Evaluating Ontology Modules Using an Entropy Inspired Metric

Paul Doran Valentina Tamma Ignazio Palmisano Terry Payne
University of Liverpool, UK
P.Doran, V.Tamma, I.Palmisano, T.R.Payne@liverpool.ac.uk

Luigi Iannone
University of Manchester, UK
iannone@cs.man.ac.uk

Abstract

The focus of ontology modularization to date has largely been on the creation of techniques to carry out ontology modularization. This creates a problem in evaluating the results of the different techniques. Ontology modularization techniques cannot solely be evaluated by examining their logical properties. Certain applications of ontology modularization, such as ontology reuse, require a new objective way to evaluate the results. This paper motivates the use of an entropy inspired measure to evaluate ontology modules by arguing that current objective measures of evaluation do not reconcile with the subjective measures employed by Ontology Engineers. Experiments are conducted to show that an entropy based evaluation of ontology modules is beneficial to an Ontology Engineer evaluating the results of ontology module extraction techniques.

1 Introduction

Ontology reuse is one of the fundamental cornerstones of the Semantic Web: knowledge sharing can only be facilitated if the shared resources commit to some compatible underlying ontological models, and the reuse of existing ontologies facilitates the convergence on these models. As more organisations are publishing their ontologies on the Web, and more mechanisms for discovering them [7] become available, ontology reuse has become increasingly desirable and feasible. Ontology engineering methodologies often include a reuse step in their development process; however they don't prescribe how reuse should be achieved [11]. For OWL[1] ontologies, the `owl:imports` statement allows reference within the ontology being created to the definitions of another ontology identified by its URI. However, this mechanism includes all the definitions of the imported ontology, and this might result in the inclusion of several definitions that are irrelevant to new ontology, thus making the computation of the inferred model more cumbersome (as additional inferences may be needlessly computed due to these redundant definitions). Therefore novel mechanisms are needed that permit knowledge engineers to identify, according to some criteria, the part of an ontology they plan to reuse.

Research into the problem of modularizing ontologies (including ontology module extraction and ontology partitioning) has largely focussed on the modularization process itself [8, 4, 6, 14, 19, 20], and on identifying conditions for including or excluding elements of an ontology module. However, there is little work on *evaluating* the modules themselves, and currently there are few metrics that objectively assess the quality and utility of an ontology module, thus making a comparative analysis of the modules difficult.

Most of the approaches in the literature consider the notion of *size* of the module, based on the number of concepts contained, as a factor to evaluate modularization techniques [4]. However, such a metric fails to consider the properties and restrictions relating concepts, and thus is orthogonal to the underlying semantics. Whilst size can be used as a metric to discriminate between modules on the basis of the concepts they include, it is too vague, and there is no precise way of determining what qualifies as a small module, or whether one size is better than another. A possible alternative would be to discriminate between the relevant elements of an ontology (classes, and properties) with respect to the modularization criterion. The *precision* and *recall* metrics have been widely used by the Information Retrieval community when evaluating the results of queries that include as many relevant documents as possible whilst minimising the irrelevant documents found. Doran *et al.* [8] have adapted these metrics to evaluate their modularization technique; however they found that whilst these metrics may have been more suitable for capturing the content of a module, they considered only the hierarchy of the ontology. Thus, the full structure of the ontology, including the semantics entailed by the properties and restrictions in the concept definitions were not considered.

This paper proposes a metric for evaluating ontology modules based on the notion of *entropy*[3], which can exploit the complete definition of concepts (including proper-

[1]Here by OWL representation of an ontology we refer primarily to the T-Box of a DL theory, as it is the part containing the concept definitions to be shared.

ties and restrictions) within the ontological module to determine its information content. The use of entropy was previously investigated by Calemt & Daemi [2] as a means of estimating "...the amount of information that some concepts contribute to a specific target concept..." The entropy calculation assumed a graph-based representation of the ontology[2], but considered all edges (that represent relationships between concepts; i.e. the nodes in the graph) as equal, and thus disregarded the different semantics that could be associated with an edge, and the direction of these relationships.

In this paper we therefore propose a reformulation of the entropy metric to evaluate the amount of information carried by both the ontology structure, and also by the language elements (i.e. the semantics associated with the edges in the ontological graph). To evaluate this approach, the reformulated metric is empirically compared to Calemt & Daemi's original entropy metric, for a variety of different sized modules. The results suggest that not only can entropy differentiate between structurally different modules of the same size, but that our improved entropy metric provides a finer grain differentiation than the original entropy metric.

We introduce the use of entropy and the entropy based metrics for ontologies in Section 2, before then describing the reformulation (in terms of domain and language level entropy) in Section 3. The empirical analysis of the new metrics is described in Section 4. Related work, regarding both ontology modularization and ontology evaluation, is then discussed in Section 5, before providing future directions and concluding in Section 6.

2 Entropy Based Measures

The notion of entropy has been applied to information theory [3] in order to provide a quantitative measure of the information contained in a message. Shannon defines entropy as a measure of the average information content the recipient is missing when they do not know the value of a random variable, that is the measure of uncertainty associated with the random variable, calculated as:

$$H(X) = -\sum_{i}^{n} p(x_i) \log p(x_i)$$

where $p(x_i) = Pr(X = x_i)$ and X is a discrete random variable. Calmet & Daemi exploited this notion for measuring the reduction of uncertainty of one concept with respect to another, by considering possible target concepts in between them [2]. This was represented through a *probability mass function*, $p(x_i)$, which was calculated for each vertex in the graph (corresponding to some concept), by dividing the degree of the vertex; i.e. number of edges (i.e. properties) connected to that concept, with the sum of all degrees of V (where $v_i, v \in V$ are vertices):

$$p(v_i) = \frac{deg(v_i)}{\sum_{v \in V} deg(v)}$$

However, this entropy-based approach is limited as it considers all edges as equal. For instance, a relationship between two concepts (represented by the OWL statement `<owl:ObjectProperty>`) is treated in the same way as an equivalence between two concepts (represented in OWL by `<owl:equivalentClass>`), even if these two notions carry very different meanings, and have different consequences with respect to a modularization technique: equivalent concepts should always be grouped together in a module, whilst this is not necessarily the case with object properties. Figure 1 illustrates the case where both graphs have an entropy value of 2.81, however, if we hypothesise that in one instance all the edges are `<owl:equivalentClass>` and in the other they are `<owl:ObjectProperty>`[3] then the entropy values should be different.[4] Indeed, when these edges represent `<owl:equivalentClass>`, then graph A represents the uninferred model and graph B the inferred model (i.e. the extra edges in graph B are implicit in graph A), where these edges have been made explicit and have an effect on the entropy measure. Analogously, when each edge represents a different `<owl:ObjectProperty>` then the entropy values should be different because the second graph has more properties linking the concepts.

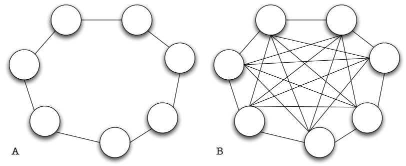

Figure 1. Graphs with equal entropy.

To overcome the limitations of this measure for evaluating ontologies, and adapt it to evaluate the result of modularization techniques, we propose a reformulation of the entropy measure of Calmet & Daemi that accounts for the different types of relationships that can exist between concepts. This reformulation separates the notion of *language level entropy*, from *domain level entropy*. *Language level entropy* estimates the information content carried by the edges that represent language level constructs.

[2]The use of graph-based representations of ontologies has frequently been used for ontology evaluation [9], and the transformation of an OWL ontology into a graph has been defined (`http://www.w3.org/TR/owl-semantics/mapping.html`).

[3]Assuming the vertices at either end of the edge are the domain and range.

[4]It is unlikely that a real ontology would contain such a pattern, but it is useful here to highlight a limitation of the measure.

These constructs are part of the ontological representation that is being used, for instance the OWL statements `<owl:equivalentClass>` or `<owl:subClassOf>` are language level constructs. The notion of *domain level entropy* is concerned with the domain specific relationships; these are the constructs that allow an Ontology Engineer to tailor the ontology to their domain. Such a construct in OWL would be the definition of an object property through the `<owl:ObjectProperty>` statement. Domain level entropy captures the information content that a relationship contributes to an ontology or to a module. The next section details how the entropy measure can be split.

3 Splitting Entropy

The entropy measure we define in this paper is calculated by considering the graph representation of the ontology. This model is an edge-labelled directed multigraph $G = (V, E)$ where: V is a finite set of vertices, representing the concepts defined in the ontology. $E = L \cup D$, where:
$L \subseteq V \times \Sigma_L \times V$ is a ternary relation whose elements (v_m, l_i, v_n) are language level edges, where $l \in \Sigma_L$, and Σ_L is the set of all the constructs in the ontology language that represent relationships between concepts. In OWL, Σ_L is equivalent to the set of properties in the OWL vocabulary[5]; for example, `<rdfs:subClassOf>`, or `<owl:complementOf>` are elements of Σ_L.
$D \subseteq V \times \Sigma_D \times V$ is a ternary relation whose elements (v_t, d_j, v_u), where $d \in \Sigma_D$, and Σ_D is the set of relationships defined to capture links between domain entities.
$\Sigma_L = \{l_1, ..., l_n\}$ and $\Sigma_D = \{d_1, ..., d_n\}$ are sets of labels which will label the edges of L and D respectively. Whilst the labels in Σ_L are defined by the specification of the ontology language used to represent the ontology, the labels in Σ_D are decided by the ontology developer. The following functions assign a label to each edge from the respective alphabets:$label_l(L) : L \rightarrow \Sigma_L$, and $label_d(D) : D \rightarrow \Sigma_D$.

3.1 Language Level Entropy - $H_L(X)$

The language level entropy ($H_L(X)$) calculates the entropy associated with the language level edges. We consider $G_L = (V, L)$ where $G_L \subseteq G$. We assume that all language level edges have equal weight and thus the probability mass function $p(v_i)$ is defined as:

$$p(v_i) = \frac{degOut(v_i)}{|L|}$$

where $degOut() = V \rightarrow \mathbb{R}$ for each v that exists in V such that $degOut(v) = |L_v|$ where $L_v = \{(v, l, x)|v \in V\}$.
The function $degOut(i)$ counts the number of outgoing edges from a given v, *i.e.* the *degree* of the node (concept) v. Thus $\sum_{v \in V} degOut(V) = |L|$ because for every element of V, the outgoing edges are considered and all the elements of L (v_i, l, v_m) must have $v \in V$ as the first element.

3.2 Domain Level Entropy - $H_D(X)$

The domain level entropy ($H_D(X)$) calculates the entropy associated with the domain level edges. We consider $G_D = (V, D)$ where $G_D \subseteq G$. We assume that the elements of Σ_D that appear more frequently in D split their information content evenly, thus the weight associated with these edges should be lower. For example, in an ontology modelling the relationships between a `PhD Student` and their `Supervisors` the relationships `coAuthorOf` can link a PhD student and a supervisor, or two PhD students, thus appearing more than once. Therefore, the information carried by this relationship is split between the contribution to the definition of `PhD Student`, and the contribution to the class `Supervisor`.

For every $d \in D$ we define a weighting function $w() = \Sigma_D \rightarrow \mathbb{R}$ that assigns a real number corresponding to the weight to every element of the alphabet Σ_D. The weights $w_d, d \in \Sigma_D$ are defined as $w(d) = \frac{1}{|D_d|}$ where $D_d = \{(x \times \sigma_D \times y)|label_d(d) = \sigma_D\}$; that is, the weights are determined on the number of elements of the relationship $D \subseteq V \times \Sigma_D \times V$ for which the label $sigma_D \in \Sigma_D$ is the same. The weights of the edges are normalised between 0 and 1, with the edges that appear more frequently getting a lower weight and the edges that appear less frequently getting a higher weight. The probability mass function $p(i)$ that we use for calculating $H_D(X)$ is:

$$p(i) = \frac{weightsFromNode(i)}{\sum_{v \in V} weightsFromNode(v)}$$

where $weightsFromNode() = V \rightarrow \mathbb{R}$ for each v that exists in V such that $weightsFromNode(v) = \sum_{f \in F} w(f)$ where F is the set of edges from D involving v. Thus, the weights of the edges outgoing from v are summed and divided by the sum of the weights of the outgoing edges for all elements of V.

	OE		IE (LE + DE)	
	Graph A	Graph B	Graph A	Graph B
No Edge Label	2.81	2.81	2.81 (2.81+0)	2.81 (2.81+0)
Equivalent	2.81	2.81	2.81 (2.81+0)	2.59 (2.59+0)
ObjectProperty	2.81	2.81	2.81 (0+2.81)	2.59 (0+2.59)

Table 1. Values of Figure 1 entropy measures.

3.3 Recombining The Entropy Measure

The ontology entropy measure $H(X)$ is calculated as the sum of the language, and the domain entropies:

$$H(X) = H_L(X) + H_D(X)$$

[5]The complete list of properties is available in Appendix C of the OWL refrence document (`http://www.w3.org/TR/owl-ref/`)

Depending on the semantics encoded in the graph it may be necessary to consider $\top$ and $\bot$. Assuming that $\top$ and $\bot$ are elements of V then they are considered in the above formula. However, one may just wish to consider the entropy amongst the user declared elements of V, as $\top$ and $\bot$ are usually required elements of the language (e.g., OWL). In this case, the entropy measure for the ontology would be $H(X) = (H_L(X) + H_D(X)) - (H(\top) + H(\bot))$.

By applying the reformulation of the entropy measure (IE) to both graphs in Figure 1. The original entropy (OE) and the improved entropy (IE) were calculated for graphs A and B depicted in Figure 1 for three cases. These were when no edge was labeled, when the edges were given the `<owl:equivalentClass>` labels and lastly when the edges were given the `<owl:ObjectProperty>` label. The results are shown in Table 1. These results show that the improved entropy measure discriminates between the two graphs: the entropy now changes in graph B when the improved entropy measure is applied. As it is the sum of the domain entropy (DE) and language entropy (LE) it is possible to identify the contribution of different types of edges to the overall entropy value. Thus, whilst graph B for both `<owl:equivalentClass>` and `<owl:ObjectProperty>` have equal entropy, it is possible to assess which type of edge contributed to the entropy.

4 Empirical Evaluation

To evaluate whether the proposed entropy-based metric is more discriminating than other entropy metrics or the size-based metric, we have compared them over ontology modules produced by various ontology module extraction techniques. Four such techniques (proposed by Doran *et al*[8], d'Aquin *et al*[6] and the 'upper' and 'lower' variants proposed by Cuenca-Grau *et al*[12]) have been applied to three ontologies of varying expressivity; the Family($\mathcal{ALC}$), AKT-Portal($\mathcal{ALCHIOF(D)}$) and MindSwap($\mathcal{ALCHIF(D)}$) ontologies. In each case, a module was created for each of the concepts (i.e. the concept is used as a seed concept for each method) in the ontology resulting in a module set. For each of the twelve resulting module sets, we indicate the size (denoting the number of concepts in the module), the value obtained from applying the original entropy measure (*OE*) and the improved entropy measure (*IE*) proposed in this paper, as well as its two constituent entropy elements: the language level entropy (*LE*) and the domain level entropy (*DE*). The evaluation of these results is performed along two dimensions: an *intra-technique* evaluation (see Section 4.1) and an *inter-technique* evaluation (see Section 4.2). The intra-technique evaluation determines if the entropy based measures discriminate between modules of the same size when the signatures supplied to the algorithm were different. The inter-technique evaluation reflects the perspective of an Ontology Engineer wishing to reuse an ontology module; where there is a need to discriminate between two equally sized ontology modules produced by different techniques.

Family ontology					
	Interval of Module Size	**OE**	**LE**	**DE**	**IE**
Doran	18	0.157	0.249	0.208	0.431
d'Aquin	5	0.004	0.241	1	1.241
Cuenca Lower	26	0.126	0	0	0
AKT Portal ontology					
Doran	1	0.018	0	0	0
	34	0.005	0	0.013	0.012
	36	0.015	0.002	0.018	0.018
	37	0.012	0.003	0.010	0.013
	38	0.088	0.064	0.050	0.112
	39	0.012	0	0.024	0.025
	186	0.121	0.137	0.156	0.291
d'Aquin	40	0.162	0.680	2.030	1.475
	41	0.283	0.709	2.053	1.511
	42	0.168	0.629	2.075	1.509
	43	0.202	0.702	2.071	1.511
	46	0.187	0.644	2.05	1.501
	48	0.176	0.690	1.396	0.933
Cuenca Upper	20	0.370	0.741	1.437	1.128
	21	0.416	0.849	1.182	0.346
	22	0.450	0.762	1.536	1.397
	23	0.507	0.843	1.573	1.258
	24	0.473	0.698	1.600	1.212
	25	0.504	0.733	1.642	1.415
	27	0.450	0.857	1.506	1.084
	28	0.644	1.005	1.427	1.027
	29	0.487	0.690	1.450	1.201
	30	0.483	1.036	1.025	0.383
	40	0.394	0.892	0.464	0.428
	42	0.396	0.606	0.996	0.692
Cuenca Lower	9	0.299	0.322	0	0.322
	14	0.243	0.590	0.722	1.000
	15	0.412	0.965	0.906	1.004
	24	0.216	0.319	0.066	0.384
Mindswap ontology					
Doran	1	0.142	0	0	0
	17	0.004	0.011	0	0.011
	19	0.018	0.024	0.300	0.309
	20	0.067	0.024	0.291	0.291
	23	0.002	0.016	0	0.016
d'Aquin	1	0	0	0	0
Cuenca Upper	11	0	0	0	0
Cuenca Lower	1	0	0	0	0

Table 2. Intervals in the entropies for the Family, AKT Portal, and Mindswap ontologies

4.1 Intra-technique Evaluation

Each of the module sets extracted were grouped by size, and the entropy was calculated (each of the four metrics under evaluation were used). An *interval* was then determined for a set of some given size, based on the difference between the maximal and minimal entropy calculations for the modules in that set. These intervals are listed in Table 2 for each of the three ontologies using the four module extraction techniques. No results are shown for those sets where the interval value was zero for all entropy metrics evaluated.

For the Family ontology, the results show that in two cases the new entropy based metrics are more discriminating than the original entropy metric (OE), whilst in the case of Cuenca-Grau *et al*'s 'lower' technique, only the OE metric provided some discrimination (i.e. there was a difference of 0.126 between the highest and lowest entropy values for different modules of size 26). However, for many of the ontology modules produced (seven sets, which are not reported in the table), there was no difference in entropy values. This is due to the ontology being highly interconnected, it contains many concepts in terms of complex class restrictions (for example, $Grandfather \equiv Father \sqcap \exists hasChild.Parent$).

For most of the modules generated from the AKT-Portal ontology, the improved entropy metric (IE) provided greater discrimination than the OE metric. This difference varied, depending on the module extraction technique; from an average of 0.039 (OE) compared to 0.067 (IE) for the Doran technique, to an average of 0.196 (OE) compared to an average of 1.407 (IE) for d'Aquin's technique. The OE metric identified the smallest intervals for the majority of module sets, and in general, the Domain-Level entropy metric produced the greatest intervals.

The MindSwap ontology produced some anomalous results, with two of the four techniques (d'Aquin *et al* and Cuenca-Grau *et al*'s 'lower' technique)[6] producing modules of size 0[7] or 1. As the entropy metrics rely on there being edges between concepts, they fail on graphs with single (or a very small number of) concepts. Doran *et al*'s technique produced several modules with a range of sizes. However, unlike the intervals generated for the other ontologies, a greater number of sets had intervals at the language level rather than at the domain level, suggesting that the modules tend to contain the same domain level edges.

LCO	d'Aquin	Doran		
Import Directives	none	support.owl		
Expressivity	$\mathcal{ALCF}$	$\mathcal{ALCOF(D)}$		
	Defined	**Defined**	**Imported**	**Total**
Classes	22	9	17	26
Datatype Properties	0	4	10	14
Object Properties	10	1	5	6
Annotation Properties	1	1	3	4
Individuals	6	2	13	15
GCIs	5	0	0	0
SubClass Axioms	18	23	0	23
Disjoint Axioms	0	7	0	7
Base Model Triple No.	359	64	284	348
Inferred Model Triple No.	572	120	575	692

Table 3. Comparison between d'Aquin and Doran approaches on *LCO*

[6]These results suggest that the extraction techniques of d'Aquin *et al* and Cuenca-Grau *et al* place strict criteria in certain circumstances on what is included within a module, and thus may fail to produce usable modules.

[7]An ontology module of size zero would typically contain either $\top$ or $\bot$. This may be of value when considering the modularization process itself, but of little pragmatic use to the Ontology Engineer.

Approach	OE	LE	DE	IE
Doran	4.048	4.963	3.864	8.826
d'Aquin	4.975	4.655	3.936	8.591

Table 4. Entropy values for LCO modules.

4.2 Inter-technique Evaluation

Two modules extracted from the Portal with the signature set to 'Learning Centered Organization' (LCO) by the d'Aquin and Doran approaches are described by means of some metrics computed with SWOOP[8] in Table 3. These results show that the two approaches generate similar modules w.r.t. size, but their entropy values are different (see Table 4) and indeed their content is largely different. This shows that two modules of the same size extracted by different techniques that produce an ontology module about the same concept are better discriminated objectively via an entropy based measure; and that the improved measure allows an Ontology Engineer to better identify where the difference is. The following now examines some of the differences between the two modules.

One important difference is the fact that the Doran approach leaves the *owl:imports* directives in the module. This helps to keep track of dependencies, as well as allowing the extracted module to be reused should the imported ontologies change, but importing large ontologies may lead to very large modules. The d'Aquin approach would require the ontology module to be rebuilt if any change occurs in the imported ontologies, but the module is self contained.

Another difference is the expressivity: d'Aquin does not include datatypes and nominals in the module. However, the relation between modularization methods and expressivity needs deeper investigation before meaningful conclusions can be drawn. Looking at the specific differences, we note that the only common named concept between the two modules, not including the imported ontology, is the root of the model: LCO. Focusing on the differences, in the d'Aquin module 13 subclass relationships where LCO is the subject were included, one with *Organization* and 12 with anonymous classes, which represent the definition of LCO. In the Doran module, there is only one *isDefinedBy* property that states that LCO is defined according to `http://www.aktors.org/ontology/portal`, and 7 statements relating LCO to its named subclasses. d'Aquin seems to capture the definition of the root concept, while Doran aims at to capture the portion of the ontology that specialises the root concept; this is confirmed by the respective motives outlined in[8, 6]

5 Related Work

There are two relevant areas of the literature that contribute to this work: ontology modularization and ontology evaluation. Both are reviewed in this section.

[8]`http://code.google.com/p/swoop/`

The literature on ontology modularization can be split into two categories: ontology partitioning and ontology module extraction. Ontology partitioning [5, 20] divides an input ontology into a number of, not necessarily disjoint, partitions. Ontology module extraction techniques[4, 6, 8, 15, 19] take an input ontology and extract an ontology module about a supplied signature. The ontology module extraction literature can be further subdivided into two distinct categories: *traversal approaches* and *logical approaches*. The traversal approaches [8, 6, 19, 15] solve the ontology module extraction problem via a graph traversal. The logical approaches focus [4]on maintaining logical properties.

Doran, Tamma and Iannone[8] focus on extracting an ontology module that describes a single concept, supplied by the user, for the purpose of ontology reuse. d'Aquin, Sabou and Motta[6] present an extraction process which forms part of a knowledge selection process that constraints the size of the module produced. Seidenberg and Rector[19] take one or more classes of the ontology as input, and anything that participates (even indirectly) to the description of an included class has to be included. Noy and Musen's[15] approach is based upon the notion of *traversal view extraction*. Starting from one class of the considered ontology, the approach traverses the relations of this class recursively to include related entities; it is intended as an interactive tool.

Cuenca-Grau *et al* [4] define a module as a minimal, conservative extension [13] of the original ontology with respect to the considered sub-vocabulary. In [4] a module is that if the new axioms added to the original ontology are a conservative extension[13] of the original signature, then the original ontology (not the extension) is said to be a module of all the axioms together. Cuenca-Grau *et al* [4] also show that computing a minimal module, with respect to this definition, is undecidable. [4] describes two different approximations based on *locality*. The first method makes use of a reasoner to check the semantic locality of the axioms, this is decidable. The second syntactically tests the locality of the axioms and is in polynomial time.

Ontology evaluation has been recognised as important from the infancy of Ontology Engineering, and most methodologies include an explicit evaluation stage [11]. Despite this, a number of studies have addressed the challenge of evaluating ontologies from several perspectives.

Surveys of evaluation methods have been conducted [1, 10], with [10] broadly classifying evaluation methods/tools along two dimensions: the functionalities and goals of the method, and the perceived utility. A number of methods are based on the definition of a set of metrics grounded in Graph Theory and Metric Theory; however they are often criticised for not being directly applicable due to their level of abstraction. Cohesion metrics have been proposed in order to evaluate the degree of the relationships defined between components of an ontology [24]. These cohesion metrics are a good indicator of the level of modularity in the ontology, but they are mainly related to the taxonomic structure (being based on path traversal), and fail to take account of any axiomatic semantics that may exist. This criticism has similarly been raised for many ontology metrics, and [22] porposes to improve the design of ontology metrics for OWL ontologies [22].

Other approaches evaluate ontologies from an application specific perspective, and attempt to identify how they can be generalised for task based ontology evolution [16, 17]. However, the majority of these approaches are only successful with specific tasks and cannot be readily generalised. In particular, Sabou *et al* [18] identify two issues related to task-based ontology evaluation: the difficulty in assessing the quality of the supported task, and of creating an experimental evironment where no external factors influence the performance.

Effort has been devoted to defining and providing a principled methodology for the manual [21] or automatic [9] evaluation of ontologies. For example, manual cases are inspired by engineering approaches, whereas others coherently integrate the various aspects on evaluation, such as [9]. In some cases, formal properties of ontologies have been used to provide a principled approach to evaluating the correctness of an ontology [23].

Whilst the various works presented consider the different aspects of ontology evaluation, there are no approaches to date, to the best of our knowledge, that have explicitly considered the evaluation of ontology modules generated by ontology modularization mechanisms. Although some evaluation metrics consider the degree of modularity of an ontology (for instance, [24] measure the degree of cohesion of the components of an ontology) such methods do not assess whether or not a module is a good module. Size plays a central role in the evaluation of a module, as it is the only way to provide a comparative analysis of the different modularization techniques [4], that encompasses the requirements of the modularization algorithms. However, we argue that size is a coarse measure for comparative purposes, as modules with equal size might have different information content. Whilst there is no 'objective' way to assess the quality of the module, indeed there will probably never be an objective scale for delineating 'good' ontologies in general, better discriminating measures should be used where possible.

6 Conclusions And Future Work

This paper argues that an entropy inspired measure is a better discriminating factor than size when comparing ontology modules. The intra-technique evaluation of the experimental results show that in most cases entropy is a preferable discriminating factor than size. Furthermore, the improved entropy measure presented in Section 3 allows the

Ontology Engineer to assess what contributes to the overall entropy of the ontology module by calculating the entropy at the domain and language levels, as shown by the inter-technique evaluation (see Section 4.2).

The assumptions made at the moment, such as all language level edges being equal, could possibly be relaxed to further improve the entropy measure. For example, it is possible to argue that a ‘disjoint’ edge carries more information than a ‘subclass’ edge and, as such, should be weighted differently. This may further enhance the discriminating power of the measure.

Future work includes linking information content to a notion of usability and reusability. In principle, an ontology with low entropy has less information content, and thus is likely to be highly reusable, but not highly usable; whereas an ontology with high entropy will be the opposite. An entropy inspired measure could help to bridge the gap between the subjective evaluation of an Ontology Engineer and the objective measures available to them. There is a need to carry out an in depth study which compares existing measures used within ontology evaluation to both size and entropy inspired measures. This will hopefully identify the measures which are crucial to Ontology Engineers when they are evaluating ontology modules.

Acknowledgements EPSRC grant EP/D064287/1.

References

[1] J. Brank, M. Grobelnik, and D. Mladenić. A survey of ontology evaluation techniques. In *Proc. of 8th Int. multi-conf. Information Society*, pages 166–169, 2005.

[2] J. Calmet and A. Daemi. From entropy to ontology. In *AT2AI-4 - Fourth Int. Symposium "From Agent Theory to Agent Implementation" at 17th European Meeting on Cybernetics and Systems Research, Vienna, April 2004.*, 2004.

[3] C.E.Shannon. A mathematical theory of communication. Technical Report 27:379-423, 623-656, Bell System Technical Report, July and October 1948.

[4] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *J. of Artificial Intelligence Research (JAIR)*, 31:273–318, 2008.

[5] B. Cuenca-Grau, B. Parsia, E. Sirin, and A. Kalyanpur. Automatic partitioning of owl ontologies using e-connections. In *Proc. of the 2005 Int. Workshop on Description Logics (DL-2005)*, 2005.

[6] M. d'Aquin, M. Sabou, and E. Motta. Modularization: a key for the dynamic selection of relevant knowledge components. In *First Int.Workshop on Modular Ontologies, ISWC*, Athens, Georgia, USA., 2006.

[7] L. Ding, T. Finin, A. Joshi, Y. Peng, R. Pan, and P. Reddivari. Search on the semantic web. *Computer*, 38(10):62–69, 2005.

[8] P. Doran, V. A. M. Tamma, and L. Iannone. Ontology module extraction for ontology reuse: an ontology engineering perspective. In *Proc. of the 16th Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10*, 2007.

[9] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehman. Ontology evaluation and validation. an integrated formal model for the quality diagnostic task. Technical report, Laboratory for Applied Ontology, ISTC-CNR, 2005.

[10] A. Giboin, D. Maynard, J. Hartmann, M. del Carmen Suarez-Figueroa, R. Cuel, and Y. Sure. Methods for ontology evaluation. KWeb Deliverable D1.2.3, University of Karlsruhe, DEC 2004.

[11] A. Gomez-Perez, O. Corcho, and M. Fernandez-Lopez. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web.* Springer, July 2004.

[12] B. C. Grau, I. Horrocks, Y. Kazakov, and U. Sattler. A logical framework for modularity of ontologies. In *Proc. of the 20th Int. Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, January 6-12*, 2007.

[13] C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of the 20th Int. Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, January 6-12, 2007.*, 2007.

[14] N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proc. of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, 2000.

[15] N. F. Noy and M. A. Musen. Specifying ontology views by traversal. In *Int. Semantic Web Conference*, 2004.

[16] R. Porzel and R. Malaka. A task-based approach for ontology evaluation. In *Proc. of ECAI 2004 Workshop on Ontology Learning and Population*, Valencia, Spain, August 2004.

[17] M. Sabou, J. Gracia, S. Angeletou, M. D'Aquin, and E. Motta. Evaluating the Semantic Web: A Task-based Approach. In *Proc. of the 6th Int. Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea*, 2007.

[18] M. Sabou, V. Lopez, E. Motta, and V. Uren. Ontology selection: Ontology evaluation on the real semantic web. In *Proc. of EON 2006*, 2006.

[19] J. Seidenberg and A. Rector. Web ontology segmentation: analysis, classification and use. In *WWW '06: Proceedings of the 15th Int. Conference on World Wide Web*, 2006.

[20] H. Stuckenschmidt and M. Klein. Structure-based partitioning of large concept hierarchies. In *Proc. of the 3rd Int. Semantic Web Conference*, Hiroshima, Japan, 2004.

[21] A. L. Tello and A. Gómez-Pérez. Ontometric: A method to choose the appropriate ontology. *J. Database Manag.*, 15(2):1–18, 2004.

[22] D. Vrandecic and Y. Sure. How to design better ontology metrics. In *Proc. of the 4th European Semantic Web Conference (ESWC'07)*, 2007.

[23] C. Welty and N. Guarino. Supporting ontological analysis of taxonomical relationships. *Data and knowledge engineering*, 39(1):51–74, 2001.

[24] H. Yao, A. Orme, and L. Etzkorn. Cohesion metrics for ontology design and application. *Journal of Computer Science*, 1(1):107–113, 2005.