A Lightweight Approach for Evaluating Sufficiency of Ontologies

Lalit Mohan S, Gollapudi VRJ Sai Prasad, Sridhar Chimalakonda, Y. Raghu Reddy and Venkatesh Choppella
Software Engineering Research Center, IIIT Hyderabad, Hyderabad, India

Abstract

[1]*Ontologies have emerged as a common way of representing knowledge. Recently, people with minimal domain background or ontology engineering are developing ontologies, leading to a corpus of informal and under-evaluated ontologies. Existing ontology evaluation approaches require rigorous application of formal methods and knowledge of domain experts that can be cumbersome or tedious. We propose a lightweight approach for evaluating sufficiency of ontologies based on Natural Language Processing techniques. The approach consists of verifying the extent of coverage of concepts and relationships of ontologies against words in domain corpus. As a case study, we applied our approach to evaluate sufficiency of ontology in two example domains - Education (Curriculum) and Security (Phishing). We show that our approach yields promising results, is less effort intensive and is comparable with existing evaluation methods.*

Keywords : Ontology; Ontology Evaluation; Sufficiency

1 Introduction

An ontology is essentially a shared understanding, a unifying framework, a world view of a domain of interest. Ontologies can be about any topic of interest, and as they can be readily merged and made into hybrid structures, it is quite possible that the ontologies can be large. Ontologies are considered significant and reusable as they contained core knowledge structures that require rigor for both development and evaluation. To keep rigor, multiple parameters are checked and detailed criteria is considered for evaluation of ontologies by various researchers [3], [9]. The criteria listed by Vrandevic [16] contains accuracy, adaptability, clarity, completeness, computational efficiency, conciseness, consistency, and other parameters for evaluation. The emergence of semantic web has triggered a need to connect a multitude of web applications from various domains, share and exchange knowledge between them. Several ontology repositories like Protege Ontology Library[2], Linked open vocabularies[3] and search engines like Swoogle[4] and OntoSearch[5] have emerged as a way to access these ontologies. From a utility point of view, software engineers have been using them in their applications for structuring knowledge, sharing a common understanding, explicitly surfacing a given perspective, enabling interaction, navigation, etc. This extensive growth in the use of ontologies poses a critical need to evaluate the quality of ontologies.

Today, informal, loosely defined ontologies have become quite prolific. One of the plausible reason being that ontologies are developed by people with minimal background in ontological engineering, thus making it important to assess the completeness of such ontologies. Completeness is defined as 'all that is supposed to be in the ontology is explicitly stated in it, or can be inferred' [3]. Completeness [16] can be measured from various perspectives: with regards to the language, domain, applications requirements, etc. We are interested in domain and application requirements as it applies to both the goals of the software developer as well as its coverage of the domain the ontology is representing. For an ontology to be complete for a domain, it is necessary for it to represent adequate portion of the domain. However, domain completeness of an ontology cannot be checked as only some of the real world knowledge is available or aspects in real world change over a period of time. We measure completeness as the degree of coverage of real world situations available in the form of web documents. Adopting real world coverage measure for completeness, we introduce **Sufficiency** as a means to measure completeness of the loose and informal ontologies. Our definition of Sufficiency is 'the adequate coverage of specific ontology concepts and relationships for a domain corpus'. The domain corpus would be considered as adequate if the newness of obtained / extracted words tapers. For simple and small domain ontologies, the mechanisms for evaluation, especially for evaluating completeness seems to be under-represented. In our research, we are interested in the problem of evaluating sufficiency of weak, loosely defined domain ontologies.

[1]DOI: 10.18293/SEKE2017-185

[2]http://protegewiki.stanford.edu
[3]https://datahub.io/dataset
[4]http://swoogle.umbc.edu/
[5]http://www.ontosearch.com/

2. Literature Survey

Broadly, the approaches for Ontology evaluation can be classified as (i) manual, mainly driven by human interventions, either experts or users (ii) automated approaches and (iii) semi-automated approaches that fall in between. One way of classification uses black box strategies, which is primarily used from end user perspective or when ontologies are not available during construction, grey box strategies are applied throughout the life cycle of ontologies [4]. A classification by Brank et al. [1] is based on two dimensions (i) type of approach (comparison against a gold standard, application or task-based evaluation, user based evaluation, and data-driven evaluation) and (ii) level of evaluation (lexical, vocabulary, or data layer; hierarchy, taxonomy; other semantic relations; context, application; structure, architecture, design). Ren et al. [11] suggested axiomatic and formalization of competency questions for ontology evaluation. Hlomani and Stacey [6] defined ontology evaluation as verification and validation. However, modeling ontologies using first order logic and formal techniques are daunting tasks that might not be feasible in the case of simple ontologies, which is the focus of this paper. While OntoClean's approach to use formal notions from philosophy such as essence, rigidity, identity and unity for ontology correctness might not be directly relevant for our case, they emphasize the need for validation of ontological adequacy [5].
There are several lines of research that focused on metrics for ontology evaluation. For example, EvaLexon [14], assessed triples mined for text and calculated precision, accuracy and recall values for a domain. The approach had 95% confidence level for 60% coverage. Samir et al. [15] in OntoQA used schema metrics and instance metrics to evaluate ontologies and knowledge bases. They state that "goodness" or the "validity" of an ontology vary between different users or different domains, making it subjective. Astrid et al. [2] extended software product quality SQuaRE, ISO/IEC 25000:2005 to establish OQuaRE framework. The evaluation includes structural, Functional adequacy, Reliability, Performance efficiency, Operability,etc. Gomez et al. [7] proposed OntoMetric with 129 characteristics across 5 dimensions (Tools, Language, Content, Methodology and Costs) for evaluating ontology. Sabou et al. [12] states ontology evaluation is core to ontology selection and have a well laid process for evaluating large scale web based applications. Most of these metric based approaches require extensive information on specific properties of the ontologies that are generally not available for simple ontologies. The ontology coverage check method proposed by Pammer et al. [10] starts with basic domain terms coverage and extends to axioms but their method is focused on individuals with a validity threat that individuals for ontologies are generally not available. Noy et al. [8] suggested using ontology search criteria of the user for evaluating the completeness of ontology. We see that search criteria is an important aspect of evaluation, which we also use in our method but based on domain than on users or specific contexts.

3. Proposed Method

Our intention is to evaluate the sufficiency of a given onotolgy, which has been developed for a particular purpose, against a given domain. This requires us to, identify the test corpus from the domain which is adequate for our evaluation and check for the coverage of ontology in the selected test corpus of the domain.

3.1. Collecting Sufficient Test Corpus

We wish to identify the test corpus of the domain that should be used in our completeness evaluation. The choice of which specific document to consider as corpus is related to the purpose/goal of the ontology. We make an assumption that both the goal and the access to real-to-life test corpus is available. Based on this, we suggest that the type of corpus and the search strings for obtaining corpus should be driven by goals, set for the ontology. The quantity of the test corpus that needs to be considered can indeed vary. To contain this, we bring in the notion of adequate domain corpus. The process for collecting adequate quantity of test corpus is : we select a document, search for unique words in it and count them. If the next document contains more than $Suf\%$ of new words, we add them to the list of unique words and then continue with next document, else, we stop the process. We believe that, after some point, the corpus of words stops being significantly unique. We cut off at $Suf\%$ difference, an arbitrary number and can be changed. The trade-off is that the smaller this number, the more test documents are needed to feed the system. The result of this step is to conclude on the quantity of documents (*SDC* - Sufficient Coverage) that is sufficient for checking our coverage.

3.2. Checking for Coverage

The *SDC* provides sufficient corpus for evaluating the coverage of a specimen ontology. Each individual document of the domain corpus within the set of *SDC* is used for evaluating ontology Concepts, Concepts + Relationships, and Concepts + relationships + Concepts coverage. Individual Concepts label or a Relationship label are represented by C. Concepts + Relationships by R, this R is more restrictive than C because it defines a Concept and potential Relationships of the Concept. For Tuple, Concept + Relationship + Concept is represented as T, this is most descriptive as it contains various destination concepts.

3.2.1 Step 1: Identify Test Ontologies

For our evaluation purposes, we may either have a test ontology or we may need to get one from a ontology repository. For obtaining a preexisting one, we suggest selecting our ontology from a set of at least 3 possible alternatives. The reason we suggest 3 is that if for a domain if we have less than 3, then there is no question of selection and evaluation. The choice is self evident.

3.2.2 Step 2: Extract Labels

A list of concepts and relationships of an ontology give a rough idea of the overall scope and capability of that specimen ontology. Extract and create a list of concept and relationship (C, R, and T) for each of the specimen ontology from a OBO-XML, OWL RDF/XML format. For small, weak, loosely defined ontologies, number of nodes and edges are not expected to be high in number, so we expect this process can be automatic or manual and simple. At the end of this step, there should be three structures for each specimen ontology: One, a list containing clusters of words for each ontology.

$$\{O_1, O_2, ...O_n\}$$

This is C and it should contain all the labels (Concepts and Relationships) in the ontology. Two, A two dimensional array highlighting Concept + Relationships for each ontology.

$$\{O\{c\}\{r\}_1, O\{c\}\{r\}_2, ...O\{c\}\{r\}_n\}$$

Three, L is same as Two but with consideration of Lemmas (grouping of different inflected words such as teach for teaches, taught, etc.) in the corpus. Four, a three dimensional array for each ontology showcasing the Tuples present in the ontology.

$$\{O\{c\}\{r\}\{c\}_1, O\{c\}\{r\}\{c\}_2, ...O\{c\}\{r\}\{c\}_n\}$$

3.2.3 Step 3: Obtain Synonyms

In our work, the implication of using words is that we may be only looking for exact string matches and not consider either common concepts or related words as same. So, if a 'student' is the search word, then it will not match with either a 'pupil' or a 'participant'. There are existing Natural Language Processing (NLP) techniques including Hypernyms, Synonyms to cluster similar words together. On comparison with Synonyms, the coverage boosted between ontologies (Concept and Relationship words) and sample web document sizes. This suggests that a Synonym will effect all data similarly and will not enhance one ontology over the other. Due to this common impact, we rejected usage of Synonyms or other similarity techniques.

3.2.4 Step 4: Identify Test Corpus

Each of the document identified as part of *SDC* and also the aggregation of the text in the documents is part of the test corpus. If the test corpus is hard to identify or define, then chances are that the ontology is no more simple or loosely defined. For such cases, the process needs to be more rigorous and systematic as detailed in literature survey.

3.2.5 Step 5: Pre-process

After selecting test corpus, the content needs to be preprocessed. Pre-processing involves (i) Extract text - in some cases this could be grabbing text from websites or from PDF documents (ii) Ensure that appropriate text in images, tables, audio/video (subtitles) is accessible for extraction while removing non-textual elements (iii) Perform Part-of-Speech (POS) tagging to list Nouns and Verbs in the document (iv) Remove unwanted repeat / stop words like 'and', 'but', 'if' etc. These can be considered as prepositions, conjunctions and other common English words that may not be relevant to domain ontology. The intent of these steps is to ensure that the test corpus is machine ready for evaluation.

3.2.6 Step 6: Collect Unique Words

The aggregate text file content from pre-processing step is processed for extracting the list of most frequently used words. Any text analysis or information retrieval library or online tool that serves this purpose can be used for gathering a bag of words.

3.2.7 Step 7: Compare

This is done by comparing the list of C, R, L and T words of the specimen ontology with all the list of unique words extracted from each document of *SDC* set to determine sufficiency. For comparison, we are attempting to string match the label (of a Concept or a Relationship) and also match the lemmas of the text. The reason for inclusion of lemmas is because a test corpus is always more grounded in instances, whereas a ontology is typically more abstract and at a higher level. As we intend to keep the matching algorithm lightweight, we are not proposing rigorous NLP techniques such as identification of Hypernym, Hyponym, Bi-grams, etc. or any similarity algorithms such as Latent Semantic Analysis and Word2Vec (trained on 100 Billion words). In our work, to check for degree to which an ith ontology is sufficient, we apply this equation

$$(O_i MWFC/OWFC) = O_i \text{Sufficiency (SC)} \qquad (1)$$

Where O_i represents the ith ontology, $MWFC$ represents the frequency count of the matching words, and $OWFC$ represents the total word frequency count of the ontology for

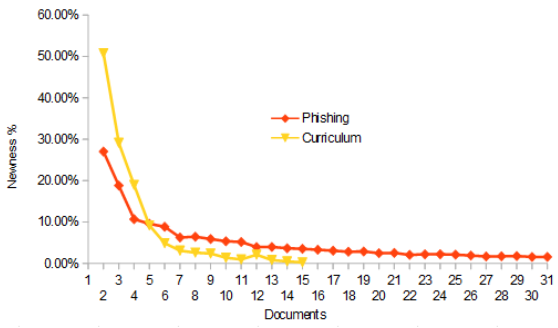

Figure 1. Sufficient Domain Coverage

a given corpus. To check and evaluate something as sufficient, we first take the words from the ontology and string search them within the corpus for C, R, L and T. Once the ontology related words are found in the corpus, then they are identified and their frequency counts are obtained. The total frequency count of the matched words is checked against the overall frequency count of all words. This ratio is said to be the measure of Sufficiency SC.

4. Evaluation of the Approach

We now evaluate our approach by applying it to the various publicly available Curriculum and Phishing ontologies. Our goal is to select an ontology with the highest degree of Sufficiency for a hypothetical project.

4.1. Selection of Test Corpus

To gather test corpus, we used popular search engines like Google, Bing, Yahoo, Yandex and Baidu with search string as 'Curriculum' and 'Phishing'. From the search results, a union collection of documents are gathered as our test corpus [13]. The text in each of the documents is concatenated into a single document and the newness of words is identified. The Sufficient Domain Coverage figure 1 shows that the newness tapers after 12 documents for Curriculum and after 25 documents for Phishing. The site specific words such as Contact address, Organization name, creative English by writers, etc. are probable reasons for the newness value not being zero.

4.2. Pre-processing of Corpus

Pre-processing involves conversion of PDF to Text mode for some cases, scrapping content (removal of html tags, css, images, etc.), removal of stop words, etc. Removal of stop words and word frequency count was done by a Java application[6] that we developed. Along with this, we also performed POS tagging to identify the list of Nouns and Verbs using Stanford NLTK. The count [13] of stop words, unique words and the newness for Curriculum and Phishing Ontology is obtained.

4.3. Handling of Ontologies

A web search on the word Curriculum and Phishing was used to identify three Ontologies including Ontosearch and Swoogle. After earmarking the specimen ontologies, the next task of extracting labels from each of ontologies separately was done.

4.4. Word Comparison

After extracting labels from ontologies, each of those labels were string searched for potential matches in the words list of the test corpus. Whenever there was a match, the matching frequency was obtained and aggregated. This gave us O_i MWFC or the matching word frequency count of the ith ontology.

4.5. Checking for Completeness

For the ontology, the last task is to calculate the sufficiency as degree of a completeness score. This score is calculated (as given by equation 1) by dividing O_i MWFC by the OWFC value. See in [13], D1 through D12 indicates documents identified as part of the corpus and C12 indicates the combination of all the documents. Sample Ontologies for Curriculum (O_1^c, O_2^c and O_3^c) and Phishing (O_1^p, O_2^p and O_3^p) are used for determining ISC - Individual Sufficient Completeness (concepts and relationships). An average of ISC is considered for arriving at Sufficient Completeness SC, however, any other statistical approach can be considered for the calculation.

4.6. Results and Discussions

We tested 12 and 25 sample sets of corpus against 6 (3 of Curriculum and 3 of Phishing) ontologies. As observable in [13], unique words constitute 35-40% of overall word count with some words related to domain being more prevalent [7]. An ontology can be said to be sufficiently complete if, after matching the goals of the ontology, C, R and L extracted from the ontology fully encompass the words of the corpus.

- For Concepts or Relationships C of Curriculum is 64.06% for O_1^c and 61.25% for O_2^c. For Phishing, the score is 67.13% for O_2^p and 51.28% for O_1^p.

[6] https://github.com/lalitsanagavarapu/OntologyEval

[7] http://tinyurl.com/UniqWord

- For Concepts and Relationships R of Curriculum is 60.26% for O_1^c and 24.24% for O_2^c. For Phishing, the scores is 51.75% for O_2^p and 40.17% for O_1^p. The concepts and relationships are compared as Nouns and Verbs after POS tagging the web documents.
- For Concepts and Relationships with application of Lemma L on Curriculum, the score is 73.72% for O_1^c and 70.45% for O_2^c. For Phishing, the score is 59.79% for O_2^p and 45.30% for O_1^p.
- From results, Ontology O_1^c for Curriculum and O_2^p for Phishing stand out as better suited for our application.

Like the check for completeness, coverage of corpus in an ontology too appears to be an audacious goal. Hence, we reject our hypothesis that

$$C \subset O_i$$

. In our evaluation, we considered $Suf\%$ of 2% as the cut-off percentage for newness with 12 documents of Curriculum and 4% (12 documents) for Phishing. The test corpus selection is subjective step in our proposed approach. Hence, we performed the test to see if the concepts and relationships of any one of the earmarked ontologies are present in the R_8 random (include text from the novel Pride & Prejudice; Wiki content on Auto, Health, Sport, Finance, Food, Travel; and a magazine article on 'top technological trends') corpus. The lower Sufficiency Coverage value in the results [13] indicates that most of the ontologies are poorly represented in the random corpus. The combined random corpus content with its 186,360 words and with its 9,801 unique words did not gain much in completeness. The lower numbers for our random sample also indirectly reinforces the other point that the sufficiency value of O_1^c for Curriculum and O_2^p for Phishing is not accidental, but indeed intentional and specific to the ontology. Our approach that is automatic, informal using web documents as domain data as compared to various evaluation approaches. Our approach is similar to OntoQA [15] but has lesser steps and lesser metrics to evaluate ontology.

5. Conclusions and Future Work

There are many techniques to evaluate ontology and most of them appear to be rigorous and tend to target the evaluation of well defined and large ontologies. In such context, we sought and evaluated a lightweight approach for checking sufficiency of smaller ontologies. The approach is simple as it relies on concept and relationship matching and conventional web search techniques. Our evaluation explored veracity of the approach and established the feasibility on two different domains and could extend for other weak and loosely defined ontologies. As a forward plan, we plan to make a tool online instead of running it as a batch process so that other users can leverage it. We also plan to use the domain knowledge available in the web documents including text cohesiveness to evolve ontologies based on identifiable patterns.

References

[1] J. Brank, M. Grobelnik, and D. Mladenic. A survey of ontology evaluation techniques. In *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*, pages 166–170, 2005.

[2] 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*, 40(7):2696–2703, 2013.

[3] A. Gómez-Pérez. Ontology evaluation. In *Handbook on ontologies*, pages 251–273. Springer, 2004.

[4] G. Grigonyte. *Building and evaluating domain ontologies: NLP contributions*. Logos Verlag Berlin GmbH, 2010.

[5] N. Guarino and C. A. Welty. An overview of ontoclean. In *Handbook on ontologies*, pages 201–220. Springer, 2009.

[6] H. Hlomani and D. Stacey. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, pages 1–5, 2014.

[7] A. Lozano-Tello and A. Gómez-Pérez. Ontometric: A method to choose the appropriate ontology. *Journal of database management*, 2(15):1–18, 2004.

[8] N. F. Noy, P. R. Alexander, R. Harpaz, P. L. Whetzel, R. W. Fergerson, and M. A. Musen. Getting lucky in ontology search: a data-driven evaluation framework for ontology ranking. In *International Semantic Web Conference*, pages 444–459. Springer, 2013.

[9] L. Obrst, W. Ceusters, I. Mani, S. Ray, and B. Smith. The evaluation of ontologies. In *Semantic web*, pages 139–158. Springer, 2007.

[10] V. Pammer, P. Scheir, and S. Lindstaedt. Ontology coverage check: support for evaluation in ontology engineering. In *FOMI 2006. The 2nd workshop: Formal Ontologies Meet Industry*, 2006.

[11] Y. Ren, A. Parvizi, C. Mellish, J. Z. Pan, K. Van Deemter, and R. Stevens. Towards competency question-driven ontology authoring. In *European Semantic Web Conference*, pages 752–767. Springer, 2014.

[12] M. Sabou, V. Lopez, E. Motta, and V. Uren. Ontology selection: Ontology evaluation on the real semantic web. 2006.

[13] L. Sanagavarapu. Ontology - corpus comparison. `http://tinyurl.com/OntoLightWeight`, 2017.

[14] P. Spyns. Evalexon: Assessing triples mined from texts. *STAR*, 2005(09):09, 2005.

[15] S. Tartir, I. B. Arpinar, and A. P. Sheth. Ontological evaluation and validation. In *Theory and applications of ontology: Computer applications*, pages 115–130. Springer, 2010.

[16] D. Vrandečić. Ontology evaluation. In *Handbook on Ontologies*, pages 293–313. Springer, 2009.