

Ontology Evaluation Using Wikipedia Categories for Browsing

Jonathan Yu

James A. Thom

Audrey Tam

School of Computer Science and IT
RMIT University, GPO Box 2476V
Melbourne, Australia

{jonathan.yu,james.thom,audrey.tam}@rmit.edu.au

ABSTRACT

Ontology evaluation is a maturing discipline with methodologies and measures being developed and proposed. However, evaluation methods that have been proposed have not been applied to specific examples. In this paper, we present the state-of-the-art in ontology evaluation - current methodologies, criteria and measures, analyse appropriate evaluations that are important to our application - browsing in Wikipedia, and apply these evaluations in the context of ontologies with varied properties. Specifically, we seek to evaluate ontologies based on categories found in Wikipedia.

Categories and Subject Descriptors

H.3.4 [Systems and Software]: Performance evaluation (efficiency and effectiveness)

General Terms

Experimentation, Measurement, Performance

Keywords

Ontology evaluation, Browsing, User Studies, Wikipedia

1. INTRODUCTION

Ontology evaluation techniques are improving as more measures and methodologies are proposed. However, few specific examples of these evaluations have been found in literature. That is, specific examples of ontologies, applications and their requirements, measures and methodologies to link these together in one cohesive evaluation. This could partly be due to the lack of good ontologies made available publicly.

An *ontology* is an explicit model of a domain of knowledge consisting of a set of concepts, their definitions and interrelationships [17]. Parties commit to an ontology and agree upon its definitions and assertions. An agreement with an ontology is called an *ontological commitment* [8]. McGuinness describes the spectrum of ontology specifications from simple ontologies to structured [14]. *Simple ontologies* possess at the least a finite con-

trolled vocabulary, unambiguous interpretation of classes and term relationships, and strict hierarchical subclass relationships between classes, for example, the Yahoo categories. *Structured ontologies* take the simple ontology further and include more specific forms of expressivity and constraints on concepts and relations as well as axioms and equivalence mappings. The difficulty with some simple ontologies is that they tend to be loosely defined, small and often not agreed upon.

Recently, Wikipedia has become a medium for allowing users to contribute to articles on numerous subject areas over the WWW. Each article in Wikipedia can be accessed using its category structure. The Wikipedia category structure is equivalent to a simple ontology according to our definition above. However, in contrast to other simple ontologies it is a real application used by many users and one that is constantly refined by Wikipedia editors.

In this work, we consider domains in Wikipedia category structure as ontologies and seek to perform ontology evaluation measures proposed in the literature on them. Specifically, we take a task-based approach in our evaluation in the context of browsing articles using the category structure. Section 2 introduces Wikipedia and its category structure, and highlights its needs and requirements in the context of browsing. Section 3 briefly discusses what browsing is. In Section 4, we look at existing ontology evaluation approaches, criteria and measures proposed in literature. After discussing the requirements of the Wikipedia categories in the context of browsing and existing evaluation techniques in literature, we discuss and analyse which evaluation measures apply. Section 6 then presents our task-based evaluation involving users and we report on our findings from that user study. Lastly, Section 7 concludes this work and discusses some future work to pursue.

2. WIKIPEDIA

Wikipedia is a multi-lingual online encyclopedia written from volunteer contributions around the world. Since 2001, it has grown into a large pool of information with topics ranging from art, technology to pop. In the English language, it has over 1.7 million articles. Whilst anyone can add their contributions, they are subject to guidelines set by editors to ensure neutrality and that information is verifiable. The nature of Wikipedia lends itself for users to find information on a wide range of topics. It is also an ongoing and evolving application where information is continuously being updated, edited and discussed.

Apart from the article text, Wikipedia articles have various metadata attached to them. Within an article, it may contain hyperlinks to other related articles, external web pages, as well as one or more related categories. These categories are organised and structured to allow users to browse their way around to find related information.

Wikipedia's category structure may be seen as an information hi-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'07, November 6–8, 2007, Lisboa, Portugal.

Copyright 2007 ACM 978-1-59593-803-9/07/0011 ...\$5.00.

erarchy. It is by no means a strict and logically grounded ontology as it has many inconsistencies and is loose in its definition of relationships. However, it can be seen as a simple ontology rather than a formal one, as it has an explicit, shared and agreed upon conceptualisation. In this manner, it can be seen as one of the largest public ontologies available on the web having a large coverage of information, utilised by many users and is constantly evolving. However, it is not without any guidelines or requirements for its specification, which we discuss next.

2.1 Requirements of Wikipedia categories

The editors of Wikipedia have established guidelines for which categories are to be created (refer to online guidelines¹). The Wikipedia category structure is not complete nor is it a perfect one. However, these categories allow users to navigate around to find related information. Hence we adopt the Wikipedia category structure as our dataset and derive task domain for our ontology evaluations from it. We elaborate on the requirements of the Wikipedia categories drawn from the editors guidelines below.

Specifically, the requirements are:

1. Allowing intersecting category structure.

In Wikipedia, multiple views of categories exist at any one time. The rationale is for the category structure to be highly intersecting. This allows users to browse alternate but somewhat related domains and sometimes articles that they may not have expected to encounter but may find useful.

2. Group similar articles.

Categories help users find information on Wikipedia. Given an article, users are able to view similar articles by looking up its associated categories.

3. Categories should have the ‘right’ number of subcategories.

Given a category, the number of subcategories needs to be balanced. There should be a sufficient number of categories to facilitate effective browsing. On the other hand, too many subcategories will impede the user experience as the user needs to consider a large number of subcategories.

4. Avoiding cycles in the category structure.

Wikipedia does not prevent cycles in its category structure but the editors strongly discourage them. In general, they are not helpful for users as it can be confusing. In addition, cycles may impede some automated processes in its use of the category structure.

The type of information gathering activity, in using and navigating the Wikipedia category structure, fits with the characteristics of browsing more than it does with search. We will consider the distinction between these in the next section.

3. BROWSING

Browsing is affected by the **user’s knowledge of the domain** and the **specificity** of the browse task. It is characterised by **movement**. Thompson and Croft [16] describe browsing as an “informal or heuristic search through a well connected collection of records in order to find information relevant to one’s need”. In a browsing activity, users evaluate the information that is currently displayed, its value to the information need, and what further action to take. Thus, it is an informal search through a connected collection of documents.

A browsing activity is distinguished from a *search* activity. Both have goals in mind, however, Baeza-Yates and Ribeiro-Neto [1]

¹<http://en.wikipedia.org/wiki/Wikipedia:Category>

differentiate *search* from *browse* by the clarity of user goals. In *search*, users enter keywords into a system that is related to their information need. They are then presented with a list of results the system returns as related and users can decide to select one of the results or refine their search query. In comparison to this, *browsing* displays a different type of behaviour. There may not be a specific query as such associated. However, answers to user goals and information needs can be readily recognised in a browsing activity. Thus, it is the clarity and mode of accessing this information that differs in *browse*.

There are a different kinds of browsing. Marchionini [13] discusses these types of browsing from a directed browse to an undirected browse. A directed or intentional browsing behaviour is usually associated with tasks that are **closed or specific**. These refer to a task where there is usually not more than a few answers to the information need. On the other hand, an undirected browse is exploratory in nature and its browsing behaviour is associated with tasks that are more **open or broad**. These refer to a task where there may be many answers to the information need.

4. ONTOLOGY EVALUATION

Having introduced the Wikipedia categories as a simple ontology and its requirements, we now look at some methods of ontology evaluation. In this section, we will introduce the 3 main approaches to ontology evaluation, criteria for ontology evaluation and proposed measures from literature. We will then seek to match these methods of ontology evaluation to the requirements of the Wikipedia category structure in the next section.

With the increase of ontologies being made available via the WWW, evaluating which ontology is suitable becomes a problem. It is difficult to discern whether one ontology is better than another. If one is picked, it will lack definitions, axioms or relations required in a domain or application. If none is found to be suitable, an ontology may need to be built from scratch. However, the process in which ontologies are specified can be *ad hoc* at times. Whether an ontology is to be selected from a set of candidate ontologies or an ontology is to be constructed, methods for evaluating its suitability and applicability are needed. However, what are the means for evaluating an ontology? In this section, we will discuss the main approaches in ontology evaluation, some criteria for ontologies and some measures that have been proposed in literature.

4.1 Ontology Evaluation Approaches

There are 3 main approaches to ontology evaluation:

Gold standard evaluation This approach compares an ontology with another ontology that is deemed to be the benchmark. Typically, this kind of evaluation is applied to an ontology that is generated (semi-automatically or according to a learning algorithm) to compare whether the process of generating the ontology is effective. Maedche and Staab [12] give an example of a gold standard ontology evaluation. They propose ways to empirically measure similarities between ontologies both lexically and conceptually. The measures are based on the overlap in relations — Generic Relation Learning Accuracy measure. These measures determine the accuracy of discovered relations generated from their proposed ontology learning system compared with an existing ontology.

Criteria based evaluation This approach takes the ontology and evaluates it based on proposed criteria [5]. These criteria include consistency, completeness, conciseness, expandability and sensitivity, and depend on external semantics to perform the kind of evaluation that only humans are currently able

to do. However, it is difficult to construct automated tests to compare ontologies using such criteria [2]. Also, these criteria focus on the characteristics of the ontology in isolation from the application area. Hence, while ontology criteria may be met, it may not satisfy the needs of the application despite the fact that some application area needs may correspond with the ontology criteria.

Task-based evaluation This approach evaluates an ontology based on the competency of the ontology in completing tasks. In taking such an approach, we can judge whether an ontology is suitable for the application or task in a quantitative manner by measuring its performance within the context of the application. The disadvantage of this approach is that an evaluation for one application or task may not be comparable with another task. Hence, evaluations need to be taken for each task being considered.

Also, ontologies can be measured in various ways. What is measured may not necessarily tell us much nor evaluate the ontology in a very meaningful manner either.

4.2 Ontology Evaluation Criteria

Various criteria have been proposed for the evaluation of ontologies as listed in Table 1. These criteria can be used to evaluate the design of an ontology and in aiding requirements analysis.

Researcher	Proposed Criteria
Gruber [7]	Clarity Coherence Extendibility Minimal ontological commitment Minimal encoding bias
Grüninger and Fox [9]	Competency
Gómez-Pérez [5]	Consistency Completeness Conciseness Expandability Sensitiveness
Guarino [10]	Correctness (Identity & Dependence)
Guarino and Welty [11]	Correctness (Essence, Rigidity, Identity & Unity)

Table 1: Proposed Ontology Evaluation Criteria

Some of these criteria can be successfully determined using ontology tools. Reasoners, such as FaCT and RACER, provide the means to check for errors in ontologies, such as redundant terms, inconsistencies between definitions and missing definitions. Additionally, Dong et al. [3] have used existing software engineering tools and techniques to check for errors in ontologies in the military domain.

Some criteria, such as *clarity* and *expandability*, can be difficult to evaluate as there are no means in place to determine them. Moreover, while the *completeness* of an ontology can be demonstrated, it cannot be proven.

Other criteria can be more challenging to evaluate as they may not be easily quantifiable. They require manual inspection of the ontology. For example, *correctness* requires a domain expert or ontology engineer to manually verify that the definitions are correct with reference to the real world. This may not always be feasible for a large ontology or even a repository of many ontologies.

Upon analysis, some of the criteria proposed by the different researchers address similar aspects when evaluating ontologies and do overlap. We have previously described existing criteria proposed in literature and summarised these as 8 distinct criteria [18].

These criteria are:

1. Clarity
2. Consistency
3. Conciseness
4. Expandability
5. Correctness
6. Completeness
7. Minimal Ontological Commitment
8. Minimal Encoding Bias

Before we can apply criteria to the requirements for the Wikipedia categories, we need to discuss evaluation measures. Evaluation measures may help us to determine which criteria is applicable to the requirements.

4.3 Ontology Evaluation Measures

Ontology evaluation measures are a quantitative means in assessing various aspects of an ontology. Gómez-Pérez [6] outlines a list of measures looking at possible errors that could manifest with regards to ontology consistency, completeness and conciseness. Brewster et al. [2] propose measures for analysing whether an ontology has the right ‘fit’ over a given domain by applying coverage measures like precision and recall over a corpus representing the domain. Gangemi et al. [4] present a suite of measures focusing on structure, function and usability of an ontology. Tartir et al. [15] propose measures to evaluate an ontology’s capacity or “potential for knowledge representation”. The latter two focus on the *structural aspects* of an ontology. In our evaluations, we will be considering measures presented by Tartir et al. [15] and the *structural measures* from Gangemi et al. [4] as a means for analysing the Wikipedia requirements. Below we summarise and collate these into Table 2. Also, some of these measures are equivalent. These are presented in Table 3.

Tartir et. al [15]	Gangemi et al. [4]
Schema	Classes & Instances
Relationship richness	No. Classes
Attribute richness	No. Leaf Classes
Inheritance richness	Unique No. Instances
	Avg. Instances per class
	Max. Instances per class
Knowledge base - Instance	Breadth
Class richness	Absolute, Avg. & Max.
Avg. Population	
Cohesion	
Knowledge base - Class	Depth
Importance	Absolute, Avg. & Max.
Fullness	
Inheritance Richness (c)	Parents & Children
Relationship Richness (c)	No. Parent Classes
Connectivity	No. Children Classes
Readability	Avg. Children per Parent
	Max. Parents for any given child
	Fanout factor
	Tangledness
	Density
	Degree distribution
	Meta & Logical adequacy

Table 2: Structural measures

Tartir et al. [15]	Gangemi et al. [4]	Gomez-Perez [6]
Inheritance richness	Fanout	
Cohesion	Modularity	
	Logical adequacy	Consistency measures
	Meta-logical adequacy	Semantic inconsistency

Table 3: Equivalent measures

Despite advances in this area, there may not be a complete set of measures for all aspects of an ontology. As in the case of evaluation criteria, there may be parts of an ontology which are simply not measurable. For example, we cannot prove whether an ontology is complete [6]. There may also be other aspects that are difficult to measure in an ontology. For example, how do we determine adequate *expandability*? Having ontology measures does not mean that it is significant or important to us.

Measures that are feasible but done in isolation are not as meaningful compared with measures put into the context of indicators and benchmarks from application requirements or needs. An example is comparing various ontologies for adequate *coverage* in a domain or performance measures in an application deployment. The coverage or performance measures taken in the context of the application give meaning to the measures taken.

We will elaborate on the less intuitive proposed measures. *Tangledness* and *fanout* measures are related to how each category expand up with its parents and downward with its children. Measures looking at *relationship richness*, *attribute richness*, *class richness*, *average population* look at the quality of the overall ontology or knowledge base (if we include instances of classes). *Connectivity* and *cohesion* look at relations in the ontology. The definitions are drawn existing work [4, 15], which we will consider in our analysis are shown in Figure 1.

$$\text{Fanout factor (leaf to nodes)} = \frac{\text{No. Leaf Classes}}{\text{No. Classes}}$$

$$\text{Tangledness} = \frac{|\{G \wedge \exists a_1, a_2 (isa(m, a_1) \wedge (isa(m, a_2)))\}|}{n_G}$$

where n_G is the cardinality of G and $|\{G \wedge \exists a_1, a_2 (isa(m, a_1) \wedge (isa(m, a_2)))\}|$ is the cardinality of the set of nodes with more than one ingoing isa arc in graph g . That is, proportion of nodes that have more than one parent to all nodes in the graph.

$$\text{Relationship richness} = \frac{\text{No. Relations}}{\text{No. Subclasses} + \text{No. Relations}}$$

$$\text{Attribute richness} = \frac{\text{No. Attributes In All Classes}}{\text{No. Classes}}$$

$$\text{Class richness} = \frac{\text{No. Classes With Instances}}{\text{Total No. Classes}}$$

$$\text{Avg. population} = \frac{\text{No. Instances}}{\text{No. Classes}}$$

$$\text{Importance} = \frac{\text{No. Instances Belonging To Subtree}}{\text{No. Instances}}$$

Connectivity = No. Instances of other classes connected to instances of a given class

Cohesion = No. Separate Connected Components

Figure 1: Elaborated measures

5. MATCHING REQUIREMENTS

Our initial intention was to map requirements to criteria and criteria to the relevant measures. However, mapping criteria to the measures proved difficult. Some measures do not resolve to any criteria, for example, *connectivity* and *importance*. Some measures cover a range of criteria but not completely, for example, *depth* and *breadth*. These two measures related to *expandability* but the relation is limited. Some criteria are difficult to quantify, for example, *correctness*, *completeness*, *logical adequacy* and *minimal ontological commitment*. Furthermore, some measures are relevant to the requirements but do not resolve to any criteria, for example, *tangledness*.

In light of this, we perform an analysis to obtain measures which directly address requirements from our application - Wikipedia. We would still find some measures which will not directly nor fully address each requirement but at the very least, these measures are quantifiable.

5.1 Measures analysis

For brevity, we will be presenting measures which would partially or directly address the requirements.

5.1.1 Measures which address requirements

Depth / Breadth / Fanout.

These address requirement 3, having the ‘right’ number of subcategories. There may be cases where extremes in these measures will indicate that it may not be ‘right’, for example, high breadth or fanout. This may indicate that there are too many subcategories for a given category. In another example, high depth and low breadth may indicate too much categorisation happening or an incomplete set of subcategories at each level.

Tangledness.

Tangledness measures something of the distribution of multiple parent categories. This measure may help us understand how intersected the category structure is.

Degree distribution / Density.

Degree distribution and density are related measures and ascertain the probability a vertex has a certain ‘degree’. That is, the sum of parent categories and child categories. It may help indicate an ineffective subcategory structure if it is too dense or not dense enough. This may address the requirement of ‘right’ number of subcategories.

Cohesion / Modularity.

This measure gives the number of ‘islands’ or disjoint sets of categories. This could indicate that a more cohesive organisation of the categories is required. However it is unlikely to encounter disjoint category sets in a subtree of Wikipedia. All categories lead to the root category.

Importance / Connectivity.

Importance measures the distribution of instances in a subtree. The assumption is that if a class subtree has more instances, it indicates that it is more important than another subtree that does not have as many instances.

Connectivity indicates which nodes are highly connected.

An emerging pattern could be that the more highly connected categories will be general ones — like history, arts, information. Perhaps in combination, importance and connectivity could help suggest relevant categories that may allow a more intersecting structure.

Class richness.

Class richness measures the number of classes that are utilised by looking at the instances that have been attributed to them. This may highlight those classes that do not have articles. Although in Wikipedia it is unlikely for a category to be without an associated article (if we took articles to be instances of a category).

Circularity error.

Measuring circularity errors would address requirement 4 in avoid-

ing cycles within the category structure. However, we can avoid cycles by simply removing them. Hence, this measure is of little impact in our evaluations.

Table 4 shows a summary of measures that satisfy the requirements outlined in our application domain.

5.1.2 Measures not considered

Relationship richness and *attribute richness* measures are not considered because, with regards to Wikipedia, there are no other relationships between categories besides: a) parent and child categories; b) attributed articles to categories; and c) links from within article text to categories.

The *readability* measure is also not considered because the category structure does not have annotations regarding the design of the structure itself.

5.1.3 Measures addressing requirements

Of the measures indicated, *tangledness* addresses requirement 1 directly. Regarding *tangledness*, a highly tangled ontology would not be desirable for structured ontologies. However, in the context of Wikipedia, it is deemed beneficial and a requirement as it allows for greater intersectedness of the domain.

Depth, *breadth*, and *fanout* measures partially address requirements 2 and 3. These are easily measured. *Depth* and *breadth* measures are related. For ontologies with a similar number of classes, we would expect one that is very broad would be less deep. Conversely, a deep ontology would not be as broad. *Breadth* and *fanout* measures are also related. Changing the fanout factor would increase the breadth of an ontology. Thus, we propose to measure these as well.

5.1.4 Measures addressing requirements but not considered

For measures of *connectivity*, *importance* and *density* we find them to partially address requirements 1 and 3 respectively. However, it is not considered in our evaluations here as we do not expect these to have high impact on the requirements. Furthermore, these measures would also be difficult to vary in a systematic way. For example, how do we allow additional relations between classes in a meaningful way? Also, in the context of browsing in Wikipedia, allowing more relations to vary measures like density may hinder the browsability of the category structure.

In summary, we have identified *tangledness*, *depth*, *breadth* and *fanout* as measures to consider in our analysis looking at addressing requirements in browsing Wikipedia articles using its category structure.

6. TASK-BASED EVALUATION

In carrying out a task-based approach to ontology evaluation, we propose to model the task on the browsing of an information space using a given category structure — much in the same way users would do when browsing categories from Wikipedia. In this section, we describe the dataset used and ontologies taken from Wikipedia’s category hierarchy, the experimental design for the evaluations and present outcomes from a user study we undertook.

6.1 Wikipedia dataset and categories used

For this user study, we considered categories from the English-language version of Wikipedia and its associated articles. The articles were taken directly from a database dump of the articles from Wikipedia². Regarding the Wikipedia category structure, we ob-

²<http://download.wikimedia.org/enwiki>

Table 4: Measures-Requirements Analysis

Requirements:

- 1 - Allowing intersecting category structure
- 2 - Group similar articles
- 3 - Categories should have the ‘right’ number of subcategories
- 4 - Avoiding cycles in the category structure

Tartir et al. [15]	1	2	3	4
Schema				
- Relationship richness				
- Attribute richness				
- Inheritance richness				
Knowledge base - Instance				
- Class richness				
- Avg. Population				
- Cohesion				
Knowledge base - Class				
- Importance	o			
- Fullness				
- Inheritance Richness (c)			o	
- Relationship Richness (c)				
- Connectivity	o			
- Readability				

Gangemi et al. [4]	1	2	3	4
Depth		o	o	
Breadth		o	o	
Fanout			o	
Density			o	
Differentia Specifica				
Tangledness	•			
Modularity		o		
Logical adequacy				
Meta-logical adequacy				
Degree distribution			o	

Gómez-Pérez [6]	1	2	3	4
Consistency - Inconsistency Errors				
- Circularity errors				•
- Partition errors				
- Subclass partition with common instances				
- Subclass partition with common classes				
- Exhaustive subclass partition with common instances				
- Exhaustive subclass partition with common classes				
- Exhaustive subclass partition with external instances				
- Semantic inconsistency errors				
Completeness - Incompleteness Errors				
- Incomplete concept classification				
- Partition errors				
- Subclass partition omission				
- Exhaustive subclass partition omission				
Conciseness - Redundancy Errors				
- Grammatical redundancy errors				
- Redundancies of subclass-of relations				
- Redundancies of instance-of relations				
- Identical formal definition of some classes				
- Identical formal definition of some instances				

Brewster et al. [2]	1	2	3	4
Coverage				
Precision				
Recall				

• - Addresses o - Partially Addresses

tained this from System One’s RDF representation of it³. This was from a Wikipedia database dump dated March 2006. In it, each article and category is represented as an RDF triple with category and inter-article relations. The relations represented in the Wikipedia categories are: *category-subcategory*, *category-article* and *article-article relations*.

Upon analysis, we found that for a given category, no restrictions are put on the number of parent and sub categories. That is there may be multiple parent and child categories. Also, there are no restrictions as to the number of categories to associate an article with (as long as it is related).

However, there are some limitations with regards to the Wikipedia categories. Some categories are administrative in nature, for example, ‘*Sporting stubs*’. These are categories which contain articles that have yet to have information written for it but have been linked from another article previously. Also, not all articles have categories associated with it. This means that some articles are not viewable from navigating the category structure. Despite this, the Wikipedia categories are overall a content-rich organisation, which we will use as the basis for our task-based evaluations.

We applied measures to the Wikipedia categories, discussed in Section 3, which were feasible to apply. In processing the categories, we traversed the subtree in breadth first search fashion starting from the category ‘*Categories*’, which we take to be the root of the content section, and present these measures in Table 5.

Measure	Value
No. categories	111287
No. articles	1079246
Avg. articles per category	25.7
Levels	14
Categories with multiple parents	76578
No. parents	23978
Avg. no. parents	2.0
Max. parents for any given child	56
No. leaf categories	87309
Avg. no. children	4.64
Max. children	1760
Avg breadth	8559.5
Max breadth	33331
Avg depth	5.8
Max depth	13
Fanout factor	0.78
Tangledness	0.69

Table 5: Wikipedia Categories Measures

From Table 5 we observe that the Wikipedia categories have a ratio of about 1:10 with regards to the number of categories and its associated articles. Also, we find that the category structure is not deep considering the number of articles and categories with the number of levels as 14. Instead, we find it to be quite broad with an average breadth of 8559.5 in a given level. The overall Wikipedia category structure is also quite tangled with 69% of the all Wikipedia categories having multiple parents — that is, categories which have more than one parent.

6.2 Experimental setup

The goal of our task was to examine properties of the category structure through *browsability*. That is, being able to **locate information** using the category structure for articles for a given information need by browsing. Users should also be able to *explore the*

³<http://labs.systemone.at/wikipedia3>

domain space in an intuitive manner with reasonable information organisations.

6.2.1 Subtrees considered

For this user study, we needed to vary the original subtree in a manner that was: semantically reasonable, utilised all the categories in the subtree and comparable to the original subtree. There were two options presented to us — either vary the original Wikipedia subtree or generate a subtree category structure according to an automated technique — which was a variation on a document clustering technique.

Method for removing tangledness.

In exploring both options, we found that varying the original Wikipedia structure to remove tangledness was reasonable. Removing tangledness meant removing occurrences of multiple parents in a given category. The specific algorithm we used was *Dijkstra’s algorithm* for finding a single-source shortest path tree. This is the most appropriate shortest path algorithm since we know the root of the subtree. Where there were more than one parent candidate categories we chose the one that was most similar to the category being considered. We performed a *TF-IDF cosine similarity* measure on article titles within categories of a given subtree. We found this worked well and kept the subtree mostly semantically equivalent.

Method for generating subtrees.

We looked at varying the original Wikipedia subtree in a reasonable manner for reducing or increasing the number of subcategories to consider breadth and fanout factors. However, we could not find a feasible way of systematically varying the number of subcategories. Specifically, the difficulty faced was in increasing the number of subcategories in a sensible manner. Thus, we considered the second option from above — using a form of document clustering.

For a given subtree of the Wikipedia category hierarchy, we removed all category relations from it and applied a document clustering technique over the categories contained in the base subtree. We used *partition-based criterion-driven document clustering* on features gathered from a combination of the category title and associated article information [19] provided in the Cluto clustering toolkit⁴. Algorithm 1 describes the pseudocode for varying a given category subtree.

Algorithm 1 Varying a subtree

```

Let  $N :=$  maximum number of elements in cluster
Add root of subtree to queue  $q$ 
repeat
  Let  $c :=$  next item in  $q$ 
  Obtain clusters  $I$  for  $c$  from elements in its cluster
  for all  $i$  in  $I$  do
    Nominate element in  $i$  as representative category  $r$  for  $i$ 
    Add  $r$  as subcategory of  $c$ 
    Let  $clustersize :=$  number of elements in cluster  $c - 1$ 
    if  $clustersize \geq N$  then
      Add  $i$  to queue
    end if
  end for
until queue has no more clusters to process

```

We used the category title and clustered on a few varying data parameters: category title only, category title and the associated article titles, and category title and the associated article text.

⁴<http://glaros.dtc.umn.edu/gkhome/views/cluto>

We also varied the clustering technique based on the number of features considered and also the resulting number of clusters on each clustering event. We used the *cosine similarity function* for this.

Using the two methods discussed above, we can obtain 4 varied subtrees for a given domain. Table 6 presents the original and varied subtrees we considered. We use Subtrees *a* and *b* to look at the effects on tangledness and in future work, Subtrees *c*, *d* and *e* to consider depth, breadth and fanout.

Label	Subtree
<i>a</i>	Wikipedia original
<i>b</i>	Wikipedia original (remove tangledness)
<i>c</i>	Generated (untangled)
<i>d</i>	Generated (Fewer subcategories)
<i>e</i>	Generated (Increased subcategories)

Table 6: Subtree variations from Wikipedia

6.2.2 Tasks and Domains

We now outline the tasks and domains we used in our user studies. In each experiment, participants were given a set of tasks to complete within a 10 minute duration (inclusive of pre and post task questions). The given tasks were domain specific, and hence would not be reasonable in another domain. We chose to use domains that were as separate from each other as possible so as to reduce the learning effect from completing tasks on a given domain. Also, we chose 3 levels of specificity regarding the nature of the tasks (see Table 7). We proposed Tasks 1 to 3 and Tasks 4 to 6 to have increasing levels of specificity, from broad to specific, in their respective domains X and Y. For example, *International racing competitions* (Task 1) covered a broad range of possible answers within the Racing Sport domain (X). Whereas *Makers of F1 racing cars* (Task 3) was a very specific task type in the same domain.

Domain	Task Description
Racing Sport (X)	T1: International racing competitions
	T2: Racing sports without wheeled vehicles
	T3: Makers of F1 racing cars
Foods (Y)	T4: Non-alcoholic beverages
	T5: Different cuisines of the world
	T6: Wine regions in Australia

Table 7: Experiment design for comparing Wikipedia with generated structure

The tables below outline the experimental design we used to compare various aspects of the generated subtrees. We propose two experiments to obtain results for different comparisons. In each experiment, we used the *Latin squares* method of determining the order participants use the subtrees to be compared. We did this to remove the learning factor of users progressing from one subtree to another in a given domain and to increase statistical significance in our studies. Using this configuration we guarantee each user to have a unique sequence and for users to use each subtree in a different position. We also applied blocking on the domain. Lastly, we rotated the domain after 9 users.

Experiment 1.

First, in Table 8, the original Wikipedia subtree *a* is compared with two other variations: 1) the same subtree altered to remove

multiple parents *b*, hence being *untangled*; and 2) a generated subtree *c* with similar properties using the document clustering technique.

Participant	X			Y		
	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>
user 1	t1	t2	t3	t4	t5	t6
user 2	t2	t3	t1	t5	t6	t4
user 3	t3	t1	t2	t6	t4	t5
	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>
user 4	t1	t2	t3	t4	t5	t6
user 5	t2	t3	t1	t5	t6	t4
user 6	t3	t1	t2	t6	t4	t5
	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>
user 7	t1	t2	t3	t4	t5	t6
user 8	t2	t3	t1	t5	t6	t4
user 9	t3	t1	t2	t6	t4	t5
		Y			X	
	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>
user 10	t4	t5	t6	t1	t2	t3
user 11	t5	t6	t4	t2	t3	t1
user 12	t6	t4	t5	t3	t1	t2
	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>
user 13	t4	t5	t6	t1	t2	t3
user 14	t5	t6	t4	t2	t3	t1
user 15	t6	t4	t5	t3	t1	t2
	<i>c</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>a</i>	<i>b</i>
user 16	t4	t5	t6	t1	t2	t3
user 17	t5	t6	t4	t2	t3	t1
user 18	t6	t4	t5	t3	t1	t2

Table 8: Experiment design comparing *a*, *b* and *c*

Experiment 2.

We propose a second experiment, with a similar setup as the previous experiment but instead using the generated untangled Subtree *c* as our base subtree and compare it with generated subtrees that had varying number of subcategories – with Subtrees *d* and *e* having more subcategories and fewer subcategories respectively. With the generated subtrees, it is possible to adjust its breadth and fanout by altering the number of subclusters. This was to investigate the effect of depth, breadth and fanout. However, this depends on Subtrees *b* and *c* being approximately equivalent in performance. We needed to establish this in the first experiment described above.

6.2.3 Analysis of varied ontologies

After varying each subtree for the two domains, we took measurements on these to analyse the changes and present them in Tables 9 and 10. For the Racing sports domain (X), we had 1185 categories. For the Foods domain (Y), we had around 652 categories in total. These were ideal sizes for the time given to each user to browse through in that they were sufficiently large such that users would probably not look at all categories. They were also more general topic areas. These were typically between 15 and 20 articles in a category.

We observed that for each domain, Subtrees *b* and *c* do not have any multiple parents nor are they tangled. Untangled subtrees also reduce the number of parents in total in comparison with the Wikipedia original subtree (*a*). In effect, untangling a subtree removes links from the subtree.

Measure	All		
No. categories	1185		
No. articles	18178		
Avg articles per category	15.3		
	Subtree		
	<i>a</i>	<i>b</i>	<i>c</i>
Levels	7	7	4
No. parents	305	213	292
Categories with multiple parents	293	0	0
Avg. no. parents	1.3	0.9	0.9
Max no. parents for a given child	5	1	1
Leaf nodes	880	972	893
Avg. children	4.9	5.6	4.1
Max. children	54	53	20
Avg. breadth	169.3	169.3	296.3
Max breadth	459	458	765
Avg. depth	3.6	3.6	2.9
Max. depth	6	6	3
Fanout	0.74	0.82	0.75
Tangledness	0.25	0.00	0.00

Table 9: Racing sports subtrees

Measure	All		
No. categories	642		
No. articles	12630		
Avg. articles	19.7		
	Subtree		
	<i>a</i>	<i>b</i>	<i>c</i>
Levels	9	9	3
No. parents	187	135	51
Categories with multiple parents	132	0	0
Avg. parents	1.2	0.9	0.9
Max. parents for a given child	4	1	1
Leaf nodes	455	507	580
Avg children	4.2	4.8	12.4
Max children	48	48	51
Avg breadth	71.3	71.3	210.3
Max breadth	141	141	579
Avg depth	3.9	3.9	2.0
Max depth	8	8	2
Fanout	0.71	0.79	0.92
Tangledness	0.21	0.00	0.00

Table 10: Foods subtrees

The generated subtree (*c*) had fewer levels as they were generally broader than the others. This is also reflected in the average breadth. The effect of this is presenting the user with about twice as many narrower category links compared with the other subtrees.

We had several versions of generated subtrees with varying parameters. The best subtree was chosen on the basis of the number of *category-subcategory* relations in the generated subtree (*c*) that appeared in the original Wikipedia subtree (*a*). This tended to yield broad subtrees using our method for generating subtrees.

We also observed that broader subtrees had an effect in reducing subtree depth.

6.2.4 Evaluation

To evaluate the performance of users with regards to browsing and marking relevant articles for a given task, we propose browsing efficiency and effectiveness measures.

For efficiency, we looked at the number of backtracking a user does. Included are the number of clicks a user makes to:

1. go back to the **previous** category
2. go back to the **top** category
3. click on a **past category or article** from history links

For effectiveness, we considered the number of relevant articles users marked for each task. For each article marked, we evaluated the marked article as:

Not relevant: does not relate to the task

Somewhat relevant: has bits of relevant information

Mostly relevant: most of article is relevant

Definitely relevant: all of article is relevant

6.3 Results

We summarise our user study findings below:

User behaviour:

- exhibited exploratory behaviour if subtree did not help
- tended to backtrack more if the subtree was not helping with the task at hand

Original Wikipedia subtree (*a*):

- helped users perform better with tasks that were less broad

Wikipedia untangled subtree (*b*):

- generally users backtracked less on broader tasks
- generally found more definite relevant articles in the broadest task in both domains

Generated subtree (*c*):

- users obtain more *mostly-relevant* but not *definitely-relevant* articles in Domain X
- users tended to perform better from Domain Y than X
- was not equivalent to Wikipedia untangled

Figure 2 shows a comparison between the average number of user clicks for a given task on a given system and the breakdown of those clicks into average number of: *backtracking clicks*; *category clicks*; *article clicks*; and *other clicks*⁵

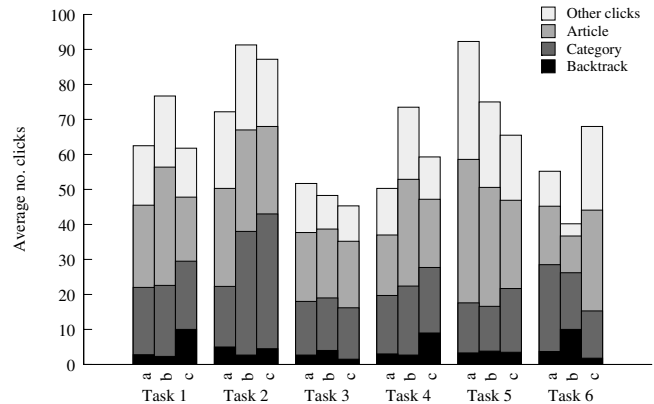


Figure 2: Average no. user clicks breakdown

Figure 3 compares systems on a given task for relevant articles retrieved by a user. The breakdown on each bar includes articles ranging from definitely-relevant to not-relevant.

We summarise our findings in Table 11 below, give further details on the main measures for each subtree and include significance tests. The major rows in Table 11 are grouped by *Individual Tasks*, followed by *Task Types*, and finally *Overall results*.

⁵*Other clicks* refers to clicks like marking relevant articles and re-viewing those articles

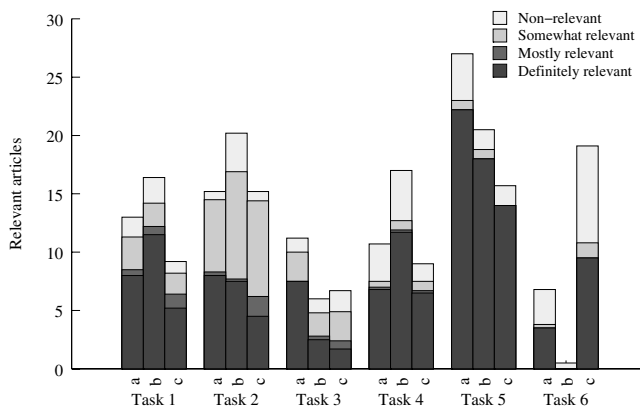


Figure 3: Relevant articles retrieved

For Overall Results, we omitted Task 6 as this task appeared to favour Subtree *c* over the others, as we elaborate later. The minor rows are grouped into backtracking clicks and measures of relevant articles found. We sum up measures of *Definitely*, *Mostly* and *Somewhat* relevant articles found and present this in the table as relevant articles found. The first column shows the task comparisons. We list the corresponding averages for each measure on each subtree beside it. The last set of columns in Table 11 presents p-values for the significance tests carried out on each measure. For the significance tests, we used a two-tailed unpaired unequal variance t-test. The p-value shows the probability that the distributions of the users’ performance values for the specific comparison are the same. We may consider the performance of a given subtree to be different from another with statistical significance if the p-value is lower than 0.05. That is, there is less than 5% chance that the two distributions are from the same population.

6.3.1 Wikipedia original vs. Untangled

Looking at the individual tasks, the main differences found between the Subtree *a* and the Subtree *b* were highlighted in Task 3 and 6. Of the set of tasks given, these two were the most specific.

In Task 3, users were asked to find articles about ‘*Formula One*’ car makers. We found that users performed three times better using Subtree *a* in finding more relevant articles. Using Subtree *a*, users found an average of 7.5 definitely relevant articles compared with Subtree *b* where they found 2.5 and the difference was statistically significant. Upon closer inspection of the category structure, the Formula One section of the subtree had many categories with multiple parents that were related which explains how the user was able to browse more effectively.

In Task 6, users had to find wine regions in Australia. Subtree *a* did significantly better than Subtree *b*. Using Subtree *a*, 4 out of the 6 users found relevant articles for this task, of which 3 users found definitely relevant articles, while all users using Subtree *b* failed to find any relevant articles.

In observing users performing Tasks 3 and 6, the key was in finding the specific **gateway** category. This gateway category opened up the relevant categories and were often clustered together around the gateway category. In task 6, this gateway category was more difficult to find in Subtree *b*. This was because there were relations missing from categories which users were looking in. The key category for task 6 in Subtree *b* was located in a related but obscure category called ‘*Herbs and Medicinal herbs*’. In contrast, users performing the task on Subtree *a* tended to find the key category *Wine* as a multiple parent of ‘*Grape varieties*’ which helped them

Task	Measure	Subtree			p		
		a	b	c	a-b	b-c	a-c
1	% backtrack	4.48%	3.00%	16.18%	0.46	0.02*	0.05
	Definitely	8	11.5	5.2	0.30	0.09	0.10
	Mostly	0.5	0.7	1.2	0.69	0.41	0.25
	Somewhat	2.8	2	1.8	0.38	0.84	0.20
	Relevant	11.3	14.2	8.2	0.43	0.12	0.10
	Non-relevant	1.7	2.2	1	0.69	0.38	0.45
2	% backtrack	6.93%	2.96%	5.16%	0.33	0.74	0.54
	Definitely	8	7.5	4.5	0.90	0.46	0.09
	Mostly	0.3	0.2	1.7	0.67	0.03*	0.05
	Somewhat	6.2	9.2	1.7	0.45	0.80	0.55
	Relevant	14.5	16.8	14.3	0.57	0.58	0.97
	Non-relevant	0.7	3.3	0.8	0.16	0.19	0.78
3	% backtrack	5.22%	8.28%	3.31%	0.61	0.53	0.82
	Definitely	7.5	2.5	1.7	0.03*	0.60	0.01*
	Mostly	0	0.3	0.7	0.36	0.50	0.10
	Somewhat	2.5	2	2.5	0.55	0.63	1.00
	Relevant	10	4.8	4.8	0.03*	1.00	0.03*
	Non-relevant	1.2	1.2	1.8	1.00	0.38	0.35
4	% backtrack	5.96%	3.67%	15.18%	0.40	0.16	0.35
	Definitely	6.8	11.7	6.5	0.18	0.14	0.89
	Mostly	0.2	0.2	0.2	1.00	1.00	1.00
	Somewhat	0.5	0.8	0.8	0.55	1.00	0.55
	Relevant	7.5	12.7	7.5	0.20	0.19	1.00
	Non-relevant	3.2	4.3	1.5	0.59	0.18	0.21
5	% backtrack	3.58%	5.07%	5.34%	0.69	0.93	0.58
	Definitely	22.2	18	14	0.54	0.35	0.23
	Mostly	0	0	0	-	-	-
	Somewhat	0.8	0.8	0	1.00	0.14	0.09
	Relevant	23	18.8	14	0.54	0.25	0.20
	Non-relevant	4	1.7	1.7	0.24	1.00	0.24
6	% backtrack	6.70%	24.88%	2.65%	0.01*	0.01*	0.31
	Definitely	3.5	0	9.5	0.08	0.02*	0.11
	Mostly	0	0	0	-	-	-
	Somewhat	0.3	0	1.3	0.17	0.08	0.17
	Relevant	3.8	0	10.8	0.07	0.03*	0.11
	Non-relevant	3	0.5	8.3	0.08	0.08	0.20
1 4	% backtrack	5.17%	3.33%	15.68%	0.25	0.01*	0.06
	Definitely	7.4	11.6	5.8	0.07	0.02*	0.27
	Mostly	0.3	0.4	0.7	0.73	0.48	0.31
	Somewhat	1.7	1.4	1.3	0.70	0.88	0.57
	Relevant	9.4	13.4	7.8	0.12	0.03*	0.34
	Non-relevant	2.4	3.3	1.3	0.50	0.10	0.14
2 5	% backtrack	5.07%	3.91%	5.24%	0.67	0.76	0.88
	Definitely	15.1	12.8	9.3	0.61	0.32	0.17
	Mostly	0.2	0.1	0.8	0.66	0.06	0.10
	Somewhat	3.5	5.0	4.1	0.54	0.72	0.79
	Relevant	18.8	17.8	14.2	0.81	0.21	0.24
	Non-relevant	2.3	2.5	1.3	0.90	0.19	0.32
3 6	% backtrack	5.93%	15.82%	2.94%	0.03*	0.01*	0.35
	Definitely	5.5	1.3	5.6	0.01*	0.05	0.97
	Mostly	0.0	0.2	0.3	0.34	0.51	0.10
	Somewhat	1.4	1.0	1.9	0.49	0.18	0.45
	Relevant	6.9	2.4	7.8	0.02*	0.03*	0.71
	Non-relevant	2.1	0.8	5.1	0.09	0.05	0.17
Overall minus 6	% backtrack	5.40%	4.34%	8.46%	0.45	0.05	0.14
	Definitely	10.5	10.2	6.4	0.90	0.04*	0.03*
	Mostly	0.2	0.3	0.7	0.63	0.03*	0.01*
	Somewhat	2.6	3.0	2.7	0.70	0.79	0.91
	Relevant	13.3	13.5	9.8	0.93	0.05	0.08
	Non-relevant	2.1	2.5	1.4	0.58	0.06	0.15

* denotes statistical significance (p < 0.05)

Table 11: Results and significance tests for subtrees

perform this task well.

Overall, Subtree *b* was comparable to Subtree *a*. If we exclude results from Task 6, we observe that backtracking on Subtree *b* was not significantly different to Subtree *a*. There is also little difference in performance between Subtree *a* and *b* in the number of relevant articles users found. However, there seems to be a trend in task types that were most general. For Tasks 1 and 4, Subtree *b* seems to perform better in terms of both backtracking and relevant articles found. This could point to users being more confident using Subtree *b* than with Subtree *a*. However, this was not observed to be statistically significant.

6.3.2 Untangled — Wikipedia vs. Generated

For the two domains, there were 2 generated subtrees made with different parameters. We found that users who performed tasks on Subtree *c* tended to have more ‘in-the-middle’ relevant articles. That is, articles that were mostly but not definitely relevant. This is highlighted in Task 2 where users found a statistically significant higher number of mostly relevant articles compared with the Subtree *b* having on average 1.7 articles that were mostly relevant as compared with 0.2 respectively. We can attribute this to how the subtree was generated using the clustering technique, which is based on term frequencies, IDF scores and the cosine similarity measure.

Inspecting the subtree showed that the clustering technique was effective in segmenting the domain but made poor decisions in choosing the representative or parent category for a given subcluster. Having looked at the higher level categories, we confirm that this did not produce a subtree that was semantically similar to the untangled subtree.

Generally, users who performed tasks on the generated subtree were often not confident. We observed a larger proportion of backtracking clicks with Subtree *c* than with Subtree *b*. From Figure 2, we can observe that this was noticeable in Tasks 1 and 4. In Task 1 and 4, Subtree *c* had on average over 5 times more clicks being backtracking clicks compared to Subtree *b*. This was found to be statistically significant in Task 1 but not in Task 4.

Users generally found more relevant articles in Subtree *b* than with Subtree *c*. This was particular highlighted in the general task types, that is, Tasks 1 and 4. In Task 6, users performed better using Subtree *c* than the other two subtrees. However, on inspection, we found that the key category was among the list of top level categories. We felt this biased the outcome but was not intentional. There was, however, no significant difference in performance for Tasks 2, 3 and 5. From Figure 3, we observed, that although Subtree *b* generally had more relevant articles, the result was close. Moreover, using the t-test did not yield any statistical difference between these tasks, with regards to finding relevant articles. Also, in Figure 2, there was no significant difference in the number of backtracking users made with regards to these tasks. Thus, it seems reasonable to carry out a further user study into the effects of depth, breadth and fanout with Tasks 2, 3 and 5 using Subtree *c*.

7. CONCLUSION AND FUTURE WORK

There are very few specific examples of ontology evaluations in existing literature. In this paper, we performed a task-based ontology evaluation using existing measures proposed in literature. This was based on requirements that came out of an analysis of a real world application — Wikipedia and its categories.

In the user studies which were carried out, we found that *tangledness* may be desirable in ontologies and category structures for browsing in general knowledge application areas like Wikipedia. This was especially significant in tasks that required specific information.

Also, we studied a method for generating a category structure but generally found that it was not comparable to a Wikipedia version. However, we found no significant differences in performance for some tasks. Thus for future work, we propose further studies into the effects of depth, breadth and fanout in an additional user study with these tasks.

Acknowledgements. Mingfang Wu at RMIT for her advice regarding the user experiments.

8. REFERENCES

- [1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Info. Retrieval*. ACM Press / Addison-Wesley, 1999.
- [2] C. Brewster, H. Alani, S. Dasmahapatra, and Y. Wilks. Data driven ontology evaluation. In *Proc. of Intl. Conf. on Lang. Resources and Eval.*, Lisbon, Portugal, 2004. European Lang. Resources Assoc.
- [3] J. S. Dong, C. H. Lee, H. B. Lee, Y. F. Li, and H. Wang. A combined approach to checking web ontologies. In *Proc. of Intl. Conf. on World Wide Web*, pages 714–722. ACM Press, 2004.
- [4] A. Gangemi, C. Catenacci, M. Ciaramita, and J. Lehmann. Ontology evaluation and validation. Technical report, Lab. for Applied Ontology, 2005.
- [5] A. Gómez-Pérez. Towards a framework to verify knowledge sharing technology. *Expert Systems With Applications*, 11(4):519–529, 1996.
- [6] A. Gómez-Pérez. Evaluation of ontologies. *Intl. Journal of Intelligent Systems*, 16:391–409, 2001.
- [7] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [8] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Intl. Journal Human-Computer Studies*, 43(5-6):907–928, 1995.
- [9] M. Grüninger and M. Fox. Methodology for the design and evaluation of ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI’95*, 1995.
- [10] N. Guarino. Some ontological principles for designing upper level lexical resources. In *Proc. of the 1st Intl. Conf. on Lexical Resources and Evaluation*, May 1998.
- [11] N. Guarino and C. Welty. Evaluating ontological decisions with OntoClean. *C.ACM*, 45(2):61–65, 2002.
- [12] A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proc. of Intl. Conf. on Knowledge Eng. and Knowledge Management*, 2002.
- [13] G. Marchionini. *Information Seeking in Electronic Environments*. Cambridge University Press, 1995.
- [14] D. L. McGuinness. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, chapter 6: Ontologies Come of Age, pages 171–195. MIT Press, 2002.
- [15] S. Tartir, I. Arpinar, M. Moore, A. Sheth, and B. Aleman-Meza. OntoQA: Metric-based ontology quality analysis. In *Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, 2005.
- [16] R. Thompson and W. Croft. Support for browsing in an intelligent text retrieval system. *Int. J. Man-Mach. Stud.*, 30(6):639–668, 1989.
- [17] M. Uschold and M. Grüninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.
- [18] J. Yu, J. A. Thom, and A. Tam. Evaluating ontology criteria for requirements in a geographic travel domain. In *Proc. of Intl. Conf. on Ontologies, DataBases and Applications of Semantics*, 2005.
- [19] Y. Zhao and G. Karypis. Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, 10(2):141–168, 2005.