TWO METHODS FOR EVALUATING DYNAMIC ONTOLOGIES

Jaimie Murdock, Cameron Buckner, Colin Allen
Indiana University, Bloomington, IN
jammurdo@indiana.edu, cbuckner@indiana.edu, colallen@indiana.edu

Keywords: ontology evaluation, ontology evolution.

Abstract: Ontology evaluation poses a number of difficult challenges requiring different evaluation methodologies, particularly for a "dynamic ontology" representing a complex set of concepts and generated by a combination of automatic and semi-automatic methods. We review evaluation methods that focus solely on syntactic (formal) correctness, on the preservation of semantic structure, or on pragmatic utility. We propose two novel methods for dynamic ontology evaluation and describe the use of these methods for evaluating the different taxonomic representations that are generated at different times or with different amounts of expert feedback. The proposed "volatility" and "violation" scores represent an attempt to merge syntactic and semantic considerations. *Volatility* calculates the stability of the methods for ontology generation and extension. *Violation* measures the degree of "ontological fit" to a text corpus representative of the domain. Combined, they support estimation of convergence towards a stable representation of the domain. No method of evaluation can avoid making substantive normative assumptions about what constitutes "correct" representation, but rendering those assumptions explicit can help with the decision about which methods are appropriate for selecting amongst a set of available ontologies or for tuning the design of methods used to generate a hierarchically organized representation of a domain.

1 INTRODUCTION

The evaluation of domain ontologies that are generated by automated and semi-automated methods presents an enduring challenge. A wide variety of evaluation methods have been proposed; but it should not be assumed that one or even a handful of evaluation methods will cover the needs of all applications. Ontology evaluation is as multifaceted as the domains that ontology designers aspire to model. Projects differ in the resources available for validation, such as a "gold standard" ontology, measures of user satisfaction, explicitly stated assumptions about the logical or semantic structure of the domain's conceptualization, or a textual corpus or dictionary whose fit to the ontology can be measured. They will also differ in the goals of the evaluation – for instance, whether they aim to use evaluation to select amongst a set of available ontologies or to tune their methods of ontology design. Further, the methods will differ in the assumptions they make about their subject domains – for no evaluation method is possible without substantive normative assumptions as to the nature of the "right" ontology.

At the Indiana Philosophy Ontology (InPhO) project[1], we are developing techniques to evaluate the taxonomic structures generated by machine reasoning on expert feedback about automatically extracted statistical relationships from our starting corpus, the Stanford Encyclopedia of Philosophy (SEP). InPhO does not assume that a single, correct view of the discipline is possible, but rather takes the pragmatic approach that some representation is better than no representation at all (Buckner et al., 2010). Evaluation allows

[1] http://inpho.cogs.indiana.edu

us to quantify our model and makes explicit the specific biases and assumptions underlying each candidate taxonomy.

In this paper, we describe a pair of evaluation metrics we have found useful for evaluating ontologies and our methods of ontology design. These metrics are designed for projects which have access to large textual corpora, and which expect the structure of their ontology to fit the distribution of terms in this corpus. The volatility score (section 4.1) measures the structural stability over the course of ontology extension and evolution. The violation score (section 4.2) measures the semantic fit between an ontology's taxonomic structure and the distribution of terms in an underlying text corpus.

Before diving into these methodologies, we will first situate them within the broader evaluation literature (section 2). Then we will describe the InPhO in further detail, along with the raw materials we will be evaluating (section 3). After this, we explore each of the two new measures, labeling their assumptions and demonstrating their capacity to guide the process of ontology design.

2 STATE OF THE ART

Approaches to ontology evaluation are heavily dependent on the positions taken towards ontology structure and design. Different assumptions underlying these positions are often left implicit and this has led to a tangled web of conflicting opinions in the literature. However, Gangemi, Catenacci, Ciaramita and Lehmann (2006) provide an excellent conceptual scaffolding for use in detangling the web by establishing three categories of evaluation techniques:

- **Structural evaluation** inspects the logical rigor and consistency of an ontology's encoding scheme, typically as a directed graph (digraph) of taxonomic and non-taxonomic relations. Structural evaluations are a measure of syntactic correctness. A few examples of structural evaluation include the OntoClean system (Guarino and Welty, 2004) and Gómez-Pérez's (1999) paradigm of correctness, consistency and completeness, which was extended by Fahad & Qadir (2008). Our proposed *volatility score* (Section 4.1) is a structural evaluation of semantic consistency during successive stages of a dynamic ontology's iterative extension and evolution.
- **Functional evaluation** measures the suitability of the ontology as a representation of the target domain. Many functional evaluations follow a "gold standard" approach, in which the candidate ontology is compared to another work deemed a good representation of the target domain (e.g. Dellschaft & Staab (2008) and Maedche & Staab (2002)). Another approach is to compare the candidate ontology to a corpus from which terms and relations are extracted (Brewster et al., 2004). Our proposed violation score (Section 4.2) is a corpus-based functional evaluation of semantic ontological fit.
- **Usability evaluation** examines the pragmatics of an ontology's metadata and annotation by focusing on recognition, efficiency (computational and/or economic), and interfacing. The recognition level emerges from complete documentation and effective access schemes. The efficency level deals with proper division of ontological resources, and proper annotation for each. The interfacing level is limited by Gangemi et al. (2006) to the examination of inline annotations for interface design, but these are not essential properties.

  One chief measure of usability is compliance to standards such as OWL and RDFa. Several frameworks for social usability evaluation have been proposed by Supekar (2004) and Gómez-Pérez (in Staab, 2004). ONTOMETRIC is an attempt to codify the various factors in usability evaluation by detailing 160 characteristics of an ontology and then weighting these factors using semi-automatic decision-making procedures (Lozano-Tello and Gómez-Pérez, 2004).

These three paradigms of evaluation are realized in different evaluation contexts, as identified by Brank, Mladenic and Grobelnik (2005):

- **Applied** – For functional and usability evaluation, using the ontology to power an experimental task can provide valuable feedback about suitability and interoperability. Applied approaches require access to experts trained in the target domain and/or ontology design. Velardi, Navigli, Cucchiarelli, and Neri's OntoLearn system (2005) utilizes this type of applied evaluation metric. Porzel and Malaka (2005) also use this approach within speech recognition classification.
- **Social** – Methods for usability evaluation proposed by Lozano-Tello and Gómez-Pérez

(2004), Supekar (2004), and Noy (in Staab, 2004) for networks of peer-reviewed ontologies, in a similar manner to online shopping reviews. Most social evaluation revolves around the ontology selection task. These evaluations involve a purely qualitative assessment and may be prone to wide variation.

- **Gold standard** – As mentioned above, the gold standard approach compares the candidate ontologies to a fixed representation judged to be a good representation (Maedche and Staab, 2002; Dellschaft and Staab, 2008). These approaches draw strength from the trainability of the automatic methods against a static target, but the possibility of overtraining of automated and semi-automated methods for ontology population means that the methods may not generalize well.
- **Corpus-based** – Approaches such as those used by Brewster, Alani, Dasmahapatra, and Wilks (2004) calculate the "ontological fit" by identifying the proportion of terms that overlap between the ontology and the corpus. This is a particularly well-suited measure for evaluating ontology learning algorithms. Our methods expand this measurement approach to cover term relations through both the violation and volatility measures.

This collection of evaluation paradigms and contextual backdrops allows us finally to consider the type of information content being evaluated. A "computational ontology", such as the InPhO, is a formally-encoded specification of the concepts and a collection of directed taxonomic and non-taxonomic relations between them (Buckner et al., 2010; Gruber, 1995; Noy and McGuinness, 2001). When evaluating information content, we must be careful to delineate those which are node-centric (focusing on concepts) from those which are edge-centric (focusing on relations). Many authors (Maedche and Staab, 2002; Guarino and Welty, 2004; Brewster et al., 2004; Gómez-Pérez, 1999; Velardi et al., 2005) focus upon node-centric techniques, asking "Are the terms specified representative of the domain?" These investigate the lexical content of an ontology. However, the semantic content of an ontology is not defined solely by the collection of terms within it, but rather by the relations of these terms. Maedche & Staab (2002) take this initial lexical evaluation and extend it to an edge-based approach which measures the number of shared edges in two taxonomies. The proposed violation and volatility scores (Section 4) are novel edge-based measures which address the semantic content of an ontology by comparing them to statistics derived from a relevant corpus as a proxy for domain knowledge. Additionally, these scores can provide insight to the ontology design process by showing the controversy of domain content and convergence towards a relatively stable structure over time.

3 OUR DYNAMIC ONTOLOGY

A wide variety of projects can benefit from the development of a computational ontology of some subject domain. Ontology science has evolved in large part to suit the needs of large projects in medicine, business, and the natural sciences. These domains share a cluster of features: the underlying structures of these domains have a relatively stable consensus, projects are amply funded, and a primary goal is often to render interoperable large bodies of data. In these projects, the best practices often require hiring so-called "double experts" – knowledge modelers highly trained in both ontology design and the subject domains – to produce a representation in the early stages of a project which is optimally comprehensive and technically precise.

There is another cluster of applications, however, for which these practices are not ideal. These involve projects with principles of open-access and domains without the ample funding of the natural sciences. Additionally, ontologies for domains in which our structural understanding is controversial or constantly evolving, and projects which utilize computational ontologies to enhance search or navigation through asynchronously updated digital resources must account for the dynamic nature of their resources – whether it is in the underlying corpus or in the judgments of the experts providing feedback on domain structure. On the positive side, these areas often have more opportunities to collect feedback from users who are domain experts but lack expertise in ontology design.

For the latter type of project we have recommended an approach to design which we call *dynamic ontology*. While a project in the former group properly focuses the bulk of its design effort on the production of a single, optimally correct domain representation, the latter cluster is better served by treating the domain representation as tentative and disposable, and directing its design efforts towards automating as much of the design process as possible. Dynamic ontology,

broadly speaking, tries to take advantage of many data sources to iteratively derive the most useful domain representation obtainable at the current time. Two primary sources of data are domain experts and text corpora. Domain experts provide abstract information about presently-held assumptions and emergent trends within a field from a source, namely their own ideas, that is hard to examine directly. Text corpora make it possible to quantify what is meant by "domain" by providing a concrete encoding of the semantic space that is available for empirical analysis, in contrast to the ill-defined abstraction of "the domain is what the experts conceive of it as". From both kinds of sources many types of data may be gathered: statistical relationships among terms, feedback from domain experts, user search and navigation traces, existing metadata relationships (e.g. cross-references or citations), and so on. As more data become available and our understanding of the subject domain continues to evolve, the domain representation will be be dynamically extended, edited, and improved.

In dynamic ontology, problems of validation loom especially large due to the combination of heterogenous data sources. Each step in the design process presents modelers with a panoply of choices for inconsistency mitigation – e.g., which sources of data to favor over others, how to settle feedback disagreements, which reasoning methods to use for population, how much feedback to solicit, and how to weigh user feedback against statistical suggestions. The automation of ontology design is a field in its infancy, and very little is known about the optimal choices to satisfy specific design goals. Additionally, dynamic ontologists might have questions regarding representational stability. If the domain is itself in flux or controversial, modelers might want to know if they have captured that change. The quantity of feedback may also influence the convergence of a population method to some stable representation. The development of precise metrics about the relationship between an ontology and a domain may be useful in a answering these questions.

The InPhO is a dynamic ontology which models the discipline of philosophy. Our approach leverages expert knowledge by augmenting it with machine reasoning, greatly reducing the need for expensive "double experts". The primary source of text data and domain experts is the Stanford Encyclopedia of Philosophy (SEP)[2]. With over 700,000 weekly article downloads, the SEP is the leading digital humanities resource for philosophy. The corpus consists of over 1,200 articles and 14.25 million words maintained by over 1,600 volunteer authors and subject editors. The tremendous depth of the encyclopedia makes it impossible for any one person to have expertise over the whole domain, necessitating the creation of a useful organization scheme to provide better editorial control and content accessibility. At the same time, the comprehensive richness of the SEP makes it a reasonable proxy for the discipline of philosophy as a whole.

We begin with a small amount of manual ontology construction obtained through collaboration with domain experts. A lexicon is established from SEP article titles, Wikipedia philosophy categories, n-gram analysis and ad hoc additions by the InPhO curators. We then build on this framework using an iterative three-step process of data mining, feedback collection, and machine reasoning to populate and enrich our representation of philosophy (see Figure 1).

First, the SEP is mined to create a co-occurrence graph consisting of several statistical measures. For each term in our lexicon, information entropy is measured, which provides an estimate of relative generality. For each graph edge, we calculate the J-measure, which provides an estimate of semantic similarity. From these measures we are able to generate hypotheses about hypernym/hyponym candidates for sets of terms in the corpus (Niepert et al., 2007). Second, SEP authors and other volunteers verify these hypotheses by answering questions about relational hypotheses. This reduces the effect of any statistical anomalies which emerge from the corpus. Finally, logic programming techniques are used to assemble these aggregated feedback facts into a final populated ontology (Niepert et al., 2008). This knowledge base can then be used to generate tools to assist the authors, editors, and browsers of the SEP, through tools such as cross-reference generation engine and context-aware semantic search.

As was mentioned in the introduction, our pragmatic approach recognizes the likelihood that there is no single, correct view of the discipline. However, even if other projects do not agree with our taxonomic projections, our statistical data and expert evaluations may still be useful. By exposing our data from each of the three steps through an easy-to-use API, we encourage other projects to discover alternative ways to construct meaningful and useful representations of the dis-

[2]http://plato.stanford.edu

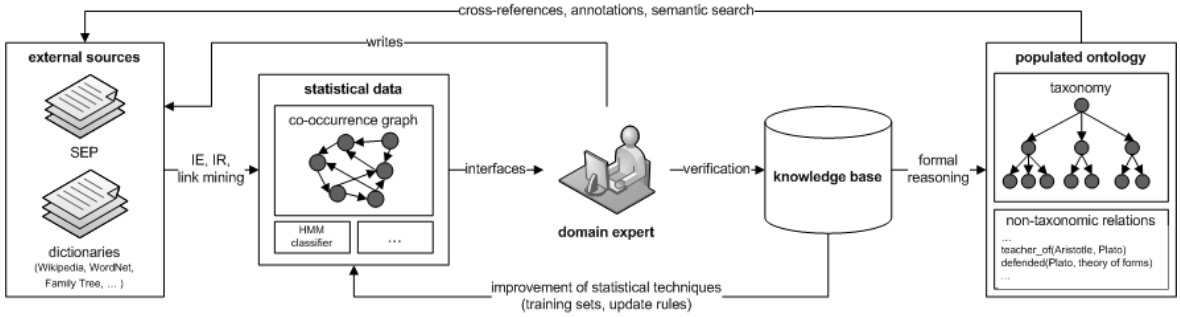

Figure 1: The InPhO Workflow

cipline. Additionally, by offering an open platform, we invite other projects to contribute relevant data and expert feedback to improve the quality of the service.

3.1 Raw Materials

In this section we describe the various components of our project which can be exploited for ontology evaluation.

3.1.1 Structure

The core of the InPhO is the *taxonomic representation* marked by the *isa* relations between concepts. *Concepts* in the InPhO may be represented as part of either class or instance relations. *Classes* are specified through the direct *isa* hierarchy of the taxonomy (see below). *Instances* are established between a concept and another concept which is part of the taxonomic structure. *Semantic crosslinks* (hereafter, *links*) can be asserted between two classes to capture the relatedness of ideas deemed mutually relevant by feedback or automatic methods.

3.1.2 Statistics

The InPhO's ontology population and extension techniques rely upon an external corpus (the SEP) to generate hypotheses about similarity and generality relationships. From this corpus we generate a co-occurrence graph $G = (V, E)$ in which each node represents a term in our set of keywords. An edge between two nodes indicates that the terms co-occur at least once.

For each node, the information content (Shannon entropy) is calculated:

$$H(i) = p(i) \log p(i) \tag{1}$$

For each edge, the directed J-measure (Smyth and Goodman, 1992; Niepert et al., 2007) and conditional entropy (Shannon, 1949) is calculated bidirectionally. The conditional entropy calculates the information content of a directed edge $i \to j$. This is used as a measure of semantic distance between two terms:

$$H(j \mid i) = p(i, j) \log \frac{p(i)}{p(i, j)} \tag{2}$$

The J-measure calculates the interestingness of inducing the rule "Whenever idea i is mentioned in a fragment of text, then idea j is mentioned as well" (Niepert et al., 2007). This is used as a measure of semantic similarity between two terms:

$$\begin{aligned} f(i \to j) =& p(j \mid i) \log \frac{p(j \mid i)}{p(j)} \\ &+ (1 - p(j \mid i)) \log \frac{1 - p(j \mid i)}{1 - p(j)} \end{aligned} \tag{3}$$

$$J(i \to j) = p(i) f(i \to j) \tag{4}$$

3.1.3 Methods

The taxonomy itself is populated through the use of answer set programming (Niepert et al., 2008). A population method $M(R, S, F)$ is specified by a set of rules R, a seed taxonomy S, and a set of expert feedback or statistical hypotheses F. Changes in F allow us to measure the impact of groups of expert feedback and to evaluate an ontology extension method. Proposed ruleset changes can be evaluated by maintaining the same set of inputs while testing variations in R. The seed taxonomy is used to reduce the computational complexity of a methodology, and changes to this seed can be used to strengthen the ontology design process. We currently have two

years of data collected on nightly repopulation of the published InPhO taxonomy, which is used for evaluation of our ontology extension methods.

3.2 Our Challenges

As hinted above, our dynamic approach to ontology design presents several unique challenges which require that appropriate validation methods be developed to address them. Specifically, there are a variety of different ways that our answer set program could infer a final populated ontology from aggregate expert feedback. For example, there are different ways of settling feedback inconsistencies (e.g. by leveraging user expertise in various ways (Niepert et al., 2008)), by checking for inconsistency between feedback facts (e.g. looking only at directly asserted inconsistencies or by exploring transitivities to look for implied inconsistencies), and by restricting the conditions in which an instance or link relationship can be asserted (e.g. forbidding/permitting multiple classification, forbidding linking to a node when already reachable by ancestry, etc.). It is difficult or impossible to decide which of these design choices is optimal *a priori*, and some precise evaluation metric would be needed to determine which ruleset variations tend to produce better results in certain circumstances.

Furthermore, our current methodology uses a manually-constructed seed taxonomy and populates this taxonomic structure through user feedback. Many options are possible for this initial hand-coded structure, and different experts would produce different conceptualizations; we might want a measure of which basic conceptualization tends to produce representations which best fit the distribution of terms in the SEP. More ambitiously, if we allow the answer set program to use disjunctive branching rules with regards to instantiation (thus creating multiple candidate ontologies from a single set of input), we could produce a large space of possible ontologies consistent with user feedback and a general theory of ontologies; the task would then be to rank these candidates according to their suitability for our metadata goals. Again, a precise evaluation metric which could be used to select the "best" ontology from this space is needed.

Another question concerns the amount of expert feedback needed before we begin to see diminishing returns. For example, we can only collect a limited amount of feedback from volunteer SEP authors and editors before the task becomes onerous; as such, we want to prioritize the collection of feedback for areas of the ontology which are currently underpopulated, or even pay some domain experts to address such sparseness. To optimize efficiency, we would want to estimate the number of feedback facts that are needed to reach a relatively stable structure in that area.

Finally, given that philosophy is an evolving domain rich with controversies, we might wonder how much our evolving representation of that domain captures these debates as they unfold. One of the alluring applications of dynamic ontology is to archive versions of the ontology over time and study the evolution of a discipline as it unfolds. This is doubly-relevant to our project, as both our domain corpus (the asynchronously-edited SEP) and our subject discipline are constantly evolving. The study of this controversy and the evolution resulting from it could be greatly enhanced by using metrics to precisely characterize change across multiple archived versions of the ontology.

4 OUR SCORES

By stressing the dynamic nature of philosophy, we do not mean to imply that the sciences lack controversy, or that scientific ontologies do not need ways of managing change. Nevertheless, whereas the sciences typically aim for empirically-grounded consensus, the humanities often encourage interpretation, reinterpretation, and pluralistic viewpoints. In this context, the construction of computational ontologies takes on a social character that makes an agreed-upon gold standard unlikely, and makes individual variation of opinion between experts a permanent feature of the context in which ontology evaluation takes place. Because of the dynamic, social nature of the domain, we do not try to achieve maximal correctness or stability of the InPhO's taxonomy of philosophical concepts in one step. But by iteratively gathering feedback, and improving the measures by which the ontology fit to various corpora can be assessed, we can hope to quantify the extent to which a stable representation can be constructed despite controversy among users. Our *volatility* score is designed to provide such a measure.

Many approaches to ontology evaluation, such as our volatility score, focus solely on syntactic (formal) properties of ontologies. These methods provide important techniques for assessing the quality of an ontology and its suitability for com-

putational applications, but stable, well-formed syntax is no guarantee that semantic features of the domain have been accurately captured by the formalism. By using the SEP as a proxy for the domain of philosophy, our *violation* score exploits a large source of semantic information to provide an additional estimate as to how well the formal features of our ontology correspond to the rich source material of the SEP.

4.1 Volatility Score

Most generally, a volatility score provides a measure of the amount of change between two or more different versions of a populated ontology.[3] Such a metric can serve a number of different purposes, including *controversy assessment* and *stability assessment* for a proposed methodology. As mentioned earlier, the ever-changing copora and domains modeled by a dynamic ontology are riddled with controversy. By comparing the changes between multiple archived versions of a populated ontology through a "directed volatility" score, we are able to track the evolution of a knowledge base over time. At the same time, we expect a proposed methodology to handle inconsistencies gracefully. By using random samples of expert feedback, we are able to test a ruleset variation's stability through a "grab-bag volatility" score. By adjusting the size of these random samples, we can also use this measure to determine how much feedback to solicit before reaching a point of diminishing returns with regards to stability.

While "volatility" represents a family of related methods, they all share the same basic intuition that some value is added to the aggregate volatility score each time the method "changes its mind" about asserting some particular link in the ontology (e.g. an instance switches from being asserted to not asserted under some class). For example, consider the representation of controversy over time: if *behaviorism* is said to be highly related to *philosophy of language* but a handful of expert evaluations indicate otherwise, our model would "change its mind" about asserting a link between *behaviorism* and *philosophy of language*. As other experts choose sides and weigh in on the matter, the volatility continues to increase, further pointing to an area of conflict. To consider another application, volatility can be used to indicate how much feedback is needed to reach stability for some area of the ontology by taking random subsets of feedback facts, and assessing the amount of volatility between ontologies generated from those random subsets. By increasing the size of the subset, we then see how much impact new feedback is having. Once we reach an acceptably low threshold for volatility, we can decide that collecting more feedback is not worth the effort and cost.

[3]We thank Uri Nodelman for early discussion of this idea.

4.1.1 Assumptions & Requirements

Volatility measures the structural stability of a set of ontologies or (derivatively) an ontology population method. Many in the semantic web community hold that domain ontologies are supposed to be authoritative descriptions of the types of entites in a domain (Smith, 2003). However, ontology development is often an iterative process (Noy and McGuinness, 2001), especially in dynamic ontology. The volatility score carries with it this assumption that a "final answer" description will not respond to the metadata needs of a dynamic corpus such as the SEP, Wikipedia, or WordNet. Additionally, a domain can undergo wide paradigm shifts, dramatically changing its conceptual landscape (Kuhn, 1962). The advent of new theories like quantum mechanics or new technologies like computers, for example, radically reshaped the conceptual landscape of philosophy. Therefore, the volatility score must be evaluated by domain experts to determine whether instability is due to undesirable errors/omissions in feedback or the machine reasoning program, or whether it instead properly highlights ongoing controversy within the field. In the former case, changes to the ontology extension methods can be made and evaluated against the old measure using the violation score. In the latter, these highlighted areas of controversy could be used to inform research in the field. In the case of the InPhO project, this could help facilitate analytic metaphilosophy (see Section 6.1 of Buckner, Niepert, and Allen (2010)).

4.1.2 Formalization

There are two subfamilies of volatility scores. One is the "directed volatility" which assesses the number of times an instance flips from being asserted to not asserted given an ordered set of ontologies. "Directed volatility" can be used to examine archived versions of an ontology and provide feedback about ontology extension methods. However, these directed measures will not be useful in calculating the amount of feedback needed

for the domain representation to reach some desired threshold of stability, as any ordering of populated ontologies derived from n random samples of z feedback facts would be entirely arbitrary. Thus we want a measure which does not require the ontologies to be ordered, but rather provides an estimate of how volatile that whole set is when mutually compared.

One way to achieve this is to consider the set of feedback facts not as a single entity which evolves over time, but rather as a supply of materials that can be used to populate an ontology. In a similar manner, we conceive of the populated ontology not as a whole representation, but as a bag of inferred instances. We then assess, for a set of n ontologies generated from random samples of z feedback facts and any pair of terms P and Q, the relative proportion of times $instance\_of(P, Q)$ is asserted vs. non-asserted. Thus, for any two terms P and Q, the basic formula for assessing the contribution of that pair to the overall volatility score is

$$v(P, Q) = 1 - \frac{|x - \frac{n}{2}|}{\frac{n}{2}} \quad (5)$$

where x is the number of times that the $instance\_of(P, Q)$ is asserted in the set under consideration. The total volatility is given by

$$volatility(z) = \frac{1}{count(P, Q)} \sum_{\forall P,Q} v(P, Q) \quad (6)$$

However, a complication is introduced here in that there are different etiologies which could lead $instance\_of(P, Q)$ to switch from being asserted/non-asserted. One way is for there to be a lack of any feedback facts relevant to that instance which could lead to the assertion of an $instance\_of$ relation; another is due to the resolution of an inconsistency in feedback facts (e.g. in one ontology a connection is asserted between P and Q due to a user's feedback, but not asserted in another because of contrary feedback from another user with a higher level of expertise). In order to isolate these issues, we adopt a "conservative" approach to assessing volatility: for any given pair of terms, we will only assess a volatility contribution across the subset of ontologies where at least minimal raw materials are present for asserting an $instance\_of$ relationship (e.g., $more\_specific(P, Q)$ and $highly\_related(P, Q)$). (It follows from this that no violation is assessed for pairs of terms which never have the raw materials for assertion across those random subsets of feedback.) We should still want to normalize this measure for the whole set of generated ontologies, because we would want to count an $instance\_of$ fact asserted 25 times out of 50 relevant ontologies (i.e. ontologies generated from the relevant raw materials) as more volatile than a $instance\_of$ fact which shifted 10 times out of 20 relevant ontologies (out of the 50 total generated). In this case, the equation is modified to

$$v'(P, Q) = 1 - \frac{|x - \frac{m}{2}|}{\frac{m}{2}} \frac{m}{n} \quad (7)$$

which reduces to

$$v'(P, Q) = 1 - \frac{|x - \frac{m}{2}|}{\frac{n}{2}} \quad (8)$$

and gives the sum volatility of

$$volatility(z) = \frac{1}{count(P, Q)} \sum_{\forall P,Q} v'(P, Q) \quad (9)$$

where m is the number of ontologies possessing raw materials for a possible feedback assertion, and n is the total number of ontologies generated for random sampling of z feedback facts.

4.1.3 Interpretation of Results

Depending upon the modeler's goals and assumptions about the domain, the volatility metric can be displayed in different ways and given different interpretations. Suppose, for example, that we want to visualize an unfolding controversy in the discipline. We may take some set of archived ontologies from the temporal beginning and ending of the controversy, and superimpose the volatility heat-maps for each pairwise volatility comparison between a time slice of the ontology and its temporal successor, coloring areas of change, perhaps gradually fading from one color to another as time goes on. "Hotter" areas of the visualization indicate areas of more persistent controversy, and the color shade indicates the trajectory of the dialectic over time. This would allow an expert to visualize the evolution of a controversy and its effects rather effectively in a quick display.

Suppose instead that our goal was to determine the amount of feedback needed for comprehensive and authoritative coverage of an area of our ontology. In that case, the volatility metric would be summed as indicated above for random samples of z feedback facts, and the net result would provide a volatility estimate for z facts that

could be compared to measures for other numbers of feedback facts or a predetermined threshold. In this case, volatility indicates not controversy, but rather the stability of the representation given that number of feedback facts, as well as how likely that representation is to change with the addition of more. Furthermore, we could look not just at the aggregate sum of individual pair volatilities, but rather display those on a heat map again. "Hotter" areas on this visualization might indicate areas which require more comprehensive or authoritative expert feedback, and thus could be used to direct the feedback solicitation process towards areas where it is most needed.

4.1.4 Preliminary Results

While we do not currently have enough feedback facts to reliably estimate the amount of feedback needed to achieve diminishing returns, we have tested the measure by taking random samples of z = 2000, 4000, 6000, and 8000 feedback facts, confirming that volatility does indeed decrease with increasing amounts of feedback even for our small data set. A problem for small data sets, however, is that the formalization of "grab-bag" volatility above depends upon the idealization that one can draw y non-overlapping random samples of z feedback facts from the whole population of possible feedback. Our current feedback consists of n = 8006 feedback facts. This is severely limiting to the type of evaluation we can presently do: At z = 2000, we can only take four samples without overlap. As z approaches n, the probability that the very same feedback facts will be chosen at each random sample increases exponentially (and thus exponentially reduces the volatility metric). While there are several possible methods to control for this confound, we require a much larger sample of feedback facts from which to draw our random samples. Further ideas as to how to deal with this confound are described in the Future Work section below.

4.2 Violation Score

For a candidate taxonomy, we introduce a "violation score" that is computed by assessing the degree to which its relative placement of terms diverges from statistically generated expectations about those terms relative locations in semantic space (as estimated by their corpus-derived similarity and relative generality measures). Similar to Dellschaft and Staab (2008), we consider violation on both a local and the global level. For local violations we only look at parent-child taxonomic relations. For the global violations, we look at the weighted pathwise distance between two terms in a taxonomy.

4.2.1 Assumptions & Requirements

One goal of ontology design is to produce a representation which captures the semantic structure of a domain. In order to have a concrete standard for evaluation, the violation score uses the distribution of terms in corpus, e.g. a reference work in that domain, as a proxy for the domain itself. Evaluation may thus draw upon the statistical measures outlined in Section 3.2.2. However, any metric relating an ontology's taxonomic relations to statistical measures carries with it implicit assumptions regarding the semantic interpretation of the ontology's structural properties, such as the interpretation of edges, pathwise distance, or genealogical depth. In order for the representation to be useful in end user applications (such as visualization, semantic search, and ontology-guided conceptual navigation), we consider several approaches to interpreting ontological structure, which may be adopted with varying degrees of strength:

- **Topic neutrality** – One might simply wish to regiment all of the vocabulary in a common structure representing only the *isa* relationships that exist among the various terms. The goal of such a taxonomy is simply to enforce a hierarchical structure on all the terms in the language. According to this approach, there is no implied semantic significance to the node depth (aka, genealogical depth) or to path length between pairs of nodes beyond the hierarchical semantics of the *isa* relation itself. For example, if English contains more levels of classificatory terms for familiar animals than it does for relatively unfamiliar organisms, a term such as "dog" may sit at a greater depth in the taxonomy from the root node than terms for other organisms that are similarly specific, but nothing of any semantic significance is implied by this depth (or the distance between term nodes) beyond the existence of the intervening terms in the language.
- **Depth as generality** – One might desire that all sibling nodes have approximately the same level of generality in the target domain,

making node depth (distance from the root node) semantically significant. On this view, the terms *dog* (a species) and *feline* (a family) should not be at the same depth, even if the language of the domain or corpus contains the same number of lexical concepts between *dog* and *thing* as between *feline* and *thing*. Here one expects the entropy of terms at the same depth to be highly correlated.[4]

- **Leaf specificity** – One might desire that all leaf nodes in the structure represent approximately the same grain of analysis. On this view, regardless of node depth, leaves should have similar entropy. Thus, for example, if *hammerhead shark* and *golden retriever* are both leaf nodes, leaf specificity is violated if these terms are not similarly distributed across the corpus that is standing proxy for the domain.

Choices among these desiderata are central to any argument for edge-based taxonomic evaluation. This is especially true for gold standard approaches which implicitly hold the relations of two candidate ontologies to be semantically equivalent. Additionally, we suspect that most domains have asymmetric taxonomic structures: subtrees of sibling nodes are not typically isomorphic to one another, and this means that even within a given taxonomy, path length between nodes and node depth may not have the same semantic significance.

In our comparison methods we assume that node depth is topic neutral – that is, node depth bears little correlation to specificity or generality on a global level. However, by definition, a child node should be more specific than its parent node. Thus, we measure local violation by comparing the information content of the parent and child nodes. When two terms are reversed in specificity we can count this as a syntactic violation of the taxonomic structure. Additionally, we can expect sibling instances to be closely related to one another and to their parent node by statistical measures of semantic distance. An instance is in violation if it is an outlier compared to the rest of its siblings.

[4]Edge equality provides a special case of depth as generality. The latter requires only that all edges at a given level represent the same semantic distance, whereas edge equality also requires these distances to be consistent between the different levels (e.g., the movement from a species to a genus represents the same conceptual distance as that between an order and a class).

We propose that overall violation is an emergent property from these localized semantic violations. These violations are each weighted by the magnitude of the error, ensuring that an ontology with several large mistakes will have greater violation than one with many minute errors.

4.2.2 Formalization

A *generality violation* (g-violation) occurs when two terms are reversed in specificity (e.g., the statistics propose that *connectionism* is more specific than *cognitive science* but the answer set asserts that *cognitive science* is more specific). For two terms S and G, where S is more specific than G, we hypothesize that the conditional entropy will be higher for for G given S than for S given G.

$$H(G \mid S) > H(S \mid G) \tag{10}$$

This makes intuitive sense if one considers the terms dog (S) and mammal (G). The presence of the term *dog* will lend far more certainty to the appearance of *mammal* than the other way around -- mentioning *mammal* is not very predictive of dog.

If this inequality does not hold, we take this as a generality violation (g-violation):

$$gv(S, G) = H(S \mid G) - H(G \mid S) \tag{11}$$

The mean of the g-violations is then taken to give the overall g-violation.

$$violation_g(O) = \frac{1}{count(S, G)} \sum_{\forall S,G} gv(S, G) \tag{12}$$

A *similarity violation* (s-violation) occurs when an instance's semantic similarity to its parent class is an outlier compared to the rest of its siblings. For example, the entity *(ideas about) federalism* has been observed under both *(ideas about) social and political philosophy* and *(ideas about) forms of government*. However, the siblings of *federalism* under *forms of government* are much closer to their parent node, than those under *social and political philosophy*. Therefore, a taxonomy asserting that *federalism* is an instance of *social and political philosophy* will recieve higher violation than one in which *federalism* is an instance of *forms of government*.

Semantic similarity can be measured using a variety of measures reviewed in Jiang and Conrath (1997) and Resnik (1999). We use the measure presented in Lin (1998):

$$sim(x_1, x_2) = \frac{2 \times \log P(C)}{\log P(x_1) + \log P(x_2)} \quad (13)$$

Such that x_1 and x_2 are entities in the taxonomy, and C is the most specific class which subsumes x_1 and x_2. As we are simply comparing an instance S to its parent G, we can use:

$$sim(S, G) = \frac{2 \times \log P(G)}{\log P(S) + \log P(G)} \quad (14)$$

The degree of s-violation can be determined by the standard score, which normalizes the values by standard deviation:

$$sv(S, G) = \frac{sim(S, G) - \mu}{\sigma} \quad (15)$$

where x is the raw semantic distance, μ is the mean of the semantic distance to the parent of all sibling nodes and σ is the standard deviation of this population. The final s-violation is calculated as the mean of s-violations.

$$violation_s(O) = \frac{1}{count(S, G)} \sum_{\forall S,G} sv(S, G) \quad (16)$$

4.2.3 Interpretation of Results

The violation score is intended as way to select the best representation of a given set of input parameters. In our methodology, the violation score is used to test variations in ruleset changes or seed taxonomies. This evaluation can be used throughout the ontology design process to perfect methodology. We have used violation to examine changes to the assertion of semantic crosslinks and in the weighting of expert feedback obtained from novice philosophers, undergraduate majors, graduate students, and professors of philosophy.

Additionally, we are able to use the violation score to compare different samples of expert feedback by using the same seed taxonomy and ruleset. The changes in violation scores exposed a steady increase in taxonomic fit from novices to undergraduates to graduate students, before a slight decrease with professors. Further investigation of violations found that our highest-level experts were more likely to go against the statistical prediction in often useful ways, further justifying the solicitation of feedback. Note that this starkly illustrates the limits of this method of corpus-based ontology validation: in this case, we solicited expert feedback precisely because we regarded the co-occurrence statistics as less than perfectly reliable, and in general the judgments of experts are regarded as more trustworthy than the evaluation metrics generated from those co-occurrence statistics. As such, we would obviously *not* infer that the ontology generated from the inclusion of expert feedback is less desirable than that without. In general, one should keep in mind during evaluation that one should not evaluate representations generated using one source of data against evaluation metrics generated using another, less-trusted source of data. In practice, this complicates even comparisons between different versions of the ruleset, for we must carefully reason through whether some particular ruleset change could be subtly biasing the representation towards or against expert feedback (e.g., in the way it settles inconsistency between users and experts).

4.2.4 Experimental Results

Since deploying the initial version of our answer set program (described in Niepert et al. 2008), we discovered a number of possible improvements, but could not be sure a priori which version of the ruleset would produce better results. The violation score provides us with a way to compare these options in terms of their suitability. We identified three binary parameters along which our program can vary, and have compared the violation scores for each possible combination (resulting in a 2x2x2 matrix). The three parameters are briefly described under their abbreviated names below.

- **"plink"** – Our original ruleset (Niepert et al. 2008) included non-taxonomic "links" to allow reachability between entities which were semantically related but which, for various reasons, could not be connected taxonomically. To minimize unnecessary taxonomic relations, we added a rule (hereafter, the "nins" rule) which blocked an instance X from being asserted as an instance of a class Y if there was also evidence that X was an instance of class Z and Y was possibly linked ("plink"ed) to Z (since in that case X would already be reachable from Y via the $Y \rightarrow Z$ link). Unexpectedly, we found that this occasionally produced an undesirable "reciprocal plink deadlock" (see Figure 2): whenever links were possible from both $Y \rightarrow Z$ and $Z \rightarrow Y$, the nins rule blocked X from being inferred as an instance of either Y or Z (and thus X often

became a taxonomic "orphan"). As such, we created a second version of the program which added a "no plink" restriction to the "nins" rule, preventing this reciprocal plink situation. The "plink" parameter indicates that this restriction was added to the nins rule.

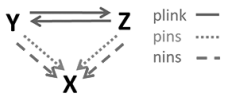

Figure 2: The reciprocal plink problem

- **"voting"** – An important innovation of our project involves the stratification of user feedback into different levels of self-reported expertise and using this information in a two-step process to resolve feedback inconsistencies. The first step in this process involves the application of a "voting filter" which settles *intra*-strata feedback inconsistencies using a voting scheme and can be completed as a preprocessing step before the answer set program is run (as described in Niepert et al 2009). The "voting" parameter indicates that this filter was run.
- **"trans"** – Much of the information on which our program operates is derived from the transitivity of the "more general than"/"more specific than" feedback predicates. The second step of our method for settling feedback inconsistencies involves settling *inter*-strata inconsistencies, which is completed from within our ruleset. However, transitivities in feedback can be computed either before or after these inter-strata inconsistencies are resolved (the former resulting in many more inconsistencies requiring resolution). The "trans" parameter thus indicates that this version of the ruleset computes transitivities *before* (vs. *after*) our ruleset settles inter-strata inconsistencies.

Each modification was then compared to the current ruleset using both the s-violation and g-violation metrics using corpus statistics and user evaluations from July 24, 2010 (see Figure 3). The number of instances asserted is also included. As we can clearly see, every proposed change decreased both violation scores, with the best results provided by adopting all three changes[5]. The decrease in s-violation can be interpreted as the development of denser semantic clusters subsumed under each class. The decrease in g-violation can be interpreted as movement towards greater stratification in the heirarchy. This is quantitative evidence that the principled design choices outlined above will provide useful additions to the ontology enrichment process.

[5]g-violation was lowest when adopting the plink and voting changes, but not trans. However, the result with all three changes was second lowest.

5 FUTURE WORK

With these methods of evaluating ontology structure and function in hand, along with preliminary results on our limrited feedback collection, we propose to continue these evaluation experiments as new feedback is rapidly collected from SEP authors. These scores will allow us to pursue a long-desired use of our answer set programming to infer a space of populated ontologies and select an optimal one by ranking them according to violation scores. We can then see how consistent ruleset selection is.

We might also ask how feedback from people with different levels of expertise in philosophy affects the placement of terms in the InPhO. For instance, Eckart et al (2010) have already gathered feedback data from Amazon Mechanical Turk (AMT) users and compared their responses to those of experts. Although we know that as a whole they differ statistically from experts, we do not yet know how much this matters to the structure that is eventually produced from those feedback facts.

As for the confound of overlapping samples in the calculation of "grab-bag" volatility (see section 4.1.4), an ideal solution is to solicit more feedback, increasing the amount of non-overlapping samples of a size z. Collecting generalized feedback from lower levels of expertise is economically feasible using AMT. Additionally, we can isolate small sections of the ontology to gather a very large amount of expert feedback from SEP authors in order to determine the point of diminishing returns for that location and extrapolate that result to estimate the amount of feedback required for other sections.

Finally, the InPhO has daily archives of its populated ontologies from October 23, 2008 to the present (July 25, 2010). By using the volatility measure on this data set, we should gain insights into our own ability to capture controversy and convergence within a field and be able to present that to philosophers through the visual-

	s-violation		g-violation		instances	
	all-in	voting	all-in	voting	all-in	voting
current	0.8248	0.8214	-0.1125	-0.1170	417	456
plink	0.8111	0.8089	-0.1182	-0.1227	521	568
trans	0.8119	0.8094	-0.1133	-0.1168	452	491
plink, trans	0.8061	0.8031	-0.1153	-0.1188	502	546

Figure 3: Violation score evaluations on the InPhO using feedback and corpus statistics from July 24, 2010

izations described in Section 4.1.3.

6 CONCLUSIONS

In this paper we have proposed two methods for evaluating the structural and functional aspects of a corpus-based dynamic ontology. Our work focuses on the semantic evaluation of taxonomic relations, rather than the lexical evaluation undertaken by Brewster et al. (2004) and Dellschaft & Staab (2008). The violation score gives us a concrete measure of how well an ontology captures the semantic similarity and generality relationships in a domain by examining statistical measures on an underlying corpus. The volatility score exposes areas of high uncertainty within a particular ontology population method, which can be used for many purposes. Directed measures of volatility can indicate the evolution of a knowledge base and highlight areas of controversy. Non-directed measures can indicate the stability of a ruleset variation by using random samples of expert feedback. This can also estimate the amount of expert feedback required for a convergent representation. We also have examined the considerations necessary to examine a taxonomy, and demonstrated how these methods have been used to enhance the enrichment process of the Indiana Philosophy Ontology Project through experiments on ruleset variations, expert feedback stratification and stability.

ACKNOWLEDGEMENTS

During the preparation of this manuscript, the first author was supported by grants from the Cognitive Science Program and Hutton Honors College at Indiana University. The research described in this paper has been funded with grants from the United States National Endowment for the Humanities Division of Preservation and Access and the NEH Office of Digital Humanities.

REFERENCES

Brank, J., Grobelnik, M., and Mladenic, D. (2005). Survey of ontology evaluation techniques. In *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD)*.

Brewster, C., Alani, H., Dasmahapatra, S., and Wilks, Y. (2004). Data driven ontology evaluation. In *Proceedings of LREC*, volume 2004.

Buckner, C., Niepert, M., and Allen, C. (2010). From encyclopedia to ontology: Toward dynamic representation of the discipline of philosophy. *Synthese*.

Dellschaft, K. and Staab, S. (2008). Strategies for the Evaluation of Ontology Learning. In Buitelaar, P. and Cimiano, P., editors, *Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*, pages 253–272. IOS Press.

Eckert, K., Niepert, M., Niemann, C., Buckner, C., Allen, C., and Stuckenschmidt, H. (2010). Crowdsourcing the Assembly of Concept Hierarchies. In *Proceedings of the 10th ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, Brisbane, Australia. ACM Press.

Fahad, M. and Qadir, M. (2008). A Framework for Ontology Evaluation. In *Proceedings International Conference on Conceptual Structures (ICCS), Toulouse, France, July*, pages 7–11. Citeseer.

Gangemi, A., Catenacci, C., Ciaramita, M., and Lehmann, J. (2006). Modelling ontology evaluation and validation. In *The Semantic Web: Research and Applications*, pages 140–154. Springer.

Gómez-Pérez, A. (1999). Evaluation of taxonomic knowledge in ontologies and knowledge bases. In *Proceedings of the 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Alberta, Canada*.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43(5):907–928.

Guarino, N. and Welty, C. A. (2004). An overview of OntoClean. In Staab, S. and Studer, R., editors, *Handbook on ontologies*, chapter 8, pages 151–159. Springer, 2 edition.

Jiang, J. and Conrath, D. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, number Rocling X, Taiwan.

Kuhn, T. (1962). *The Structure of Scientific Revolutions*. University of Chicago Press.

Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Citeseer.

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

Maedche, A. and Staab, S. (2002). Measuring similarity between ontologies. *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web*, pages 15–21.

Niepert, M., Buckner, C., and Allen, C. (2007). A dynamic ontology for a dynamic reference work. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, page 297. ACM.

Niepert, M., Buckner, C., and Allen, C. (2008). Answer set programming on expert feedback to populate and extend dynamic ontologies. In *Proceedings of 21st FLAIRS*.

Noy, N. and McGuinness, D. (2001). Ontology development 101: A guide to creating your first ontology.

Porzel, R. and Malaka, R. (2005). A task-based framework for ontology learning, population and evaluation. In Buitelaar, P., Cimiano, P., and Magnini, B., editors, *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press, Amsterdam.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of artificial intelligence research*, 11(4):95–130.

Shannon, C. E. (1949). *A mathematical theory of communication*. University of Illinois Press, Urbana, Illinois.

Smith, B. (2003). Ontology. In Luciano, F., editor, *Blackweel Guide to the philosophy of computing and information*, pages 155–166. Blackwell, Oxford.

Smyth, P. and Goodman, R. (1992). An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):301–316.

Staab, S., Gómez-Pérez, A., Daelemans, W., Reinberger, M.-L., Guarino, N., and Noy, N. F. (2004). Why evaluate ontology technologies? because it works! *IEEE Intelligent Systems*, 19(4):74–81.

Supekar, K. (2004). A peer-review approach for ontology evaluation. In *8th Int. Protege Conf*, pages 77–79. Citeseer.

Velardi, P., Navigli, R., Cucchiarelli, A., and Neri, F. (2005). Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. In Buitelaar, P., Cimiano, P., and Magnini, B., editors, *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press, Amsterdam.