correspond to the usual cycle and partition-checks which are the only inconsistency verifications for many systems, for instance [5]. An Example rejected according to Theorem 1 is the following[7]:

$$\textsc{sub}(x, \textit{town.1.1}), \textsc{pars}(x, \textit{church.1.1})$$
$$\textsc{sub}(y, \textit{church.1.1}), \textsc{pars}(y, \textit{town.1.1})$$

[7] For better understandability, examples where translated from German to English

An example rejected according to assumption 1 is:

$$\begin{aligned} \textsc{sub}(\textit{residence.1.1}, \textit{building.1.1}),\\ \textsc{pars}(\textit{building.1.1}, \textit{fort.1.1}),\\ \textsc{sub}(x, \textit{fort.1.1}), \textsc{pars}(x, \textit{residence.1.1}) \end{aligned}$$

Table 2. A collection of applied axioms. For performance reasons, Theorem 1 was added as additional (redundant) axiom.

Axiom	Number of relations	Name
N_1	6	Asymmetry of SUB
N_2	3	Asymmetry of PARS
N_3	98	Transitivity of SUB
N_4	20	Transitivity of PARS
N_5	59	Incompatibility of SUB and PARS
N_7	5	ELMT and *etype*
N_9	5	Irreflexivity of *etype*
N_{10}	5	SUB and *etype*
N_{11}	34	Assumption 1
-	57	Theorem 1

The number of contradicted relations derived by a certain axiom is given in Table 2. Furthermore, recall and precision of the error disambiguation (see Section 7) were evaluated. Precision specifies the relative frequency with which a predicted error candidate is actually erroneous. Recall denotes the relative frequency of erroneous entries determined by the automatic theorem prover which were actually identified by the disambiguation as erroneous. All relations leading to contradictions were annotated for correctness by members of our department. The following evaluation values were determined: Precision: 0.72, Recall 0.91, F-Measure: 0.80.

10 Conclusion and Outlook

In the above sections, we have presented a method for detecting knowledge base inconsistencies employing an automatic theorem prover. This method was applied on a knowledge base automatically extracted by text mining. The evaluation showed that our method detects a reasonable number of incorrect relations which the usual cycle/partition check and purely statistical methods failed to find (incorrect in spite of a high score). For future work, we are going to test our approach on a larger axiom set as well as on larger ontologies with focus on such that hold more ANTO and ELMT relations.

By using a huge number of axioms which is not currently possible due to the limited computing power of state of the art computers, we believe that logical methods will allow to detect the majority of incorrect relations and, thus, be essential to guarantee a high quality of large knowledge bases.

ACKNOWLEDGEMENTS

We thank all members of our departments for their support. This work was in part funded by the DFG project *Semantische Duplikatserkennung mithilfe von Textual Entailment* (HE 2847/11-1).

REFERENCES

[1] B. Arpinar et al., ‘Ontology quality by detection of conflicts in metadata’, in *4th International Workshop on Evaluation of Ontologies on the Web*, Edinburgh, UK, (2006).

[2] P. Baumgartner et al., ‘Hyper tableaux with equality’, in *Automated Deduction – CADE-21*, volume 4603 of *LNCS*, 492–507, Springer, Heidelberg, Germany, (2007).

[3] H. Boley et al., ‘Design rationale of RuleML: A markup language for semantic web rules’, in *Proc. of Semantic Web Working Symposium*, pp. 381–401, (2001).

[4] P. Cimiano et al., ‘Learning taxonomic relations from heterogeneous sources of evidence’, in *Ontology Learning from text: Methods, evaluation and applications*, eds., P. Buitelaar et al., 59–73, IOS Press, Amsterdam, The Netherlands, (2005).

[5] Ó. Corcho et al., ‘ODEval: A tool for evaluating RDF(S), DAML+OIL, and OWL concept taxonomies’, in *Proc. of AIAI*, Toulouse, France, (2004).

[6] F. Costello, ‘UCD-FC: Deducing semantic relations using WordNet senses that occur frequently in a database of noun-noun compounds’, in *SemEval 2007: Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 370–373, Prague, Czech Republic, (2007).

[7] I. Glöckner et al., ‘Automatic semantic analysis for NLP applications’, *Zeitschrift für Sprachwissenschaft*, **26**(2), 241–266, (2007).

[8] W3C OWL Working Group. OWL web ontology language reference W3C recommondation, 2009. URL: `http://www.w3.org/TR/owl2-overview/`.

[9] B. Hamp and H. Feldweg, ‘GermaNet - a lexical-semantic net for German’, in *Proc. of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid, Spain, (1997).

[10] S. Hartrumpf, *Hybrid Disambiguation in Natural Language Analysis*, Ph.D. dissertation, FernUniversität in Hagen, Fachbereich Informatik, Hagen, Germany, 2002.

[11] S. Hartrumpf, H. Helbig, and R. Osswald, ‘The semantically based computer lexicon HaGenLex – Structure and technological environment’, *Traitement automatique des langues*, **44**(2), 81–105, (2003).

[12] H. Helbig, *Knowledge Representation and the Semantics of Natural Language*, Springer, Heidelberg, Germany, 2006.

[13] Q. Ji et al., ‘RaDON - repair and diagnosis in ontology networks’, in *ESWC 2009*, LNCS, 863–867, Springer, Heidelberg, Germany, (2009).

[14] A. Kalyanpur et al., ‘Debugging unsatisfiable classes in OWL ontologies’, *Journal of Web Semantics*, **3**(4), 268–293, (2005).

[15] J. Lyons, *Linguistic Semantics - An introduction*, Cambridge University Press, Cambridge, UK, 2002.

[16] P. Pantel and M. Pennachiottti, ‘Espresso: Leveraging generic patterns for automatically harvesting semantic relations’, in *Proc. of COLING/ACL*, Sydney, Australia, (2006).

[17] B. Parsia et al., ‘Debugging OWL ontologies’, in *Proc. of the International conference on World Wide Web*, (2005).

[18] S. Schlobach and R. Cornet, ‘Non-standard reasoning services for the debugging of description logic terminologies’, in *Proc. of IJCAI*, Acapulco, Mexico, (2003).

[19] S. Tartir et al. Ontological evaluation and validation, 2007. online available: `http://bit.ly/95lfnP`.

[20] T. vor der Brück, ‘Hypernymy extraction using a semantic network representation’, *International Journal of Computational Linguistics and Applications (IJCLA)*, **1**(1), (2010).

[21] M. Winston et al., ‘A taxonomy of part-whole relations’, *Cognitive Science*, **11**(4), 417–444, (1987).