International Journal of Hybrid Information Technology
Vol. 9, No.11 (2016), pp. 369-380
http://dx.doi.org/10.14257/ijhit.2016.9.11.31

Cohesion Metrics for Evaluating Semantic Web Ontologies

Lili Liao, Guohua Shen, Zhiqiu Huang and Fei Wang

College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
lililiao2013@163.com, ghshen@nuaa.edu.cn

Abstract

With the widespread development of the Semantic Web, large-scale ontologies are being developed in more real-world applications to represent and integrate knowledge and data. There is an increasing need for measuring the cohesion of these ontologies for better understanding, maintenance, reuse and integration. The ontology cohesion metrics proposed in this paper can be used as a very useful complementarity of existing ontology cohesion metrics. Specifically, we first propose a set of evaluation metrics to measure the cohesion of ontologies based on directed acyclic graph. Following the framework for software measurement validation, we then evaluate the effectiveness of the proposed metrics theoretically. Finally, we conduct experiments using a set of classical ontologies; the results show that the proposed metrics are reasonable and effective.

Keywords: *Ontologies, Ontology Evaluation, Cohesion Metrics, Directed Acyclic Graph*

1. Introduction

With the development of the Semantic Web technology, information system has become more intelligent than before [1]. A lot of research work focused on how to provide more efficient knowledge sharing services to users, where the ontology technologies play a critical role. Ontologies, as a type of conceptual model, can describe information and data on the semantic level, providing support for knowledge acquisition, representation, analysis and application. There exist a variety of theories and tools for ontology construction, description, and analysis in practice .

On the one hand, the development of ontology description languages and editing tools is beneficial to ontology developers to build ontologies based on specific applications. However, due to the complexity of the application semantics, how to ensure the quality of ontology is an important issue in the process of building ontologies. On the other hand, with the wide application of ontology, the number of ontologies on the Internet has grown explosively. The emergence of a large number of ontologies has made it possible for ontology reuse. But, because there is a big difference in the field of ontology's coverage, understandability, and accuracy, it is difficult for users to ensure the quality of ontologies. Therefore, it is necessary to evaluate the quality of ontologies. According to the measurement results of the ontologies, developers can reconstruct ontologies, and users can select the optimal structure of the ontologies between different systems. For more reliable success in ontology development and use, ontology evaluation should be incorporated across all phases of the ontology life cycle [2].

Thus, ontology evaluation is an important issue that must be addressed for better applying ontologies to Semantic Web and other semantics-driven applications. Ontologies play an irreplaceable role in the development of the Semantic Web. The cohesion of ontology refers to the degree of relatedness of OWL classes conceptually related by the properties such as inheritance relation [3], which is an important aspect of ontology evaluation. Therefore, this paper focuses on the cohesion of ontology evaluation.

ISSN: 1738-9968 IJHIT

In this paper, we propose a set of ontology cohesion metrics to measure the modular relatedness of ontologies based on *directed acyclic graph* (DAG). These metrics consider not only the inheritance relations but also the inferred inheritance relations between the concepts, which complement the previous research on ontology evaluation. We evaluate the functionality of the proposed metrics theoretically and empirically. For theoretical analysis, we investigate the relationship between the ontology and software measurements. We adopt the frameworks for software measurement validation, developed by Kitchenham et al. [4] and Briand et al. [5], as a set of criteria for the evaluation of our new metrics. In the empirical analysis, we conduct experiments using a set of classical ontologies, and the results show our metrics are reasonable and effective. Our work is helpful for ontology engineers to evaluate and select ontology modules; it provides a practical guideline to evaluate qualified ontologies.

The rest of the paper is organized as follows: Section 2 introduces the related work about ontology evaluation. Section 3 gives some preliminaries about ontology representation and criteria of analyzing metrics. In Section 4, we propose the set of our ontology cohesion metrics. Section 5 gives the theoretical validation of these ontologies cohesion metrics. Section 6 describes the analysis of experimental results. Finally, the conclusion and the future work are presented in section 7.

2. Related Work

In recent years, many ontology metrics and measures have been proposed, with a focus on the nature of metrics and measures for ontologies in general. With the development of Semantic Web technology, evaluation of ontology quality has become a key issue for semantic-driven applications.

Some foundational work studies the theoretical validation frameworks. A quality oriented ontology description framework (*QOOD*) [6] and the O^2 and *oQual* models [7] were proposed. The authors created semiotic models for ontology evaluation and validation, and described how measures should be built in order to actually evaluate quality. A framework for metrics called OntoMetric was provided, which defines the relations between the different metrics, their attributes, and the quality attributes they capture. [8], [9] presented the evaluation of OQuaRE, which is a method for ontology quality evaluation which adapts the SQuaRE standard for software product quality to ontologies.

A lot of research work has focused on the ontology evaluation. In [10], Burton-Jones proposed a metric suite for ontology evaluation. These metrics measure the syntactic, semantic, pragmatic and social qualities of Semantic Web ontologies. The score ontology is weighted as the sum of each criterion. They considered that the scale of ontology is the determinant of ontology quality. Tartir et al. [3] introduced a set of metrics for evaluating the schema and the entire knowledge base of an ontology. Some of these metrics include relationship richness, inheritance richness, attribute richness, and class richness. Zhang et al. [11] proposed a set of metrics for measuring ontology complexity. These metrics measure the number of concepts, relations, paths and the average number of relations per concept, the average path per concept, and the average connectivity degree of each concept. Kang et al.[12] used weighted class dependence graphs to represent a given class diagrams, and then present a structure complexity measure for the UML class diagrams based on entropy distance. It considers complexity of both classes and relationships between the classes, and presents rules for transforming complexity value of classes and different kinds of relations into a weighted class dependence graphs. This method can measure the structure complexity of class diagrams objectively.

Cohesion and coupling are two important measures that can be employed for evaluating modular ontologies. There are a few proposals in the literature that investigate the notions of cohesion and coupling for ontologies. [13] presented a set of semantic metrics for evaluating ontologies and modular ontologies. These metrics measure cohesion and coupling of ontologies, which are two important notions in the process of assessing ontologies for enterprise modeling. In [14], the authors introduced the notion of coherent modules and proposed a methodology for partitioning a monolithic ontology to a set of coherent modules, while a coherent module contains a set of concepts that are dependent on each other. In this specific work, only those dependencies are considered that can be derived from the structure of a given ontology. Authors claim this structural method, contrary to any semantic method, can scale up to large ontologies. In Ma et al. [15], a set of metrics are introduced for measuring cohesion of ontologies. These metrics are *Number of Ontology Partitions* (NOP), *Number of Minimally Inconsistent Subsets* (NMIS), and *Average Value of Axiom Inconsistencies* (AVAI). These metrics are obtained based on the semantics of a given ontology rather than its syntax. Since these metrics are introduced for changing ontologies, their focus is mostly on inconsistencies that may be induced by ontology axioms. In Orme et al. [16], the coupling of an ontology is measured by counting the number of external classes that are used for defining classes and properties in the ontology, the number of references to external classes, and the number of includes in the ontology. Sunju Oh et al. [17] proposed cohesion and coupling metrics for ontology modules. They focused on adapting module metrics of software engineering to the domain ontology. Their cohesion metric for a module, *number of relations* (NR), refers to the number of all the relations between classes in the module. Coupling metrics *number of separated hierarchical links* (NSHL) and *number of separated nonhierarchical links* (NSRL) represent the number of disconnected relations during modularization. They assumed that a module is more consistent with original ontology than other modules if fewer relations are disconnected in the module.

3. Preliminaries

In this section, we briefly review the definitions of ontology and ontology modules, and introduce ontology representation and criteria of analyzing metrics.

3.1. Ontology and Ontology Module

An ontology is defined as "a formal, explicit specification of a shared conceptualization" [18] which can represent domain knowledge and facilitate knowledge sharing. The ontology structure O, proposed by Maedche [19], can be described by a five-tuple $O = \langle C, R, H^c, Rel, A^o \rangle$, where C is a set of concepts, R is a set of relations, H^c is a set of concept hierarchies, Rel is a function relating the concepts nontaxonomically, and A^o is a set of ontology axioms expressed in appropriate logical language.

An ontology module is a part of an ontology that is partitioned or extracted from an original ontology [17]. A module is a set of classes, relations, and axioms that are closely related to each other with respect to certain topics. Modules can be used independently of other modules. An ontology module should be self-contained and preserve the definite relations to other modules. Therefore, a qualifying ontology module should have high cohesion and low coupling, which is consistent with the software modules in software engineering. In addition, qualifying ontology modules should preserve the classes, relations, and other axioms of their original ontology after modularization.

Ontology modularization is beneficial to reduce complexity, enhance the understandability, testability, maintainability, and reliability. Ontology module has its own cohesion and can be used independently. Cohesion for ontology module can be measured by calculating the degree of relatedness of different concepts in ontology module.

3.2. Ontology Hierarchy

Although, different researchers have different definitions and implementations of ontologies, there is an important common feature. That is, whatever domains these ontologies are designed for and however they are defined, developers must build the conceptual model for domain knowledge at first, which is a set of concepts and relations that reflects the concept hierarchy.

In ontology conceptual model, concepts hierarchy is typically expressed using DAG (directed acyclic graph) showed in Figure 1. Each node represents a concept; each directed edge represents relation to present the hierarchical structure (inheritance relation) between concepts in ontologies; Arrows point to the parent concept.

We use the following formal notation to represent the terms defined in the ontology conceptual model.

$C= \{c_1, c_2 \ldots c_n\}$: the set of n concepts defined in an ontology explicitly. In other ontologies, concept may be named as “class” or “term”.

$R\ (c_i, c_j)$: the relation between the concept c_i and c_j. In other ontologies, relation may be named as “slot”. It only includes those inheritance relations that reflect the hierarchy of concepts, such as “is-a”.

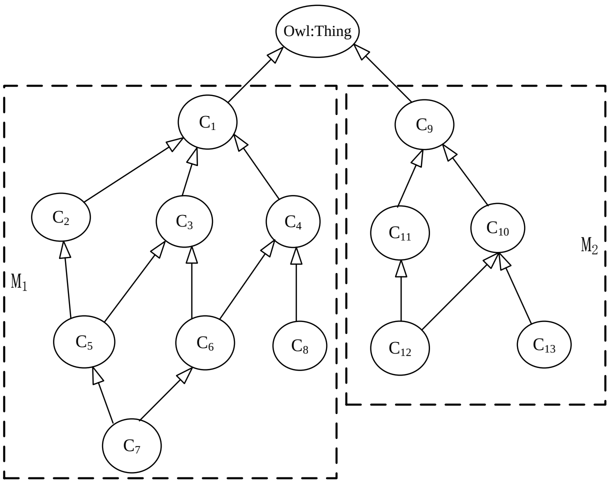


Figure 1. Ontology Hierarchy in DAG

In DAG, a path is a distinct trace that can be taken from a specific particular concept to the most general concept in the ontology, which is the concept without any parent or super class (*e.g.* c_7-c_6-c_3-c_1 in Figure 1). Let $P = \{p_1, p_2 \ldots p_n\}$ be the set of paths each concept has. For example, in Figure 1, there are 13 concepts (n=13), two general concepts (C_1 and C_9). The original ontology is usually partitioned into two modules, M_1 and M_2.

Different path has its own length, thus the path length (*pl*) is defined as the sum of relations (edges) on a path. Let $PL_i = \{pl_{i,1}, pl_{i,2}, \ldots, pl_{i,pi}\}$ be the set of path length of the concept C_i has. Path length of a particular concept indicates the semantic distance between the concept and the general concept.

3.3. Cohesion Metrics and Validation Framework

It is desirable to have a formal model and precise theoretical foundation of metrics evaluation criteria, through which we can evaluate the usefulness and correctness of measures within well-defined contexts. In traditional software measurement, the concept of cohesion refers to the degree to which the elements in a module belong together. Especially for object-oriented software, cohesion refers to the degree of the relatedness or consistency in functionality of the members in a class. Cohesion for ontologies refers to the degree of relatedness of different concepts in an ontology. Cohesion measures separation of responsibilities, independence of components and control of complexity [20]. Classes with strong cohesion are desirable for object-oriented systems. One of the most widely known object-oriented cohesion metrics was proposed by Chidamber and Kemerer [21]: Lack of Cohesion in Methods (LCOM).

According to the previous research [15], [17], measurements must adhere to the science of measurement if they are to gain widespread acceptance and validity. Before using new metrics, validating the metrics on the well-formalized validation framework is required. Traditionally, in object-oriented software engineering area, several metric validation frameworks have been developed and used [4] [5]. Because object-oriented conceptual model is closely associated with ontology representation, software metric evaluation criteria (including such measurement concepts as cohesion, complexity and coupling, *etc*.) for object-oriented software can be regarded as a candidate evaluation framework for ontology quality evaluating.

Kitchenham *et al*. [4] proposed a framework for evaluating software metrics. In this framework, they described the structures of any measure as containing the entities being analyzed (such as classes or modules), the attribute being measured (such as size), the unit used (such as lines of code), and the data scale (nominal, ordinal, interval, or ratio). Units are valid only for interval or ratio data, but they can be adapted for use with ordinal data. In order for a value to have any meaning, the entity, attribute, and units must be specified. The measure must be defined over a specified set of permissible values. To be valid, a measure should have the following:

(1) *Attribute validity*: the entity being analyzed has the attribute.

(2) *Unit validity*: the unit is appropriate for the attribute.

(3) *Instrumental validity*: the underlying model is valid and the instrument is calibrated.

(4) *Protocol validity*: the protocol used for the measurement is valid and prevents errors such as double counting. These characteristics are essential because measures should be, as far as possible, independent of the measurer and the environment.

Briand et al. [5] proposed another set of criteria for assessing metrics. These criteria were proposed to clarify software measurement concepts such as complexity and cohesion. They define some specific properties that the concept should have. For example, as for cohesion metrics, the properties are:

(1) *Non-negativity and normalization*: the value is not negative and the values are comparable between different modules.

(2) *Null value*: if there is no intramodule relationship within an ontology module, the value is 0.

(3) *Monotonicity*: the value may not decrease if a new relation is added to the module.

(4) *Cohesive module*: the value of merged module is never greater than maximum value of the original ontology modules.

In the following, *K-framework* refers to the framework for evaluating software metrics proposed by Kitchenham et al. [4] and *B-framework* refers to the criteria for assessing cohesion metrics proposed by Briand et al. [5].

In this paper, our ontology cohesion metrics will be fitted into these two frameworks for ensuring the metrics proposed theoretically correct.

4. Ontology Cohesion Metrics

In this section, we first introduce some basic definitions about ontologies, which are the prerequisite of our new ontology cohesion metrics. Then we will specifically define our metrics.

4.1. Primitive Definitions

We can define some basic definitions in ontology conceptual model based on DAG as follows.

Definition 1. NOGC (*Number Of General Concepts*): is the number of ontology modules that are partitioned from the original ontology.

For example, in the Figure 1, the ontology can be partitioned into two modules.

Definition 2. LN (*Leaf Node*): is in an ontology means the class has no semantic subclass explicitly defined in the ontology.

For example, in the Figure 1, the ontology has four leaf nodes, C_7, C_8, C_{12} and C_{13}.

Definition 3. RN (*Root Node*): is in an ontology means the class has no semantic super class explicitly defined in the ontology. RN is the general concept in an ontology.

For example, in the Figure 1, the ontology has two root nodes, C_1, C_9.

Definition 4. POI (*Path Of Inheritance*): refers to a path from the leaf node to the root node in DAG.

Definition 5. TNOP (*Total Number Of Path*): is the total number of POI in DAG.

For example, in the Figure1, TNOP=8.

Definition 6. DOI (*Depth Of Inheritance*): is the sum edge number of each POI in DAG. That is, the path length of the leaf node in DAG.

For example, in the Figure 1, from the leaf node C_7 to the root node C_1 has four paths $\{p_1$: $(c_7\text{-}c_5\text{-}c_3\text{-}c_1)$, p_2: $(c_7\text{-}c_5\text{-}c_3\text{-}\mathrm{c}_1)$, p_3: $(c_7\text{-}c_6\text{-}c_3\text{-}c_1)$, p_4: $(c_7\text{-}c_6\text{-}c_4\text{-}c_1)\}$. So $\mathrm{DOI}_{p1} = 3$, $\mathrm{DOI}_{p2} =$ 3,
$\mathrm{DOI}_{p3} = 3$, $\mathrm{DOI}_{p4} = 3$.

4.2. Ontology Cohesion Metrics

In this section, we present our new ontology cohesion metrics: *ontology module cohesion* (Coh), *the original ontology cohesion* (AOC), and *Average Depth of Inheritance Tree of Leaf Nodes* (ADIT-LN). We also give the formula for computing these metrics.

Metric 1: Let *M* be an ontology module. *Coh* is a metric for measuring cohesion of ontology module *M*, and it is defined as follows:

$$\mathrm{Coh}(M)=\begin{cases} 0 & n=0 \\ \dfrac{\sum_{i=1}^{i=n}\sum_{j\succ i}^{j=n} R(c_i,c_j)}{n(n-1)/2} & n>1 \\ 1 & n=1 \end{cases} \tag{1}$$

In (1), *n* is the number of concepts in ontology module *M*. R (c_i, c_j) represents the relation between concepts c_i and c_j.

$\dfrac{n(n-1)}{2}$ is the number of edges of complete connected graph in ontology module *M*’s DAG.

If there exists an inheritance relation or inferred inheritance relation between c_i and c_j, then R (c_i, c_j) is equal 1; if there is no relation between c_i and c_j, R (c_i, c_j) -- 0.

If there is no concept in ontology module, the cohesion is 0. If there is only one concept in ontology module, the cohesion is 1, because this concept itself must be the closest construction.

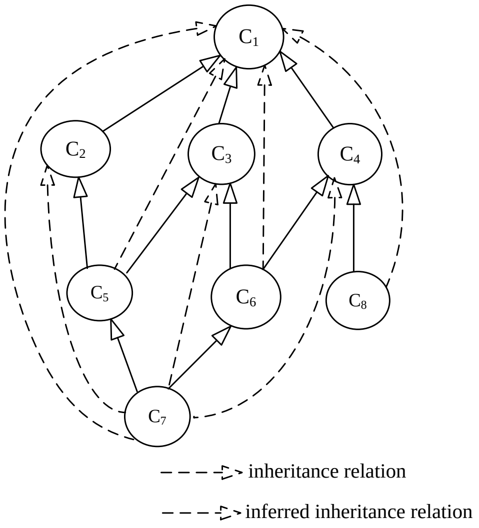


Figure 2. Ontology Module M_1

For example, in Figure 2 ontology module M_1 has ten inheritance relations (solid lines arc) and seven inferred inheritance relations (dotted lines arc). Module M_1 has eight concepts; thus, Coh (M_1) =17/28.

Metric 2: *Average Of Cohesion* (AOC) represents the cohesion of the original ontology. AOC is defined as follows:

$$\mathrm{AOC} = \frac{\sum_{i=1}^{i=n} \mathrm{Coh}(M_i)}{n} \qquad (20)$$

In (2), n refers to the number of modules divided from original ontology. Coh (M_i) is the cohesion of ontology module M_i.

For example, in Figure 1, the original ontology is divided into two ontology modules M_1 and M_2; the cohesion of these modules are calculated as follows: Coh(M_1) = 17/28, Coh(M_2) =7/10. Therefore, the cohesion of the original ontology is: $\mathrm{AOC} = \frac{\mathrm{Coh}(M_1) + \mathrm{Coh}(M_2)}{2}$ =0.5636.

Metric 3: *Average Depth of Inheritance Tree of Leaf Nodes* (ADIT-LN) represents the depth of the ontology concept hierarchy. ADIT-LN is defined as follows:

$$\mathrm{ADIT\text{-}LN} = \sum_{i=1}^{i=n} \frac{DOI_{pi}}{TNOP} \qquad (3)$$

In (3), DOI_{pi} refers to the depth of Inheritance of the path from the leaf node to the root node. n represents the total number of path of inheritance in DAG, $n=TNOP$, which is defined in Section 4.1 Definition 5 and Definition 6. For example, in Figure 1, ADIT-LN=20/8=2.5.

5. Theoretical Analysis and Validation of Ontology Cohesion Metrics

In this section, each metric is examined and validated theoretically using *K-framework* and *B-framework*, which are briefly introduced in Section 3.3.

5.1. Analysis of Coh

To analyze Coh within the *K-framework*, we first define the entity being analyzed as an ontology module, the attribute being measured as the ratio of relations to number of all possible relations, and the unit as the relation. The data scale is interval.

The *Coh* satisfies the properties of *K-framework*, which is described as follows:

(1) *Attribute validity*: the entity (module) has the attribute (relations) that enables us to provide relations explicitly defined in the ontology module to all pairs of relations in module M.

(2) *Unit validity*: the attribute is measured by counting the relations in a module.

(3) Instrumental validity: the instrument is valid as long as our algorithm counts the number of relations defined in an ontology module correctly. Relations are not counted repeatedly.

(4) *Protocol validity*: the measurement as defined in the formal notation given in this paper is consistent, unambiguous, and free from counting errors by counting relations.

We use *B-framework* to validate the *Coh* metric as follows:

(1) *Non-negativity and normalization*: the value of *Coh* is not negative and the values can be compared between different ontology modules.

(2) *Null value*: if there is no intramodule relationship within an ontology module, the value of *Coh* is 0.

(3) *Monotonicity*: the value of *Coh* may not decrease if a new relation is added to the module M. For example, for c_i and c_j, adding a relation between c_i and c_j would increase *Coh* (M) because R (c_i, c_j) >0.

(4) *Cohesive module*: the value of *Coh* of a merged module is never greater than the maximum *Coh* of the original ontology modules. For Module M_1 and M_2, the cohesion of merged module $M_{1\cup 2}$: Coh($M_{1\cup 2}$)=Coh(M_1)+ Coh(M_2)-Coh($M_{1\cap 2}$) and Coh($M_{1\cap 2}$)≥0, consequently, Coh($M_{1\cup 2}$)≤ Coh(M_1)+ Coh(M_2).

The analysis of *AOC* is the similar to that of *Coh*, because the value of *AOC* is directly related to the *Coh*.

5.2. Analysis of ADIT-LN

According to *K-framework*, the cohesion metric ADIT-LN is a direct measure to count the depth of the inheritance tree for all leaf nodes (classes) in ontologies. For this measurement, the entity is the ontology being analyzed, the attribute measured is the average depth of inheritance tree of all leaf nodes, the unit is the depth of inheritance and the data scale is interval.

The ADIT-LN satisfies the properties of *K-framework*, which is described as follows:

(1) *Attribute validity*: the entity (the ontology being analyzed) has the attribute (number of leaf nodes), which can be obtained by using algorithm.

(2) *Unit validity*: the attribute is measured by counting the depth of all leaf nodes.

(3) *Instrumental validity*: the instrument is valid as long as our metrics collecting tool parses and counts the average depth of all leaf classes defined in the ontology correctly.

(4) *Protocol validity*: calculations performed according to the formal notation given in this paper will be free from counting errors by counting the depth of all leaf nodes in DAG and the number of all leaf classes, which is consistent and unambiguous.

We use *B-framework* to validate the ADIT-LN metric as follows:

(1) *Non-negativity and normalization*: the value of ADIT-LN is never negative and the values can be compared between different ontologies.

(2) *Null value*: If there is no intramodule relationship within an ontology, the value of ADIT-LN is 0.

(3) *Monotonicity*: The value of ADIT-LN may not decrease if a new relation is added to the ontology.

(4) *Cohesive module*: the value of ADIT-LN of merged ontology modules is never greater than the maximum ADIT-LN of the original ontology. Merging unrelated modules will not increase or decrease the total number of nodes or the total number of paths in the merged modules. The value of overall ADIT-LN is the average of the values of ADIT-LN

of each ontology module. Therefore, the average of all ADIT-LN of all ontology modules is never greater than the maximum ADIT-LN of the original ontology.

6. Empirical Validation of Ontology Cohesion Metrics

To evaluate whether these ontology cohesion metrics are useful for ontologies, we implemented the metrics and algorithms in Java using Jena, and have conducted preliminary experiments. Seven ontologies are use as the data set to calculate the ontology cohesion metrics. These ontologies come from two ontology libraries: the Swoogle ontology search engine [22] and the protégé ontology library [23]. As a result, we obtain seven independent ontology candidates: koala.owl, travel.owl, beer.owl, DLP3941.owl, univ-bench.owl, dblp.owl, and iso-19115-codelists.owl.

The values of *Coh* are between 0 and 1, as shown in Figure 3. The value of *Coh* is close to 1, indicating that the ontology modules have a higher degree of cohesion, *i.e.*, the relationship between the concepts of ontology modules connected more closely. High cohesion of the ontology module allows for efficient use and maintenance of a set of related concepts and avoiding irrelevant ones.

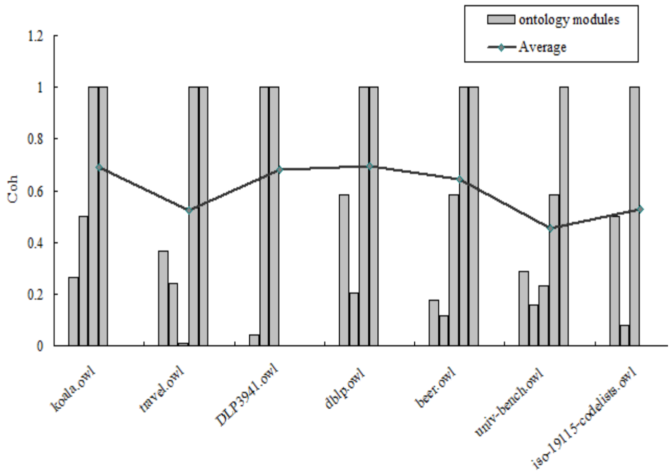


Figure 3. *Coh* in Ontology Modules

Using the AOC ontology cohesion metric to compare the above four ontologies clearly shows how they are intended to be used. Figure 3 shows the specific AOC values in each ontology. According to the figure, it can be clearly seen that the dblp.owl ontology has higher AOC value than other ontologies. This shows that the dblp.owl ontology has higher ontology cohesion than other ontologies, and is relatively easy to understand and reuse.

We also use the ADIT-LN ontology cohesion metric to compare the above the seven ontologies. Figure 4 shows the specific ADIT-LN values in each ontology. According to the figure, the DLP3941.owl ontology has the highest value of Average Depth of Inheritance Tree of Leaf Nodes, which means that the concepts in DLP3941.owl ontology have a higher hierarchical structure. This means that the ontology has more topics than other ones, and the concept organization and aggregation within the ontology are close relatively.

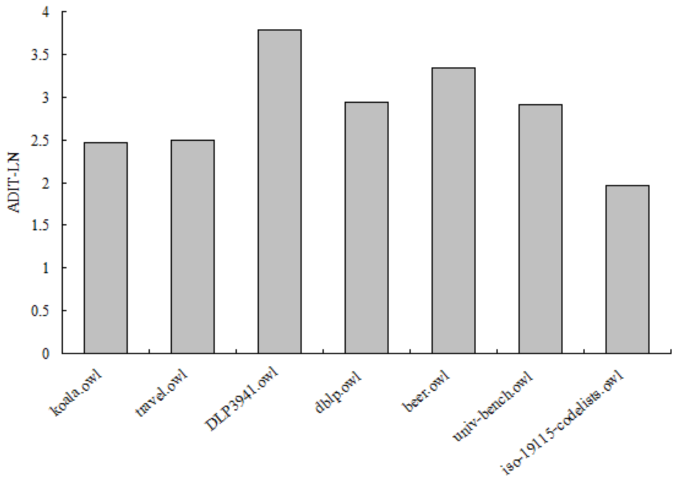


Figure 4. ADIT-LN in Original Ontologies

7. Conclusion

In this paper, we propose a set of new ontology cohesion metrics to measure the cohesion of ontologies based on directed acyclic graph. We validate these metrics theoretically and empirically. Our metrics complement the existing ontology cohesion metrics. These ontology cohesion metrics indicate cohesion quality of ontologies from different perspectives, and can help ontology developers and users to effectively evaluate the quality of ontologies.

This paper mainly focuses on the hierarchical (inheritance) relationship of concepts within the ontology for ontology cohesion. It is interesting to investigate metrics from other structural perspectives. We also would like to develop ontology evaluation frameworks that can effectively evaluate the quality of ontologies in future.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61272083, No. 61100034 and No. 61170043) and the Fundamental Research Funds for the Central Universities of China (Grant No. NS2015093). The authors wish to thank a associate professor Ou Wei for this work.

References

[1] T. Berners-Lee, J. Hendler and O. Lassila. “The semantic web”, Scientific American, vol. 284, no. 5, **(2001)**, pp. 34-43.
[2] F. Neuhaus and A. Vizedom, “Ontology Summit 2013 Communiqué: Towards Ontology Evaluation across the Life Cycle”, **(2013)**, Gaithersburg, MD.
[3] S. Tartir, I. Arpinar, M. Moore, A. Sheth and B. Aleman-Meza, “OntoQA: metric-based ontology quality analysis”, in: Proceedings of the Workshop on Knowledge Acquisition from Distributed, Autono-mous, Semantically Heterogeneous Data and Knowledge Sources (KADASH), Citeseer, **(2006)**.
[4] B. Kitchenham, S. Pfleeger and N. Fenton, “Towards a framework for software measurement validation”, IEEE Transaction on Software Engineering, vol. 21, no. 12, **(1995)**, pp. 929–944.
[5] L. C. Briand, S. Morasca and V. R. Basili, “property-based software engineering measurement”, IEEE Transactions on Software Engineering, vol. 22, no. 1, **(1996)**, pp. 68-85.
[6] A. Gangemi, C. Catenacci, M. Ciaramita and J. Lehmann, “A theoretical framework for ontology evaluation and validation”, In: Proceedings of Semantic Web Applications and Perspectives (SWAP2005), **(2005)**.
[7] A. Gangemi, C. Catenacci, M. Ciaramita and J. Lehmann, “Modelling ontology evaluation and validation”, in: Proceedings of the 3rd European Semantic Web Conference (ESWC2006), vol. 4011, **(2006)**, Springer.

[8] A. Duque-Ramos, J. T. Fernández-breis, M. Iniesta, M. Dumontier, M. E. Aranguren, S. Schulz, N. Aussenac-Gilles and R. Stevens, “Evaluation of the OQuaRE framework for ontology quality”, Expert Systems with Applications, vol. 40, no. 7, **(2013)**, pp. 2696–2703.

[9] A. Duque-Ramos, J. T. Fernández-Breis, R Stevens and N. Aussenac-Gilles, “OQuaRE: A SQuaRE-based Approach for Evaluating the Quality of Ontologies”, Journal of Research and Practice in Information Technology, vol. 43, no. 2, **(2011)**, pp. 159-176.

[10] A. Burton-Jones, V.C. Storey, V. Sugumaran and P. Ahluwalia, “A semiotic metrics suite for assessing the quality of ontologies”, Data and Knowledge Engineering, vol. 55, no. 1, **(2005)**, pp. 84–102.

[11] D. Zhang, C. Ye and Z. Yang, “An evaluation method for ontology complexity analysis in ontology evolution”, in: Proceedings of 15th International Conference, EKAW2006, Lecture Notes in Computer Science, Springer, vol. 4248, **(2006)**, pp. 214-221.

[12] D. Kang, B. Xu, J. Lu and C. Chu, “A Complexity Measure for Ontology Based on UML”, in: Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04), **(2004)**, pp. 222-228.

[13] F. Ensan and W. Du, “A semantic metrics suite for evaluating modular ontologies”, Information Systems, vol. 38, no. 5, **(2013)**, pp. 745-770.

[14] H. Stuckenschmidt and M. Klein, “Structure-based partitioning of large concept hierarchies”, The Semantic Web-ISWC 2004, **(2004)**, pp. 289–303.

[15] Y. Ma, B. Jin and Y. Feng, “Semantic oriented ontology cohesion metrics for ontology-based systems”, The Journal of Systems and Software, vol. 83, **(2010)**, pp. 143–152.

[16] A. Orme, H. Tao and L. Etzkorn, “Coupling metrics for ontology-based system”, IEEE Software, vol. 23, no. 2, **(2006)**, pp. 102–108.

[17] S. Oh and J. Ahm, “Ontology Module Metrics”, In Proceedings of the International Conference on E-Business Engineering, **(2009)**.

[18] T.R. Gruber, “A translation approach to portable ontology specifications”, Knowledge System Laboratory, KSL, **(1993)**, pp. 92-71.

[19] A. Maedche and S. Staab, “Ontology learning for the semantic Web”, IEEE Intelligent Systems, vol. 16, **(2001)**, pp. 72-79.

[20] H. Chae, Y. Kwon and D. Bae, “A cohesion measure for object-oriented classes”, Software Practice and Experience, vol. 30, no. 12, **(2000)**, pp. 1405-1431.

[21] S. Chidamber and C. Kemerer, “A metrics suite for object ori-ented design”, IEEE Transaction on Software Engineering, vol. 20, no. 6, **(1994)**, pp. 476–493.

[22] Swoogle, “Swoogle Semantic Web Search Engine”, http://swoogle.umbc.edu/index.php, **(2007)**.

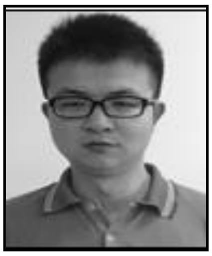
[23] Protégé, “Protégé ontology library”. http://protégé-wiki.stanford.edu/wiki/Protege_Ontology_Library, **(2012)**.

Authors

Lili Liao, she was born in 1989 and received the B.S. degree in Computer Science and Technology. Now she is a master candidate at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Jiangsu, China. Her research interests include semantic web, ontology.

Guohua Shen, he was born in 1976. He received his Ph.D. degree in Computer Science from Nanjing University of Aeronautics and Astronautics in 2009. Now he is an associate professor at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include domain engineering, software metrics, semantic web and description logic.

Zhiqiu Huang, he was born in 1965. He received his Ph.D. degree in Computer Science from Nanjing University of Aeronautics and Astronautics in 1999. Now he is a professor and Ph.D. supervisor at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. His research interests include software engineering, software metrics and semantic web.



Fei Wang, he was born in 1990 and received the B.S. degree in Computer Science and Technology. Now he is a master candidate at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Jiangsu, China. His research interests include semantic web, ontology.