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

doi:10.3233/978-1-60750-606-5-991

Diagnosis discrimination for ontology debugging

Kostyantyn Shchekotykhin and **Gerhard Friedrich** [1,2]

Abstract. Debugging is an important prerequisite for the widespread application of ontologies, especially in areas that rely upon everyday users to create and maintain knowledge bases, such as the Semantic Web. Recent approaches use diagnosis methods to identify sources of inconsistency. However, in most debugging cases these methods return many alternative diagnoses, thus placing the burden of fault localization on the user. This paper demonstrates how the target diagnosis can be identified by performing a sequence of observations, that is, by querying an oracle about entailments of the target ontology. We exploit probabilities of typical user errors to formulate information theoretic concepts for query selection. Our evaluation showed that the suggested method reduces the number of required observations compared to myopic strategies.

1 Introduction

The application of semantic systems, including the Semantic Web technology, is largely based on the assumption that the development of ontologies can be accomplished efficiently even by every day users. For such users and also for experienced knowledge-engineers the identification and correction of erroneous ontological definitions can be an extremely hard task. Ontology debugging tools simplify the development of ontologies by localizing a set of axioms that should be modified in order to formulate the intended target ontology.

To debug an ontology a user must specify some requirements such as coherence and/or consistency. Additionally, one can provide test cases [2] which must be fulfilled by the target ontology $\mathcal{O}_t$. A user has to change at least all of the axioms of one diagnosis in order to satisfy all of the requirements and test cases.

However, the diagnosis methods can return many alternative diagnoses for a given set of test cases and requirements. In such cases it is unclear how to identify the target diagnosis. In this paper we present an approach to acquire additional information by generating a sequence of queries that are answered by some oracle such as a user, an information extraction system, etc. Each answer to a query reduces the set of diagnoses until finally the target diagnosis is identified. In order to construct queries we exploit the property that different diagnoses imply unequal sets of axioms. Consequently, we can differentiate between diagnoses by asking the oracle if the target ontology should imply an axiom or not. These axioms can be generated by classification and realization services provided in reasoning systems.

2 Entropy-based query selection

In order to focus on the essentials of our approach we employ the following simplified definition of diagnosis without limiting its generality. A more detailed version can be found in [2]. We allow the user to define a background theory (represented as a set of axioms) which is considered to be correct, a set of logical sentences which must be implied by the target ontology and a set of logical sentences which must *not* be implied by the target ontology. Δ is a set of axioms which are assumed to be faulty.

Definition 1: *Given a diagnosis problem $\left\langle \mathcal{O}, B, T^{\models}, T^{\not\models} \right\rangle$ where $\mathcal{O}$ is an ontology, B a background theory, $T^{\models}$ a set of logical sentences which must be implied by the target ontology $\mathcal{O}_t$, and $T^{\not\models}$ a set of logical sentences which must* not *be implied by $\mathcal{O}_t$. A diagnosis is a partition of $\mathcal{O}$ in two disjoint sets $\mathcal{D}$ and Δ ($\mathcal{D} = \mathcal{O} \setminus \Delta$) s.t. $\mathcal{D}$ can be extended by a logical description EX and $\mathcal{D} \cup B \cup EX \models t^{\models}$ for all $t^{\models} \in T^{\models}$ and $\mathcal{D} \cup B \cup EX \not\models t^{\not\models}$ for all $t^{\not\models} \in T^{\not\models}$.*

A diagnosis $(\mathcal{D}, \Delta)$ is minimal if there is no proper subset of the faulty axioms $\Delta' \subset \Delta$ such that $(\mathcal{D}', \Delta')$ is a diagnosis. The following proposition allows us to characterize diagnoses without the extension EX. The idea is to use the sentences which must be implied to approximate EX.

Corollary 1: *Given a diagnosis problem $\left\langle \mathcal{O}, B, T^{\models}, T^{\not\models} \right\rangle$, a partition of $\mathcal{O}$ in two disjoint sets $\mathcal{D}$ and Δ is a diagnosis iff $\mathcal{D} \cup B \cup \{\bigwedge_{t^{\models} \in T^{\models}} t^{\models}\} \cup \neg t^{\not\models}$ consistent for all $t^{\not\models} \in T^{\not\models}$.*

In the following we assume that a diagnosis always exists under the (reasonable) condition that the background theory together with the axioms in $T^{\models}$ and the negation of axioms in $T^{\not\models}$ are mutually consistent. For the computation of diagnoses the set of conflicts is usually employed.

Definition 2: *Given a diagnosis problem $\left\langle \mathcal{O}, B, T^{\models}, T^{\not\models} \right\rangle$, a conflict CS is a subset of $\mathcal{O}$ s.t. there is a $t^{\not\models} \in T^{\not\models}$ and $CS \cup B \cup \{\bigwedge_{t^{\models} \in T^{\models}} t^{\models}\} \cup \neg t^{\not\models}$ is inconsistent.*

A conflict is the part of the ontology that preserves the inconsistency. Note, incoherence can be reduced to inconsistency by adding background axioms or recognized by built-in reasoning services. A minimal conflict CS has no proper subset which is a conflict. $(\mathcal{D}, \Delta)$ is a (minimal) diagnosis iff Δ is a (minimal) hitting set of all (minimal) conflict sets. In the following we represent a diagnosis by the set of axioms $\mathcal{D}$ assumed to be correct. In order to differentiate between the minimal diagnoses an oracle can be queried for information about the entailments of the target ontology.

Property 1: *Given a diagnosis problem $\left\langle \mathcal{O}, B, T^{\models}, T^{\not\models} \right\rangle$, a set of diagnoses $\mathbf{D}$, and a set of logical sentences X representing the query $\mathcal{O}_t \models X$? : If the oracle gives the answer* yes *then every diagnosis $\mathcal{D}_i \in \mathbf{D}$ is a diagnosis for $T^{\models} \cup X$ iff $\mathcal{D}_i \cup B \cup \{\bigwedge_{t^{\models} \in T^{\models}} t^{\models}\} \cup \{X\} \cup \neg t^{\not\models}$ is consistent for all $t^{\not\models} \in T^{\not\models}$. If the oracle gives the answer* no *then every diagnosis $\mathcal{D}_i \in \mathbf{D}$ is a diagnosis for $T^{\not\models} \cup \{X\}$ iff $\mathcal{D}_i \cup B \cup \{\bigwedge_{t^{\models} \in T^{\models}} t^{\models}\} \cup \neg X$ is consistent.*

Note, a set X corresponds to a logical sentence where all elements of X are connected by $\wedge$. This defines the semantic of $\neg X$.

As possible queries we consider sets of entailed concept definitions provided by a classification service and sets of individual assertions provided by realization. In fact, the intention of classification is that a model for a specific application domain can be verified by exploiting the subsumption hierarchy.

One can use different methods to select the best query in order to

[1] The research project is funded by grants of the Austrian Science Fund (Project V-Know, contract 19996). An extended version of the paper can be found at http://arxiv.org/abs/1004.5339

[2] University Klagenfurt, Austria, email: firstname.lastname@uni-klu.ac.at

minimize the number of questions to the oracle. For these methods it is essential to compute the set of diagnoses that can be rejected depending on the query outcome. For a query X the set of diagnoses $\mathbf{D}$ can be partitioned in sets of diagnoses $\mathbf{D^X}$, $\mathbf{D^{\neg X}}$ and $\mathbf{D}^\emptyset$ where

- for each $\mathcal{D}_i \in \mathbf{D^X}$ it holds that $\mathcal{D}_i \cup B \cup \{\bigwedge_{t^\models \in T^\models} t^\models\} \models X$
- for each $\mathcal{D}_i \in \mathbf{D^{\neg X}}$ it holds that $\mathcal{D}_i \cup B \cup \{\bigwedge_{t^\models \in T^\models} t^\models\} \models \neg X$
- $\mathbf{D}^\emptyset = \mathbf{D} \setminus (\mathbf{D^X} \cup \mathbf{D^{\neg X}})$

Given a diagnosis problem we say that the diagnoses in $\mathbf{D^X}$ predict yes as a result of the query X, diagnoses in $\mathbf{D^{\neg X}}$ predict no, and diagnoses in $\mathbf{D}^\emptyset$ do not make any predictions.

Property 2: *Given a diagnosis problem* $\left\langle \mathcal{O}, B, T^\models, T^{\not\models} \right\rangle$, *a set of diagnoses* $\mathbf{D}$, *and a query* X*:* (a) *If the oracle gives the answer* yes *then the set of rejected diagnoses is* $\mathbf{D^{\neg X}}$ *and the set of remaining diagnoses is* $\mathbf{D^X} \cup \mathbf{D}^\emptyset$. (b) *If the oracle gives the answer* no *then the set of rejected diagnoses is* $\mathbf{D^X}$ *and the set of remaining diagnoses is* $\mathbf{D^{\neg X}} \cup \mathbf{D}^\emptyset$.

To select the best query we make the assumption that knowledge is available about the a-priori failure probabilities in specifying axioms. Such probabilities can be estimated by observing the typical failures of specific users working with an ontology development tool. If the probabilities are not known they can be initialized with some small number. Given the failure probabilities $p(ax_i)$ of axioms, the diagnosis algorithm first calculates the a-priori probability $p(\mathcal{D}_j)$ that $\mathcal{D}_j$ is the target diagnosis. Since all axioms fail independently, this probability can be computed as [1]:

$$p(\mathcal{D}_j) = \prod_{ax_n \notin \mathcal{D}_j} p(ax_n) \prod_{ax_m \in \mathcal{D}_j} 1 - p(ax_m) \qquad (1)$$

The prior probabilities for diagnoses are then used to initialize an iterative algorithm that includes two main steps: (a) selection of the best query and (b) update of the diagnoses probabilities given the query feedback.

According to information theory the best query is the one that, given the answer of an oracle, minimizes the expected entropy of a the set of diagnoses [1]. Let $p(X_i = v_{ik})$ where $v_{i0} = no$ and $v_{i1} = yes$ be the probability that query X_i is answered with either no or yes. Let $p(\mathcal{D}_j | X_i = v_{ik})$ be the probability of diagnosis $\mathcal{D}_j$ after the oracle answers $X_i = v_{ik}$. The expected entropy after querying X_i is:

$$H_e(X_i) = \sum_{k=0}^{1} p(X_i = v_{ik}) \times \\ - \sum_{\mathcal{D}_j \in \mathbf{D}} p(\mathcal{D}_j | X_i = v_{ik}) \log_2 p(\mathcal{D}_j | X_i = v_{ik})$$

The query which minimizes the expected entropy is the best one based on a one-step-look-ahead information theoretic measure. This formula can be simplified to the following score function [1]:

$$sc(X_i) = \sum_{k=0}^{1} p(X_i = v_{ik}) \log_2 p(X_i = v_{ik}) + p(\mathbf{D_i^\emptyset}) + 1 \qquad (2)$$

where $\mathbf{D_i^\emptyset}$ is the set of diagnoses which do not make any predictions for X_i. Since, for a query X_i the set of diagnoses $\mathbf{D}$ can be partitioned into the sets $\mathbf{D^{X_i}}$, $\mathbf{D^{\neg X_i}}$ and $\mathbf{D_i^\emptyset}$, the probability that an oracle will answer a query X_i with either yes or no can be computed as:

$$p(X_i = v_{ik}) = p(\mathbf{S_{ik}}) + p(\mathbf{D_i^\emptyset})/2 \qquad (3)$$

where $\mathbf{S_{ik}}$ corresponds to the set of diagnoses that predicts the outcome of a query, e.g. $\mathbf{S_{i0}} = \mathbf{D^{\neg X_i}}$ for $X_i = no$ and $\mathbf{S_{i1}} = \mathbf{D^{X_i}}$ in the other case. $p(\mathbf{D_i^\emptyset})$ is the total probability of the diagnoses that predict no value for the query X_i. Under the assumption that *both outcomes are equally likely* the probability that a set of diagnoses $\mathbf{D_i^\emptyset}$ predicts $X_i = v_{ik}$ is $p(\mathbf{D_i^\emptyset})/2$.

Since all diagnoses are statistically independent the probabilities of their sets can be calculated as:

$$p(\mathbf{D_i^\emptyset}) = \sum_{\mathcal{D}_j \in \mathbf{D_i^\emptyset}} p(\mathcal{D}_j) \qquad p(\mathbf{S_{ik}}) = \sum_{\mathcal{D}_j \in \mathbf{S_{ik}}} p(\mathcal{D}_j)$$

Given the feedback v of an oracle to the selected query X_s, i.e. $X_s = v$ we have to update the probabilities of the diagnoses to take the new information into account. The update is made using Bayes' rule for each $\mathcal{D}_j \in \mathbf{D}$:

$$p(\mathcal{D}_j | X_s = v) = \frac{p(X_s = v | \mathcal{D}_j) p(\mathcal{D}_j)}{p(X_s = v)} \qquad (4)$$

where the denominator $p(X_s = v)$ is known from the query selection step (Equation 3) and $p(\mathcal{D}_j)$ is either a prior probability (Equation 1) or is a probability calculated using Equation 4 during the previous iteration of the debugging algorithm. We assign $p(X_s = v | \mathcal{D}_j)$ as follows:

$$p(X_s = v | \mathcal{D}_j) = \begin{cases} 1, & \text{if } \mathcal{D}_j \text{ predicted } X_s = v; \\ 0, & \text{if } \mathcal{D}_j \text{ is rejected by } X_s = v; \\ \frac{1}{2}, & \text{if } \mathcal{D}_j \in \mathbf{D_s^\emptyset} \end{cases}$$

We implemented the computation of diagnoses following the approach proposed by Friedrich et al. [2] and employ the combination of two algorithms, QUICKXPLAIN and HS-TREE. The set of minimal hitting sets computed by HS-TREE is used to create a set of minimal diagnoses $\mathbf{D}$. For each diagnosis $\mathcal{D}_i \in \mathbf{D}$ the algorithm gets a set of entailments from the reasoner and computes the set of queries. For each query X_i it partitions the set $\mathbf{D}$ into $\mathbf{D^{X_i}}$, $\mathbf{D^{\neg X_i}}$ and $\mathbf{D_i^\emptyset}$, as defined previously. In the next step the algorithm computes prior probabilities for a set of diagnoses given the fault probabilities of the axioms. To take past answers into account the algorithm updates the prior probabilities of the diagnoses by evaluating Equation 4 for each diagnosis in $\mathbf{D}$. The algorithm stops if there is a diagnosis probability above the acceptance threshold σ or if no query can be used to differentiate between the remaining diagnoses (i.e. all scores are 1). The most probable diagnosis is then returned to the user. If it is impossible to differentiate between a number of highly probable minimal diagnoses, the algorithm returns a set that includes all of them.

Our experiments were performed on artificial examples that were generated taking into account the latest studies reporting typical faults of ontology creators. The average results of the evaluation performed on each test suite (depicted in Figure 1) show that the entropy-based approach outperforms the split-in-half method as well as random query selection by more than 50% for the $|\mathcal{D}_t| = 2$ case due to its ability to estimate the probabilities of diagnoses.

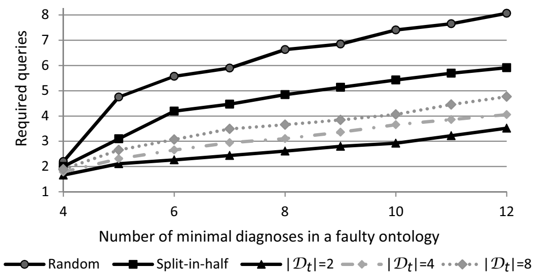

Figure 1. Number of queries required to select the target diagnosis $\mathcal{D}_t$ with threshold $\sigma = 0.95$. Random and "split-in-half" are shown for $|\mathcal{D}_t| = 2$.

REFERENCES

[1] J. de Kleer and B.C. Williams, 'Diagnosing multiple faults', *Artificial Intelligence*, **32**(1), 97–130, (April 1987).

[2] G. Friedrich and K. Shchekotykhin, 'A General Diagnosis Method for Ontologies', in *Proceedings of the 4th International Semantic Web Conference*, pp. 232–246. Springer, (2005).