Ontology Development for Classification: Spirals
A Case Study in Space Object Classification

Bin Liu, Li Yao, Junfeng Wu and Zheyuan Ding
Science and Technology on Information System and Engineering Laboratory, National University of Defense Technology, Changsha, P. R. China

Keywords: Ontology-based Classification, Ontology Development, Classification Ontology, Data-Driven Evaluation.

Abstract: Ontology-based classification (OBC) has been used extensively. The classification ontologies (COs) are the grounds of the OBC systems. It is an urgent call for a method to guide the development of CO, to get better performances for OBC. A method for developing CO named *Spirals* is proposed, taking the development of the ontology for space object classification named OntoStar as an example. First, soft sensing data and hard sensing data are collected. Then, various kinds of human knowledge and knowledge obtained by machine learning are combined to build a CO. Finally, *data-driven evaluation and promotion* assesses and promotes CO. Classification of space object based on OntoStar show that *data-driven evaluation and promotion* increases the accuracy by 4.1%. Meanwhile, OBC is more robust than baseline classifiers with respect to a missing feature in the test data. When classifying space objects with "*size*" missing in the test data, OBC keeps its FP rate, while the baseline classifiers' FP rates increase between 3.9% and 35.5%; the losing accuracy of OBC is 0.2%, while that of baseline classifiers ranges from 1.1% to 69.5%.

1 INTRODUCTION

Recently, Ontology-Based Classification (OBC) has been paid more and more attentions to. OBC accomplishes classifications through deducing on knowledge bases like a human expert, owing to the ontology's ability of expressing domain knowledge and multi-sensor data in a machine-readable format explicitly (Zhang et al., 2013, Belgiu et al., 2014). In addition, OBC keeps robustness in the dynamic open environments (Kang et al., 2015). For these reasons, OBC has been used extensively. In most OBC systems, classifications are realized by instance classification (Gómez-Romero et al., 2015) named *classification ontology* (CO). COs are ground of these systems. They determine the capabilities and performances of the OBC systems. Therefore, it is an urgent call for a suitable method to guide the development of COs, for better performances of the OBC systems and efficient development.

In the last two decades, many methodologies for ontology development have been proposed. They focus on domain problems mainly and have been validated by the developments of specific domain ontologies (Suárezfigueroa et al., 2015). Past studies indicate that it is better to choose a suitable methodology for the ontology development with respect to the domain, concerning the efficiency of the ontology development and performances of the ontology (Haghighi et al., 2013). In terms of CO development, previous methods need to be improved and supplemented. E.g., capabilities of OBC systems are enhanced by embedding the knowledge obtained by machine learning (MLK) into ontologies (Belgiu et al., 2014, Kassahun et al., 2014, Kang et al., 2015, Zhang et al., 2013). On the one hand, the performances of OBC systems and efficiency of ontology development are improved by integrating MLK. On the other hand, the way to embed MLK into the ontology needs to be further studied, for a better combination of MLK and human knowledge. However, current methods of building CO do not assess how well MLK and human knowledge are combined. In addition, MLK is coded into CO by an one-shot behavior. Thus, further improvements of CO are not considered. As a result, the performances of the OBC systems can hardly be further improved. Therefore, it is still necessary to explore better constructions for COs.

To address and solve the problems, a method of developing CO named *Spirals* is proposed, taking the development of CO of Space Object (OntoStar)

Liu, B., Yao, L., Wu, J. and Ding, Z.
Ontology Development for Classification: Spirals - A Case Study in Space Object Classification.
DOI: 10.5220/0006240002250234
In *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST 2017)*, pages 225-234
ISBN: 978-989-758-246-2

as an example. Contributions of this paper include the following aspects. First, a whole workflow of developing CO is presented, including the specific means of acquiring ontological data, how to use background knowledge, contextual knowledge and MLK to develop a CO in the first step, and how to evaluate CO. Second, unordered machine learning *classification rules* (CRs) are learned and coded with Semantic Web Rule Language (SWRL) (Horrocks et al., 2004), for comprehensibility and modification of the ontology. Last, making full use of data for ontology development, *data-driven evaluation and promotion* for CO is proposed to assess how well MLK and human knowledge are combined and to improve the ontology further.

2 RELATED WORK

2.1 Ontology based Classification

OBC has been applied extensively in the past, such as emotion recognition (Zhang et al., 2013), classifying adverse drug reactions and epilepsy types (Zhichkin et al., 2012, Kassahun et al., 2014), classifications in remote sensing (Belgiu et al., 2014, Moran et al., 2017), classifying chemicals (Magka, 2012, Hastings et al., 2012), and etc.

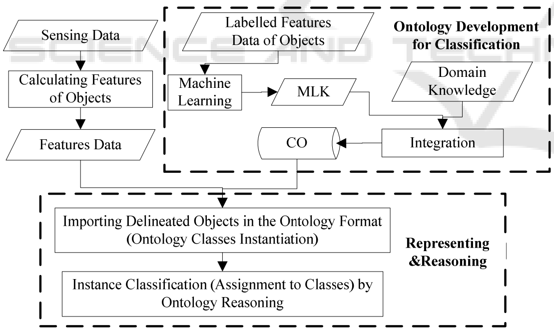

Figure 1: Common architecture of OBC

In most OBC systems, the categories of an object is derived by two steps. In the first step, namely *ontology development for classification*, CO is built using domain knowledge and MLK. In the second step, namely *representing and reasoning*, when data of the object are imported into CO or some features of the object are modified, the step reasoning about CO is started, aiming at finding matches to the descriptions of the object, and categorizes the object. Most OBC systems are under the architecture of **Figure 1**. The development of CO is a precondition of realizing an OBC system. Most COs integrate MLK. However, the effectiveness and impacts of the integration are not validated. Besides, the one-shot integration provides no mechanism to improve CO further. In addition, MLK is only for direct classification, whose type is so few that it can't be used to infer other kinds of information.

2.2 Ontology Development for OBC

2.2.1 Ontology Development

Previous studies for developing domain ontologies offer experience and references for developing CO. They specify necessary steps ontology construction (Casellas, 2011) including acquiring ontological data, converting data to ontological elements, ontology formalization and checking. Other formed guidelines include converting data to ontological elements by learning (Maedche, 2002) and reusing ontologies (Suárezfigueroa et al., 2015). However, they do not provide specific approach to realize the guidelines for CO. Some applications of OBC also provide practices and experience for the emerging approach. E.g., data are pre-processed to extract more features after obtained (Zhang et al., 2013, Belgiu et al., 2014). Decision trees are deployed to convert the data to the ontological elements, by generating CRs which are used to deduce the more specific type of the objects (Moran et al., 2017). Concepts and definitions are formalized by OWL and rules. Reasoners are used to realize the OBC systems.

2.2.2 Capturing CO's Knowledge from Data

MLK makes COs more complete. In most COs, it is learned from structured data for direct classification and obtained by two ways. One way is capturing the characterizations of concepts by machine learning as definitions, e.g., by clustering (Maillot and Thonnat, 2008) and instantiating the qualitative descriptions of concepts in classification (Belgiu et al., 2014). Another is learning rules for direct classification, e.g., rules extracted from decision trees (Zhang et al., 2013). It is a trend of expressing MLK with SWRL in COs (Belgiu et al., 2014). Knowledge related to classifications can be obtained by learning too. It is not considered in current OBC systems, including relations between concepts and attributes (Fürnkranz and Kliegr, 2015, Mansinghka et al., 2015).

2.2.3 Ontology Evaluation

Ontology evaluation is calculating the degree of the fitness for use, e.g., expressing domain knowledge (Brewster et al., 2004). According to the evaluation, how the user's requirements are satisfied can be estimated. Developers also assesses the developing ontology for improvement. Methods for ontology evaluation emerged in the past can be categorized into four types, including the gold standard-based evaluation, corpus-based evaluation, the task-based evaluation and the criteria-based evaluation (Raad and Cruz, 2015). Meanwhile, criteria for ontology evaluation are presented (Sánchez et al., 2015).

Manual work in evaluating shall be as less as possible to make the evaluation as objective and efficient as possible. However, subjectivity is a common major limitation to current evaluations (Hloman and Stacey, 2014). Because every evaluation can be regarded as a measurement for the ontology (Brank et al., 2005), subjectivity can be reduced if every process in the ontology evaluation can be quantized. On the one hand, it is thought to be impossible to compare the ontology evaluation to the evaluations in Information Retrieval, for precision and recall cannot be easily used in ontology evaluation (Brewster et al., 2004). On the other hand, task-based ontology evaluation and data-driven ontology evaluation are thought to be more effective methods of evaluation for the ontology containing MLK (Dellschaft and Staab, 2010). Fortunately, the classification results obtained by CO containing MLK can be compared to that obtained by the referred ontology (gold standard ontology). Thus, the evaluation results can be obtained (Raad and Cruz, 2015). With the accumulated labelled data, it is possible to evaluate the classification by the labelled data, similar to the data-driven evaluation of domain ontology (Brewster et al., 2004).

3 *SPIRALS*: DEVELOPING CO

The method for developing CO named *Spirals* will be described in this section, taking the Ontology for Classification of Space Object (OntoStar) as an example. As its name suggests, *Spirals* is a cyclic process which optimizes the knowledge in CO. It summarizes and provides ways to acquire data for the developing CO, emphasizes the reuse and share of existing domain knowledge, addresses the integration of MLK, and quantizes the evaluation for CO. The data acquisition in *Spirals* provides data for the learning and the evaluation. The evaluation verifies and validates the developed ontology, and assesses how the different kinds of knowledge are running in with each other. The evaluation provides cues for the further improvement of the ontology. The whole workflow of *Spirals* is shown in **Figure 2**.

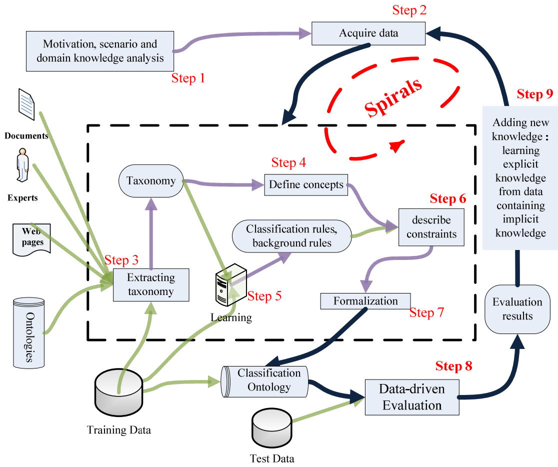

Figure 2: The workflow of *Spirals*.

As shown in **Figure 2**, there are 9 steps in *Spirals* as follows.

Step 1, analyzing motivation, scenario and domain knowledge, including identifying the scope of the domain, identifying the scenario and requirements of classification, and identifying the intended users and use of the classification.

Step 2, acquiring ontological data, such as relational database of the objects to be classified, documents, web pages, domain ontologies and experts.

Step 3, extracting taxonomy, including extracting terminology, identifying concepts and identifying properties from documents, experts and web pages. Reusing existing domain ontologies and obtaining taxonomy from databases by reverse engineering.

Step 4, analyzing ontological data to obtain knowledge for defining concepts in the taxonomy.

Step 5, obtaining knowledge by machine learning for CO.

Step 6, adding constraints to concepts in the taxonomy using MLK.

Step7, representing concepts with OWL and expressing the learned rules by SWRL to obtain the formalized CO.

Step 8, representing test data by CO, performing data-driven evaluation for CO with the aid of ontology reasoner, including consistency checking, non-trivial concept validation and verification.

Step 9, learning new CRs for the ontology according to the evaluation until the performances of OBC achieve the expectations.

To realize the four common and most important activities in the abstract methodology for developing ontology, namely acquiring data, converting data to ontological elements, formalization and ontology checking, *Spirals* proposes acquiring ontological data from multiple sources, learning unordered rules for classification and learning rules relative to the classification, representing concepts with OWL and expressing the learned rules by SWRL, and using data-driven evaluation to check CO, respectively.

3.1 Acquiring Data for Developing CO

Ontological data are a basis of developing CO. They are very important to the learning and evaluation for the building ontology. Ontological data sources such as databases, corpuses, web pages, expert knowledge and domain ontologies, can be used to extract domain knowledge and learn knowledge about classification for the developing CO. These data are from multiple sources, and are obtained by different ways. But ultimately, they are either from *soft sensing* or from *hard sensing*. Web crawling, human scouting and analysis are *soft sensing*. Collecting data by physical sensors such as Nuclear Magnetic Resonance Spectrometer or Radar is *hard sensing*.

The major focus of previous *hard sensing* for classification is capturing data and extracting features of the objects themselves. E.g., in developing COs in (Zhang et al., 2013, Belgiu et al., 2014), *hard sensing* data about the objects themselves are collected by physical instruments of the specific domain first. Then, features of the objects are extracted from the *hard sensing* data such as shape of the buildings. Finally, feature selection is applied to reduce the dimension of the features for machine learning.

Some common features which can be extracted from *hard sensing* data are shown in the following.
(1) features of the time and frequency (e.g., peak alpha frequency, power spectral density, center frequency, etc.);
(2) statistical features (e.g. standard deviation, mean value, kurtosis, skewness, etc.);
(3) nonlinear dynamical features (e.g., C0-Complexity, kolmgolov entropy, Shannon entropy, the largest lyapunov exponent, etc.).
E.g., when a sequence of Radar Cross Section (RCS) of a space object is obtained, the statistical features such as the mean value or the deviation of the sequence can be computed and used to analyze the space object.

Because the information obtained by sensors of the same type is very limited, it is a bottleneck to increase the accuracy of classification by enhancing the precision of physical sensors. Expanding the sources of information to provide more features can increase the accuracy of classification. *Soft sensing* data can not be ignored for this purpose. Firstly, *soft sensing* data can be used to describe the objects more precisely and used in the machine learning. Secondly, when the *soft sensing* information is employed in the *representing and reasoning* of OBC, more information about the objects can be inferred. E.g., in the BIO-EMOTION ontology (Zhang et al., 2013), more contextual data such as information about the social environment can be collected for more accurate emotion recognition.

3.2 Extracting Knowledge from Corpus

Reusing the analyzed knowledge from corpuses helps to construct an initial CO. The knowledge includes taxonomy and knowledge related to classifications.

The structure of concepts is expressed by taxonomy from specific to general. Knowledge is expressed by taxonomy in a more concise format and higher abstraction level (Di Beneditto and De Barros, 2004), so effective reasoning for OBC is facilitated. Taxonomies are mostly built by experts. A set of key features or attributes are the basis of building taxonomies by experts. The most well-known taxonomy built by experts is the classical Linnaeus biological taxonomy (Godfray, 2007). Taxonomy is used express MLK for classification as definitions of concepts in (Belgiu et al., 2014), and used for hierarchical classification in (Ruttenberg et al., 2015). There are expert-built taxonomies classifying space objects, e.g., the taxonomies of artificial space object (Fruh et al., 2013, Ruttenberg et al., 2015) and the taxonomy in the domain ontology of space objects (Cox et al., 2016).

Knowledge related to classification includes the knowledge characterizing the objects, contextual knowledge and the relations between features. The knowledge related to classification is described directly or indirectly in some models such as learned models, mathematical models and ontologies. E.g., characterizations of space objects are learned (Howard et al., 2015) and are described in models of space objects (Han et al., 2014, Henderson, 2014) respectively, relations between the elements of space object surveillance (Pulvermacher et al., 2000) can be used as contextual knowledge, and the feature deduction for satellites (Mansinghka et al., 2015) learns relations between features.

An initial ontology for space object classification (OntoStar) containing a taxonomy of space objects and the knowledge related to classification is shown by Protégé[1] in **Figure 3**.

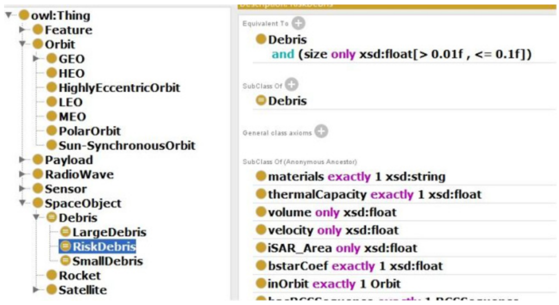

Figure 3: Initially built OntoStar.

The left part of **Figure 3** shows part of the taxonomy of OntoStar which represents concepts from general to specific. The right part shows descriptions of the selected concept "*RiskDebris*". E.g., the description "*Debris and (size only xsd:float[>0.01f,<=0.1f])*" means the range of its size is (0.01,0.1].

3.3 Learning Unordered Classification Rules and Background Rules Related to Classification

As pointed out in Section 2.2.2, MLK plays important roles in building CO. MLK in current COs is mainly knowledge used for direct classification. This kind of knowledge whose form is unordered rules in *Spirals*, is called *classification rules* (CRs). Besides CRs, background rules related to classification can also be mined for CO and are considered in *Spirals*, which include the relations between features represented by rules and the rules related to classification.

3.3.1 Learning Unordered CRs

COs built manually can hardly be perfect. Obtaining knowledge by machine learning for classification can further improve completeness of knowledge in COs. In addition, the enriched descriptions in CO make OBC more robust. Due to the advantages of SWRL mentioned in Section 2.2.2, CRs will be learned in *Spirals*. There are two types of learned CRs, decision list (ordered rules) and rules set (unordered rules) (Han et al., 2011). Because the SWRL rules in CO are unordered, the learned rules shall also be unordered. Rules extracted from C4.5 decision trees (Quinlan, 2014) are unordered and are chosen as the learned rules, because they are easy to understand and easy to be expressed by SWRL, and also because an open implementation of C4.5 is provided by WEKA[2].

The CRs in *Spirals* are learned hierarchically, with the guidance of CO. Guided by the hierarchy, the learning concentrates on a smaller range, and gains smaller decision trees. The rules obtained by the learning will be used to deduce the objects' type from general to specific corresponding to the taxonomy in CO. Therefore, when deciding the more specific type of an object, only rules which deduce the more specific type of the object from the object's known types in the taxonomy will be explored. So the searching space is expected to be smaller. E.g., the following rules expressed in SWRL-style are learned hierarchically for OntoStar from the dataset RSODS (details of the dataset will be described in Section 4).

- *so:SpaceObject(?S), so:rcs(?S,?R), ?R>7.5, so:inOrbit(?S,?O), so:GEO(?O) → so:Satellite(?S) [Annotations: Source=learn from RSODS by J48, Confidence=1.0]*
- *so:Satellite (?S), so:inOrbit(?S,?O), so:apogee(?O,?A), 39980f <= ?A <40020f, so:inclination(?O,?I), 63.4f <= ?I < 63.5f → so:Communication_Satellite(?S) [Annotations: Source= learn from RSODS by J48, Confidence=0.932]*
- *so:Satellite (?S), so:inOrbit(?S,?O), (not(so:MEO or so:EllipticalOrbit))(?O), so:owner(?S,?C), name(?C,?N), swrlb:notEqual(?N,"NRO"^^xsd:string), so:inclination(?O,?I), ?I>87.5f, so:perigee(?O,?P), ?P>765f → so:Reconnaissance_Satellite(?S) [Annotations: Source= learn from RSODS by J48, Confidence=0.8]*

The meanings of the above rules are comprehensible. E.g., the first rule can be read as: *if an object is a SpaceObject, its rcs is greater than 7.5 and it is in a GEO, then it is a Satellite; the annotations indicate that the rule is learned from RSODS by J48 and the rule's Confidence is 1.0.*

3.3.2 Mining Background Rules Related to Classification

Background rules related to classification can infer more information in OBC. Therefore, they shall not be ignored. Some background knowledge such as the relations between features and the definitions of features, can be learned and mined.

1 http://www.standford.edu/protege

2 http://www.cs.waikato.ac.nz/ml/weka/

For some important features of the objects to be classified, definitions can be learned if there is no specific knowledge about them currently. E.g., the definitions of orbits are very important background knowledge in the classification of space objects, but definition of deep highly eccentric orbit is not clear when building OntoStar, so the following rule learned from RSODS by J48 is used to define the orbit in OntoStar temporarily, instead of leaving the definition missing.

➢ *so:Orbit(?O), so:apogee(?O,?A), ?A > 1261, ?A > 70157, so:eccentricity(?O,?E), ?E > 0.291845, so:SpaceObject(?S), so:inOrbit(?S,?O) → so:DeepHighlyEccentric_Orbit(?O) [Annotations: Source= learn from RSODS by J48, Confidence=0.9090909090909091]*

When integrating multiple-source data into CO, there may be missing data if some sensors are deactivated. So some features important to the classification may be missing. At this time, relations between features play a part in estimating the missing features from known features, and can get more accurate values than general methods of data imputation. When there is no domain knowledge of inferring features which are always missing, it is an optional way to learn relations between the features and other features and use the relations to estimate the missing features. E.g., the approximate relation between the feature "*rcs*" and "*size*" of space objects can be discovered by fitting from RSODS, which is represented by the following rule.

➢ *so:rcs(?X,?RCS), swrlb:power(?P,?D,0.5f), so:SpaceObject(?X), swrlb:divide(?D,?RCS,0.79f), swrlb:subtract(?S,?P,2.57E-13f) -> so:size(?X, ?S) [Annotations: Source: learn from RSODS by Fitting, Corr_Coef=0.9999]*

Background knowledge relative to classifications can even be learned from ontologies when there are ontological data. E.g., when multiple-source data of space objects are represented by the initial OntoStar, methods like semantic rule mining (Fürnkranz and Kliegr, 2015) can be deployed to learn relations between entities and properties in the ontology. An example of this kind of learned rule is shown below.

➢ *so:SpaceObject(?S), so:Organization(?O), so:owner(?S,?O), so:Organization(?H), so:affiliate(?O,?H) → so:owner(?O,?H)*

The above rule has its realistic meaning. It can be read as the following: *if the owner of the space object S is the organization O, and O is a affiliate of the organization H, then H is also the owner of S.*

3.4 Data-driven Evaluation and Promotion for CO

As addressed in Section 2.1, validations of the effectiveness of MLK and mechanisms for further improvement of CO shall be considered. This requirement can be met by applying evaluation and promotion to the ontology development. The problem of evaluating CO is assessing how CO suits the classification. Accuracy, consistency, efficiency and completeness are the most important criteria for evaluating CO, among the eight ones summarized in (Sánchez et al., 2015), because the evaluation aims at calculating the influence of CO to the capability of the OBC system. Consistency of CO can be done by ontology reasoning. Efficiency is determined by the ontology reasoning. Therefore, only accuracy and completeness are discussed in this paper.

Because ontology evaluation is also a kind of measurement $\mathcal{M}:\mathcal{O}\rightarrow[0,1]$ (Brank et al., 2005), its results can be quantized to be more objective. In order to reduce subjectiveness and raise efficiency of the evaluation for CO, data-driven evaluation for CO is proposed, whose results are quantized. Different from the data-driven evaluation (Brewster et al., 2004) for domain ontology which tests the expressiveness of the domain ontology, the data-driven evaluation for CO tests CO's abilities of classification by comparing the results of OBC to the labels in the test data. Although the measurement of CO is obtained indirectly, testing for CO is direct.

There are indexes for evaluating classifications, such as accuracy, precision, recall, AUC, etc. (Han et al., 2011). An intuition of ontology evaluation is, that precision reflects the ratio of correct knowledge among the knowledge to be validated and verified (Brewster et al., 2004). The precision obtained in testing is defined in the following (TP: true positives; FP: false positives).

$$\text{precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{1}$$

Precision is the ratio of the objects of a given type correctly classified among the objects classified as this type. The more accurate the knowledge is, the higher the precision. Hence, precision reflects the accuracy of the knowledge of a given concept in CO.

The recall of a classifier obtained in testing is defined in the following (FN: false negatives).

$$\text{recall} = \frac{\text{TP}}{\text{TP+ FN}} \tag{2}$$

Recall is the ratio of the objects of a given type correctly classified among the objects which belong to the given type and have been classified. The more complete the knowledge is, the higher the recall. Hence, recall reflects the completeness of the knowledge about a given concept in CO.

Although a test instance classified incorrectly owes to the lack of correct knowledge, recall can not be analogized to precision. If there are unclassified objects, it is uncertain whether there is correct knowledge to classify the objects (maybe there is correct knowledge, but it is deactivated in the reasoning when conflicting with other knowledge). Despite of this annoyance that recall only measures one aspect of completeness, there is AUC, reflecting another aspect of completeness. When the recall of a given type is very high and AUC is very low, we know there are unclassified objects. It indicates CO lacks knowledge to classify the objects.

Accuracy measures comprehensive capability of the classifier. It reflects both completeness and accuracy in some degrees. So, accuracy of OBC can be used to assess the overall ability of CO. The accuracy obtained in testing is defined in the following (TN: true negatives; TN: true negatives).

$$\text{accuracy}=\frac{TP+TN}{TP+FN+TN+FP} \tag{3}$$

It is indicated by the data-driven evaluation that how CO needs to be improved. When the precision of a given concept is low, the knowledge about it is not accurate. When the recall of a given concept is low, more knowledge about it is necessary. When the recall is very high but the AUC is very low, it is sure that the knowledge about the concept often conflicts with other knowledge in CO, requiring to be refined.

4 EXPERIMENTS AND RESULTS

The dataset of space objects RSODS which is derived from the datasets NORAD_Catalog and UCS_Satellite will be used for the experiments. NORAD_Catalog describes space objects using the following attributes/features: *cospar_id*, *nord_id*, *period*, *perigee*, *apogee*, *eccentricity*, *rcs*, *amr* and labels. It contains 8071 samples with 3 labels: *Debris*, *Rocket Body* and *Satellite*. It is used to analyze orbital distributions of different types of space objects in (Savioli, 2015), and also used to derive new features' data of satellites in (Mansinghka et al., 2015). UCS_Satellite describes active satellites with the following attributes/features: *cospar_id*, *nord_id*, *period*, *perigee*, *apogee*, *eccentricity*, *orbit type*, *orbit class*, *longitude*, *power*, *dry mass*, *launch mass*, *launch vehicle*, *launch site*, *owner*, *contractor*, *users* and *purposes*. The attribute *purposes* is treated as the dataset's labelling attribute. It contains 1346 samples with 19 labels, and contains 1267 samples with at least one attribute of missing value. It is used to describe satellites in (Ruttenberg et al., 2015). NORAD_Catalog and UCS_Satellite contain 318 identical space objects. The two datasets are merged into RSODS, through left join on the attribute *cospar_id*. The 318 identical space objects in RSODS are labelled the same as their labels in UCS_Satellite.

RSODS contains 9099 samples. It has 21 types of space objects which are *Rocket* (7.6%), unknown-type *Satellite* (15.9%), *Satellite* of specific purposes (14.8%) such as *Communication Satellite* and *Global Position Satellite*, and *Debris* (61.6%). 9020 records in RSODS have at least one missing value. So RSODS is incomplete. The other 79 records without missing values in RSODS are all data of satellites of specific purposes.

OntoStar is initially developed under the guidance of *Spirals*. Concepts and the taxonomy necessary to the classification for space objects are specified. CRs and background rules are learned from RSODS. Data of space objects are imported to OntoStar using OWLAPI with the form of instances to learn more semantic rules. An OBC system of space objects named Clairvoyant is built upon OntoStar by integrating the ontology reasoning tool pellet[3]. OntoStar is evaluated by Clairvoyant with RSODS. Finally, the initially built OntoStar is further improved according to the evaluation.

Integrating Clairvoyant into WEKA, the indexes of classifying RSODS can be computed by WEKA. Ten-fold cross validations (TCV) (Han et al., 2011) are performed on RSODS using various approaches. In every validation, data are split into training data (90 %) and test data (10 %) in the TCV. All MLK of OntoStar is obtained from the training data. The baseline classifiers are C4.5 (Quinlan, 2014), SVM (Keerthi et al., 2006), Ripper (Cohen, 1995), Bayesian Network (Friedman et al., 1997), Random Forests (Breiman, 2001) named RF in the table, Backpropagation Neural Network (Erb, 1995) named BPNN and Logistic Model Trees (Landwehr et al., 2005). All the baseline classifiers are set to use their default parameters in WEKA, except that RandomForest is setup with 50 trees and BPNN is setup with 4 hidden layers.

When applying the *data-driven evaluation and promotion* of Spirals to build OntoStar, the training data are split into data for learning (90%) and

[3] https://github.com/Complexible/pellet

validating data (10%). CRs are then obtained by C4.5 with confidence factor of 0.25 from the data for learning. After that, the validating data are used to test the OntoStar with the CRs integrated. Finally, the learned CRs of concepts in OntoStar whose recall or precision in the evaluation are less than 60% are replaced with more specific rules obtained by C4.5 with confidence factor of 0.5.

(a) Some common indexes for classification

The results are shown in **Table 1**. All the indexes except accuracy are weighted indexes. OBC_{Man} represents the classification based on the OntoStar containing no learned rules. OBC represents the classification based on the OntoStar built without *data-driven evaluation and promotion*. OBC_{Eval} represents the classification based on the OntoStar built by Spiral.

Table 1: results of TCV on RSODS.

M / EI	OBC_{Man}	OBC	OBC_{Eval}	C4.5	Ripper
accuracy	11.0%	86.3%	90.4%	85.2%	91.9%
TP rate	70.9%	88.2%	90.7%	85.2%	91.9%
FP rate	1.9%	3%	2.7%	12.1%	3.7%
precision	78%	87.2%	90.7%	80.0%	91.7%
recall	70.9%	88.2%	90.7%	85.2%	91.9%
AUC	83%	91.6%	93.9%	93.7%	96.6%
T	0.0	*1.2*	**227.7**	*3.9*	*3.7*
M / EI	**Bayesian Network**	**SVM**	**BPNN(4 hidden layers)**	**RF (50 trees)**	**Logistic Model Trees**
accuracy	90.0%	90.2%	84.6%	89.9%	90.1%
TP rate	90.0%	90.2%	84.6%	89.9%	90.1%
FP rate	2.6%	4.1%	4.6%	8.3%	3.9%
precision	90.5%	90.1%	81.0%	89.5%	90.0%
recall	90.0%	90.2%	84.6%	89.9%	90.1%
AUC	99.0%	95.1%	95.9%	99.1%	98.5%
T	0.2	*3.5*	**345.6**	**244.2**	**3083.2**

M: methods; EI: weighted average evaluation index; T: time for learning (seconds)

It can be seen in **Table 1**, that the integration of CRs into CO increases OBC's accuracy by 75.3%, compared to OBC_{Man}. Its accuracy increases 4.1% further by the *data-driven evaluation and promotion*, compared to OBC_{Eval}. OBC_{Eval} outperforms Random Forests, Backpropagation Neural Network, C4.5, SVM, Bayesian Network and Logistic Model Trees, and competes with Ripper.

(b) Robustness with respect to missing feature in the test data

Although the performances of OBC is not as good as that of Ripper, it is robust in the presence of missing features caused by some deactivated sensors in the open environments. OBC rarely depend upon any one path. It usually has several different ways to classify one object, so that there are always other ways to classify an object if one way fails. This ability of OBC is obtained by integrating various kinds of knowledge into CO. In terms of classification for space objects, the feature "*size*" of space objects is difficult to capture in reality. It is often missing when classifying the space object. To mimic this situation, the feature "*size*" is set to be missing in the test data. Results of TCV on RSODS with "*size*" missing in the test data using various approaches are shown in **Table 2**.

Table 2: TCV on RSODS without "*size*" in the test data.

M / EI	OBC_{Eval}	C4.5	RF (50 trees)	Ripper
accuracy	90.2%	*69.0%*	88.8%	*82.7%*
TP rate	90.4%	*69.0%*	88.8%	*82.7%*
FP rate	2.7%	**47.6%**	**12.2%**	**20.8%**
precision	90.5%	**59.9%**	88.5%	*82.6%*
recall	90.4%	**69.0%**	88.8%	*82.7%*
AUC	93.8%	*86.9%*	99%	*82.5%*
M / EI	**Bayesian Network**	**SVM**	**BPNN (4 hidden layers)**	**Logistic Model Trees**
accuracy	*86.2%*	**39.3%**	**15.1%**	*74.6%*
TP rate	*86.2%*	**39.3%**	**15.1%**	*74.6%*
FP rate	**13.8%**	**25.3%**	**7.8%**	**26.7%**
precision	*86.4%*	**52.6%**	**66.7%**	**68.4%**
recall	*86.2%*	**39.3%**	**15.1%**	*74.6%*
AUC	98.7%	**68.1%**	**61.1%**	*80.6%*

M: methods; EI: weighted average evaluation index

It can be seen in **Table 2** that OBC outperforms all baseline classifiers. Comparing the result of **Table 2** to **Table 1**, it can be seen, that a missing feature paralyze all the baseline classifiers to some extent, whereas OBC who has failed at some attempts will find other ways to proceed. When missing "*size*", SVM and Backpropagation Neural Network are almost paralyzed, C4.5, Ripper and Logistic Model lose a lot of performances especially in accuracies and FP rates, accuracies of Bayesian Network and Random Forests drop 1.1% and 3.8% respectively, FP rates of Bayesian Network and Random Forests increase 11.2% and 3.9% respectively, while OBC lose a few performances. It can be seen in **Table 2** that OBC's accuracy drops 0.2% and its FP rate stays the same.

5 CONCLUSIONS

A method for developing classification ontology (CO) named *Spirals* is proposed, taking the development of Ontology for Classification of Space Object (OntoStar) as an example. *Spirals* is composed of a set of activities among which is a cyclic subsequence of activities. It proposes acquiring soft sensing data and hard sensing data of the objects to be classified, and extracting features from these data. Then, a CO is initially developed upon the knowledge base extracted from ontological data such as corpuses, experts, databases and domain ontologies. After that, data-driven evaluation is proposed to evaluate CO and to guide the further improvement and promotion of the ontology. Data acquisition, data exploitation, ontology evaluation and mechanism for ontology promotion are addressed in *Spirals*. Especially, not only in the phase of learning, but also in the ontology evaluation, data are made full use of, aiming at enhancing the efficiency of the ontology development and the performances of OBC.

OntoStar is developed under the guidance of *Spirals*. The OBC system for space object named Clairvoyant is built upon OntoStar. Experiments conducted on the dataset of space objects RSODS show that Clairvoyant is competitive against baseline classifiers, in terms of common indexes of classification and robustness with respect to missing an important feature. The results also show that *Spirals* can further improve the performances of OBC. One of the main advantages of OBC, integrating domain knowledge to build the initial CO, is also OBC's main disadvantage, because manual work is required. *Spirals* is still in its exploration and needs further improvements. In the following step, *Spirals* will be extended and applied to develop more COs to further test its effectiveness. It will also be further studied for the optimization of its activities, including expanding the sources and types of MLK, further investigation of the data-driven evaluation and promotion of CO.

ACKNOWLEDGMENTS

This work is framed within the National Natural Science Foundation of China (No. 71371184).

REFERENCES

Belgiu, M., Tomljenovic, I., Lampoltshammer, T. J., Blaschke, T. & Höfle, B. 2014. Ontology-based classification of building types detected from airborne laser scanning data. *Remote Sensing,* 6**,** 1347-1366.

Brank, J., Grobelnik, M. & Mladenic, D. A survey of ontology evaluation techniques. Proceedings of the conference on data mining and data warehouses (SigKDD 2005), 2005. 166-170.

Breiman, L. 2001. Random Forests. *Machine Learning,* 45**,** 5-32.

Brewster, C., Alani, H., Dasmahapatra, S. & Wilks, Y. Data driven ontology evaluation. Proceedings of Int. Conf. on Language Resources and Evaluation, 2004 Lisbon.

Casellas, N. 2011. *Methodologies, Tools and Languages for Ontology Design*, Springer Netherlands.

Cohen, W. W. Fast effective rule induction. Proceedings of the twelfth international conference on machine learning, 1995. 115-123.

Cox, A. P., Nebelecky, C. K., Rudnicki, R., Tagliaferri, W. A., Crassidis, J. L. & Smith, B. 2016. The Space Object Ontology. *Fusion 2016.* Heidelberg, Germany: IEEExplore.

Dellschaft, K. & Staab, S. Strategies for the Evaluation of Ontology Learning. Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge, 2010. S256.

Di Beneditto, M. E. M. & De Barros, L. N. 2004. Using concept hierarchies in knowledge discovery. *Advances in Artificial Intelligence–SBIA 2004.* Springer.

Erb, D. R. J. 1995. The Backpropagation Neural Network — A Bayesian Classifier. *Clinical Pharmacokinetics,* 29**,** 69-79.

Fürnkranz, J. & Kliegr, T. A Brief Overview of Rule Learning. RuleML 2015: Rule Technologies: Foundations, Tools, and Applications, 2015. 54-69.

Friedman, N., Dan, G. & Goldszmidt, M. 1997. Bayesian Network Classifiers. *Machine Learning,* 29**,** 131-163.

Fruh, C., Jah, M., Valdez, E., Kervin, P. & Kelecy, T. 2013. Taxonomy and classification scheme for artificial space objects. Kirtland AFB,NM,87117: Air Force Research Laboratory (AFRL),Space Vehicles Directorate.

Gómez-Romero, J., Serrano, M. A., García, J., Molina, J. M. & Rogova, G. 2015. Context-based multi-level information fusion for harbor surveillance. *Information Fusion,* 21**,** 173-186.

Godfray, H. 2007. Linnaeus in the information age. *Nature,* 446**,** 259-260.

Haghighi, P. D., Burstein, F., Zaslavsky, A. & Arbon, P. 2013. Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings. *Decision Support Systems,* 54**,** 1192-1204.

Han, J., Kamber, M. & Pei, J. 2011. *Data mining: concepts and techniques: concepts and techniques*, Elsevier.

Han, Y., Sun, H., Feng, J. & Li, L. 2014. Analysis of the optical scattering characteristics of different types of space targets. *Measurement Science and Technology,* 25**,** 075203.

Hastings, J., Magka, D., Batchelor, C. R., Duan, L., Stevens, R., Ennis, M. & Steinbeck, C. 2012. Structure-based classification and ontology in chemistry. *J. Cheminformatics,* 4**,** 8.

Henderson, L. S. 2014. *Modeling, estimation, and analysis of unresolved space object tracking and identification.* Doctor Doctoral, The University of Texas at Arlington.

Hloman, H. & Stacey, D. A. 2014. Multiple Dimensions to Data-Driven Ontology Evaluation. *Knowledge*

Discovery, Knowledge Engineering and Knowledge Management. Springer.

Horrocks, Ian, Patel-Schneider, Peter, F., Boley, Harold, Tabet, Said, Grossof & Benjamin 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *World Wide Web Consortium*.

Howard, M., Klem, B. & Gorman, J. RSO Characterization with Photometric Data Using Machine Learning. Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, held in Wailea, Maui, Hawaii, September 15-18, 2014, Ed.: S. Ryan, The Maui Economic Development Board, id. 70, 2015. 70.

Kang, S. K., Chung, K. Y. & Lee, J. H. 2015. Ontology-based inference system for adaptive object recognition. *Multimedia Tools & Applications,* 74**,** 8893-8905.

Kassahun, Y., Perrone, R., De Momi, E., Berghöfer, E., Tassi, L., Canevini, M. P., Spreafico, R., Ferrigno, G. & Kirchner, F. 2014. Automatic classification of epilepsy types using ontology-based and genetics-based machine learning. *Artificial intelligence in medicine,* 61**,** 79-88.

Keerthi, S., Shevade, S., Bhattacharyya, C. & Murthy, K. 2006. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation,* 13**,** 637-649.

Landwehr, N., Hall, M. & Frank, E. 2005. Logistic Model Trees. *Machine Learning,* 59**,** 161-205.

Maedche, A. D. 2002. *Ontology Learning for the Semantic Web*, Kluwer Academic Publishers.

Magka, D. Ontology-based classification of molecules: A logic programming approach. Proceedings of the SWAT4LS conference, 2012.

Maillot, N. E. & Thonnat, M. 2008. Ontology based complex object recognition. *Image and Vision Computing,* 26**,** 102-113.

Mansinghka, V., Tibbetts, R., Baxter, J., Shafto, P. & Eaves, B. 2015. BayesDB: A probabilistic programming system for querying the probable implications of data. *Computer Science*.

Moran, N., Nieland, S., Suntrup, G. T. G. & Kleinschmit, B. 2017. Combining machine learning and ontological data handling for multi-source classification of nature conservation areas. *International Journal of Applied Earth Observation & Geoinformation,* 54**,** 124-133.

Pulvermacher, M. K., Brandsma, D. L. & Wilson, J. R. 2000. A space surveillance ontology. *MITRE Corporation, Bedford, Massachusetts.*

Quinlan, J. R. 2014. *C4. 5: programs for machine learning*, Elsevier.

Raad, J. & Cruz, C. A Survey on Ontology Evaluation Methods. International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2015.

Ruttenberg, B. E., Wilkins, M. P. & Pfeffer, A. Reasoning on resident space object hierarchies using probabilistic programming. Information Fusion (Fusion), 2015 18th International Conference on, 2015. IEEE, 1315-1321.

Sánchez, D., Batet, M., Martínez, S. & Domingo-Ferrer, J. 2015. Semantic variance: an intuitive measure for ontology accuracy evaluation. *Engineering Applications of Artificial Intelligence,* 39**,** 89-99.

Savioli, L. 2015. Analysis of innovative scenarios and key technologies to perform active debris removal with satellite modules.

Suárezfigueroa, M. C., Gómezpérez, A. & Fernándezlópez, M. 2015. The NeOn Methodology framework: A scenario-based methodology for ontology development. *Applied Ontology,* 10**,** 107-145.

Zhang, X., Hu, B., Chen, J. & Moore, P. 2013. Ontology-based context modeling for emotion recognition in an intelligent web. *World Wide Web,* 16**,** 497-513.

Zhichkin, P., Athey, B., Avigan, M. & Abernethy, D. 2012. Needs for an expanded ontology-based classification of adverse drug reactions and related mechanisms. *Clinical Pharmacology & Therapeutics,* 91**,** 963-965.