Mechanical Turk as an Ontology Engineer?

Using Microtasks as a Component of an Ontology-Engineering Workflow

Natalya F. Noy
Stanford University
Stanford, CA 94305, USA
noy@stanford.edu

Jonathan Mortensen
Stanford University
Stanford, CA 94305, USA
jmort@stanford.edu

Paul R. Alexander
Stanford University
Stanford, CA 94305, USA
palexander@stanford.edu

Mark A. Musen
Stanford University
Stanford, CA 94305, USA
musen@stanford.edu

ABSTRACT

Ontology evaluation has proven to be one of the more difficult problems in ontology engineering. Researchers proposed numerous methods to evaluate logical correctness of an ontology, its structure, or coverage of a domain represented by a corpus. However, evaluating whether or not ontology assertions correspond to the real world remains a manual and time-consuming task. In this paper, we explore the feasibility of using microtask crowdsourcing through Amazon Mechanical Turk to evaluate ontologies. Specifically, we look at the task of verifying the subclass–superclass hierarchy in ontologies. We demonstrate that the performance of Amazon Mechanical Turk workers (turkers) on this task is comparable to the performance of undergraduate students in a formal study. We explore the effects of the type of the ontology on the performance of turkers and demonstrate that turkers can achieve accuracy as high as 90% on verifying hierarchy statements form common-sense ontologies such as WordNet. Finally, we compare the performance of turkers to the performance of domain experts on verifying statements from an ontology in the biomedical domain. We report on lessons learned about designing ontology-evaluation experiments on Amazon Mechanical Turk. Our results demonstrate that microtask crowdsourcing can become a scalable and efficient component in ontology-engineering workflows.

Author Keywords

Semantic Web, ontology, human computation, crowdsourcing, Amazon Mechanical Turk

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Miscellaneous

WebSci'13, May 1 – May 5, 2013, Paris, France.
ACM 978-1-4503-1889-1... $10.00

General Terms

Human factors, algorithms

INTRODUCTION

Ontology engineering is a labor-intensive and knowledge-intensive task. As ontologies grow larger, scientists expend even more effort developing and verifying them. Indeed, in most cases, it is impossible for a single individual to develop or to maintain the entire ontology. Large ontologies containing tens of thousands of classes, such as the Gene Ontology (GO) [8], are possible only through collaborative development. Most large-scale ontology-development projects today involve collaborative effort [26] and there are a number of collaborative ontology editors that support this process (e.g., WebProtégé [30], OntoWiki [2], and MoKi [7]). Researchers have proposed using crowdsourcing methods for other ontology-management tasks as well. For instance, developers of tools for ontology alignment used social methods to engage a community in defining correspondences between classes in different ontologies [11, 16, 20, 35]. Our own BioPortal ontology repository [34] offers a platform for crowdsourcing ontology evaluation and alignment in the biomedical and life sciences domain. It enables users to create alignments between individual elements of ontologies, as well as to comment on ontologies and to request enhancements from ontology authors [20, 1].

In all these approaches to ontology engineering or alignment, the solicitation of contributions is passive: The majority of users come to the site because they have their own task to solve; alignment specifications, suggestions for new terms, identification of problems in the ontologies, and other similar contributions are byproducts of the tasks that users are solving. In most cases, the users perform these tasks because they themselves need the ontology in their own work. There is no expected reward, other than potential community recognition. This approach, however, faces a scalability problem. For instance, there is only a handful of user-contributed mappings in BioPortal. Scientists have their own research to conduct, want to use ontologies, but do not always have the time to contribute. Thus, we are exploring the

model of "ontology management for hire," where we do not expect users to have a specific interest in the task that they are asked to perform other than the small monetary reward that they will receive.

In this paper, we evaluate the effectiveness of using a crowdsourcing platform, such as Amazon Mechanical Turk, in verifying statements in an ontology. Specifically, we address a **hierarchy-verification task**, which we define as the task of determining whether a pair of classes in a class hierarchy should be in a subclass–superclass relationship. For example, suppose an ontology contains two classes *Dog* and $Mammal$, and defines *Dog* as a subclass of $Mammal$. The following question is a hierarchy-verification task for these two classes:

Every Dog is a Mammal.

If the user confirms that this statement is TRUE, then we have a confirmation for this subclass–superclass relation from this user. Similarly, if the hierarchy-verification statement contains two classes that are not in a subclass–superclass relationship in the ontology (e.g., *Every Dog is an Insect*), the user selecting FALSE as her response will verify the lack of a hierarchical relationship between these two classes.

We focus on the hierarchy-verification task in the experiments described in this paper because in most ontologies on the Semantic Web, the class hierarchy not only constitutes the backbone of the structure, but also is the only semantic relationship between classes that ontology developers have defined. For example, we analyzed 296 public ontologies in the BioPortal repository, the largest public repository of biomedical ontologies [18]. We evaluated all ontologies in BioPortal that had at least one relation between classes defined. In 54% of these ontologies, the subclass–superclass relationship is the *only* relationship between classes. In 68% of ontologies the subclass–superclass relationships constitutes more than 80% of all relationships.

We analyze the use of crowdsourcing for the hierarchy-verification task by addressing the following research questions (Figure 1):

1. In a hierarchy-verification tasks, how do the turkers in a microtasking environment perform, relative to participants in traditional user studies?
2. Does the turkers' performance change depending on how general or specific the ontology is (upper ontology vs application ontology)?
3. Can we use microtask crowdsourcing effectively for verifying ontologies in specialized domains such as biomedicine?

Specifically, this paper makes the following contributions:

- We demonstrate that workers in a crowdsourcing platform such as Amazon Mechanical Turk perform similarly to undergraduate students in a user study on a hierarchy-verification task.
- We evaluate the effect of an ontology domain on the performance of turkers in a microtask outsourcing environment.
- We compare the performance of turkers to domain experts in a hierarchy-verification task for an ontology in biomedical domain.
- We introduce a novel workflow for ontology engineering that includes microtask crowdsourcing as one of its core components.

TURKERS VS STUDENTS: FEASIBILITY STUDY

In order to analyze the feasibility of using microtask crowdsourcing to perform the hierarchy-verification task, we designed our experiment based on a user study published by Evermann and Fang [6]. In this study, the authors presented study subjects (undergraduate students) with hierarchy-verification questions from two upper-level ontologies. Evermann and Fang used the correctness and speed of the subjects' answers to categorization questions derived from the ontology hierarchy as a way to evaluate the correspondence of the ontology with the subjects' apparent conceptual model. We used the results of the experiments that were published by Evermann and Fang and compared their results to the turkers performing exactly the same task, under the conditions that were as similar to the original experiment as possible. Specifically, we tested the following null hypothesis in our experiment:

HYPOTHESIS 1. *There is no statistically significant difference between the accuracy of turkers and students in the Evermann and Fang experiment on the hierarchy-verification task.*

Background: Evermann and Fang Experiment

Because we wanted to replicate the conditions of the original experiment as closely as possible, we not only used the journal paper that describes the experiment in detail [6], but also contacted the authors to obtain the technical report with additional details (such as training questions, order of questions).

For the experiment, Evermann and Fang used two ontologies: (1) BWW (Bunge-Wand-Weber) ontology, which is an upper ontology for business process modeling [9] and (2) SUMO (the Suggested Upper Merged Ontology), which describes general-purpose terms [19]. The authors constructed sentences for the hierarchy-verification task using the hierarchies in the two ontologies. Specifically, Evermann and Fang created 7 sequences of three terms X, Y, and Z from each ontology, such that Z is a subclass of Y and Y is a subclass of Z. They generated two sentences for verification from each of these triples: "Every Z is a Y" and "Every Z is a X". This process produced 14 TRUE statements. We present these pairs of terms in the first two columns of Table 1. Evermann and Fang also generated 14 FALSE statements

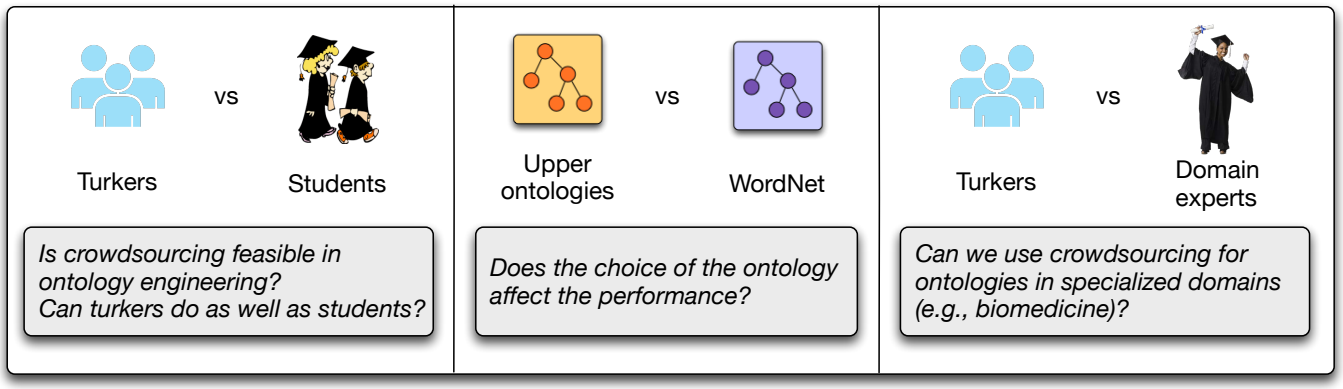

Figure 1. The roadmap for the sets of experiments that we performed.

Child	Parent	Students Correct (out of 32)	MTurk Correct (out of 32)
Ontology: Bunge-Wand-Weber ontology (BWW)			
Binding Property	Mutual Property	18 (56%)	14 (44%)
Binding Property	Property	25 (78%)	31 (97%)
Composite Event	Event	30 (94%)	31 (97%)
Composite Event	Change	21 (66%)	17 (53%)
Juxtaposition of Things	Composite Thing	18 (56%)	20 (63%)
Juxtaposition of Things	Thing	27 (84%)	22 (69%)
Lawful Event	Event	29 (91%)	31 (97%)
Lawful Event	Change	12 (38%)	12 (38%)
Non-binding Property	Mutual Property	13 (41%)	4 (13%)
Non-binding Property	Property	27 (84%)	30 (94%)
Null	Basic Thing	13 (41%)	9 (28%)
Null	Thing	14 (44%)	10 (31%)
World	Composite Thing	23 (72%)	28 (88%)
World	Thing	24 (75%)	24 (75%)
	Mean	21.1 (66%)	20.6 (63%)
Ontology: the Suggested Upper Merged Ontology (SUMO)			
Biological Process	Process	32 (100%)	31 (97%)
Biological Process	Internal Change	22 (69%)	21 (66%)
Chemical Process	Process	32 (100%)	31 (97%)
Chemical Process	Internal Change	23 (72%)	18 (56%)
Geopolitical Area	Agent	14 (44%)	7 (22%)
Geopolitical Area	Object	19 (28%)	12 (38%)
Group	Collection	32 (100 %)	31 (97%)
Group	Object	19 (28%)	14 (44%)
Sentient Agent	Agent	29 (91%)	31 (97%)
Sentient Agent	Object	16 (50%)	13 (41%)
Separation	Dual Object Process	22 (69%)	19 (28%)
Separation	Process	29 (91%)	30 (94%)
Substitution	Dual Object Process	18 (56%)	14 (44%)
Substitution	Process	29 (91%)	28 (88%)
	Mean	24.0 (75%)	21.44(67%)

Table 1. The pairs of term from the Evermann and Fang experiment that were used for the sentence verification tasks and the results for student and turkers. The table shows the data only for TRUE statements because that was the data that Evermann and Fang reported.

that contained non-hierarchically related terms from the ontologies. The question order was then randomized.

Evermann and Fang conducted within-subject experiments, recruiting 32 students using an advertisement on campus. They offered a $50 award for the best performance. Each subject was presented with 28 sentences, one sentence at a time. The subjects were asked to determine whether each sentence was TRUE or FALSE. Prior to the test experiments, the subjects were presented with twelve training questions, which also used the terms from the ontologies.

Table 1 (column 3) presents the fraction of correct responses out of 32 total responses that Evermann and Fang collected. Note that, on some questions, the response was extremely poor, with a number of questions receiving fewer than 50% of correct responses. On average, for BWW, 66% of student responses were correct; for SUMO, the number was 75%.

We relied on the results of Evermann and Fang experiments as the basis for our feasibility study on using the Amazon Mechanical Turk microtask crowdsourcing platform in hierarchy-verification tasks. We designed the MTurk experiments to replicate as closely as possible the conditions of the Evermann and Fang experiment to enable comparison of the student performance in their experiments to the turker performance in our experiments.

Turkers vs Students: Experiment setup

We published two tasks on Amazon Mechanical Turk, corresponding to the two ontologies (BWW and SUMO). Each task contained the same 28 questions for each ontology that Evermann and Fang used. The task presented a page with all 28 questions in the same order as in the original experiment. Evermann and Fang randomized the question order and we use the same order that they had. We used the same formulation of the question as in the original experiment ("Every X is a Y"). Figure 2 shows a screenshot of how the questions appeared to the turkers. We required that the turkers answer every question for the assignment to be accepted.

Training questions The turkers had to pass a *qualification test* in order to be able to take our task. This qualification test consisted of exactly the same twelve training questions that Evermann and Fang presented to their subjects. The turkers had to answer 8 out of 12 questions correctly in order to pass the qualification test. The turkers needed to pass the test only once in order to access both tasks. Thus, the turkers who completed the qualification had effectively the same training as the students in the original experiment, allowing for a valid comparison.

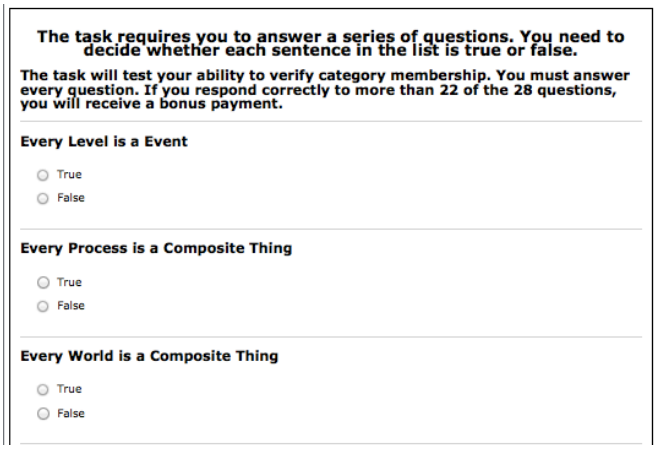

Figure 2. The turker interface for the evaluation task. The screenshot presents the first of several questions for the BWW ontology.

Monetary rewards We paid \$0.10 to each turker who completed the 28 questions. We paid the turkers immediately after task completion, regardless of the number of correct answers. The task also specified that turkers who answer more than 75% of the questions correctly will receive a bonus. We manually reviewed the responses and awarded \$0.10 bonuses to each worker who provided correct answers to 21 or more questions out of the 28.

Number of responses. For each of the two tasks, we requested that 40 turkers complete the assignment. We disqualified all the turkers who had more than 23 identical answers out of 28 (i.e., selecting TRUE or FALSE more than 23 times), removing their responses from the analysis. This filtering step allowed us to filter the turkers who appeared to have performed the task by selecting responses randomly. We then considered the first 32 of the remaining responses. When disqualifying turkers left us with fewer than 32 valid responses, we opened the task again to collect additional responses.

Measuring the performance. To determine whether there was a statistically significant difference between the responses of the students and turkers, we compared the number of correct responses for each TRUE question (Table 1, columns 3 and 4). We use an independent, unpaired, two-tailed Student's t-test. Evermann and Fang did not report the number of correct responses for FALSE questions.

Base Protocol. In the experiments reported in this paper, we based additional experiments on this initial setup. We describe modifications for each experiment when necessary.

Turkers vs Students: Results

We summarize the results in Table 1. The table reports the number of correct responses out of 32 total responses. We found that there was no significant difference between the sets of responses for both BWW and SUMO. For BWW responses we saw a $p = 0.80$ ($p > 0.05$) and SUMO showed $p = 0.38$ ($p > 0.05$). Therefore, we cannot reject the null-hypothesis that there is no statistically significant difference between the sets.

On average, students identified 66% of TRUE statements for BWW correctly. For turkers, this number was 63%. For SUMO, the students answered 75% of questions correctly on average; for turkers this number is 67%.

Turkers vs Students: Discussion

We designed this set of experiments to analyze whether the turkers can answer the hierarchy-verification questions as well as undergraduate students that Evermann and Fang recruited. The lack of any statistically significant difference between the responses confirms our hypothesis that the students and the turkers perform similarly for BWW and SUMO.

Our assignment stated that turkers will receive a bonus if they answered more than 75% of the questions correctly. First, researchers showed that promise of a bonus for correct answers improves the quality of the responses [32]. Second, Evermann and Fang offered a \$50 award to a student with the best result. Our promise of a bonus replicated that incentive.

While we tried to replicate the Evermann and Fang experiment as closely as possible, we could not avoid several differences. In their experiment, each subject saw only one question at a time. In our experiment, each turker saw all questions for the same ontology on the same screen. In order to understand the effects of this difference in the experiment setup, we ran a pilot study where we presented the turkers with one hierarchy-verification question at a time (at \$0.01 per question). We found that the correctness of the results was essentially the same as in the design that we report in this paper. However, this experiment does not fully replicate the Evermann and Fang design either: In this setting, we cannot require that each turker answers all questions. We also cannot control the order in which the turkers answer the questions. Thus, we used the 28-questions-per-page design in all our experiments in this study.

UPPER ONTOLOGIES VS WORDNET

Our feasibility study demonstrated that turkers achieve performance similar to that of undergraduate students on hierarchy-verification tasks. The Evermann and Fang experiments focused on upper ontologies—and demonstrated that these ontologies are difficult for non-experts to verify. In the following experiment, we analyze the effect of the ontology on the performance of the turkers. We used WordNet in order to evaluate the performance that we can achieve with turkers on an ontology that covers common-sense knowledge. Specifically, we test the following hypothesis:

HYPOTHESIS 2. *Turkers perform better on the hierarchy-verification task for a common-sense application ontology than for an upper ontology.*

Ontology: WordNet		
Child	Parent	Correct (out of 32)
TRUE statements		
Antitrust legislation	Law	24 (75%)
Antitrust legislation	Legal document	24 (75%)
Bacteria	Microorganism	28 (88%)
Bacteria	Organism	29 (91%)
Civil engineer	Engineer	31 (97%)
Civil engineer	Person	31 (97%)
Computer	Device	32 (100%)
Computer	Machine	30 (94%)
Pebble	Natural object	22 (69%)
Pebble	Rock	23 (72%)
Seawater	Liquid	31 (97%)
Seawater	Water	32 (100%)
Weather satellite	Equipment	31 (97%)
Weather satellite	Satellite	29 (91%)
FALSE statements		
Assembly	Natural object	29 (91%)
Assembly	Rock	30 (94%)
Distilled water	Microorganism	28 (88%)
Distilled water	Organism	28 (88%)
License	Equipment	28 (88%)
License	Satellite	31 (97%)
Plant tissue	Liquid	22 (69%)
Plant tissue	Water	23 (72%)
Programmer	Law	29 (91%)
Programmer	Legal document	29 (91%)
Telecommunication system	Engineer	31 (97%)
Telecommunication system	Person	30 (94%)
Virus	Device	30 (94%)
Virus	Machine	31 (97%)
	Mean	28.4 (89%)

Table 2. The pairs of term for the sentence verification tasks for WordNet. The table shows the data both for TRUE and FALSE statements.

Upper ontologies vs WordNet: Experiment setup

We generated a set of questions using the hierarchy in WordNet. Just as in the case of the upper ontologies, we extracted sequences of three terms, which were in a direct hierarchical relationship to each other. We intentionally selected what appeared to be very common-sense sequences of terms. Our goal for this set was to determine how well the turkers would do when given a set of relatively easy subsumption statements in order to test our hypothesis. We followed the base protocol wherein we used upper ontology qualification questions. Table 2 (columns 1 and 2) has the terms for TRUE and FALSE statements for WordNet. We presented these statements to the turkers in a randomized order.

Upper ontologies vs WordNet: Results

Table 2 (column 3) reports the number of correct responses out of 32 total responses. Figure 3 compares the results for correct responses on all questions (both TRUE and FALSE statements) for all the turkers for the three ontologies (BWW, SUMO, and WordNet). The blue bars show the percentage of correct responses. The performance is the worst for BWW (59%) and the best for WordNet (89%). Given that each turker had to answer 28 questions, answering 19 questions correctly constituted a statistically significant result for a turker ($\alpha = 0.05$ via the hypergeometric distribution). In other words, if a turker answered 19 or more questions correctly, there is less than 5% chance that the responses were random. The red bars in Figure 3 show the percentage of turkers whose results were unlikely to be random (i.e., statistically different from random). The results differ from only 5 out of 32 for BWW, to almost all (31 out of 32) turkers for WordNet. The green bars show the percentage of turkers who answered more than 75% of the questions correctly in any given set and received a bonus. The difference between the ontologies here is even more pronounced: for the BWW ontology, only one turker received a bonus; for WordNet, 29 turkers out of 32 did.

Upper ontologies vs WordNet: Discussion

Our results for the turkers performance for different ontologies indicate that the "simpler" the terms in the ontology, the better the turkers do. Our WordNet experiment intentionally used common-sense terms and we were able to achieve the accuracy of 89% in the turkers responses. This high percentage indicates that, in general, the Amazon Mechanical Turk platform may be appropriate for verification tasks for ontologies that are designed to reflect general knowledge.

ANATOMY ONTOLOGY: TURKERS VS DOMAIN EXPERTS

Our final set of experiments addresses the following question: Can turkers perform hierarchy-verification tasks efficiently in a specialized domain, such as biomedicine? Specifically, we test the following hypothesis:

HYPOTHESIS 3. *Turkers perform as well, or similar to, experts in the domain on hierarchy-verification tasks.*

We chose the Common Anatomy Reference Ontology (CARO) [10] ontology for this experiment: it is a small ontology, with 49 classes. It represents the basics concepts in human anatomy. Every class in CARO has a textual definition. For the evaluation, we compared the accuracy of responses from turkers to that of domain experts.

In this experiment, we also measured the effect of providing context for the terms by providing a textual definition if one exists in the ontology.

Turkers vs Domain experts: Experiment setup

We follow a similar protocol to previous experiments to collect responses from the turkers. To create the hierarchy pairs, we select pairs of ontology terms from CARO that are hierarchically related, and also pairs that are not. We perform two verification tasks on these pairs: (1) standard hierarchy verification, and (2) hierarchy verification with the addition of term definitions. In this experiment, we also implement a biology related qualification test. We used a set of questions from a high-school biology test as our qualification questions.

Because asking domain experts to set up an account on Amazon Mechanical Turk would increase the barrier to their participation, we performed a modified experiment with them. We reached out to our collaborators through social-media channels of the National Center of Biomedical Ontology (NCBO) and mailing lists such as

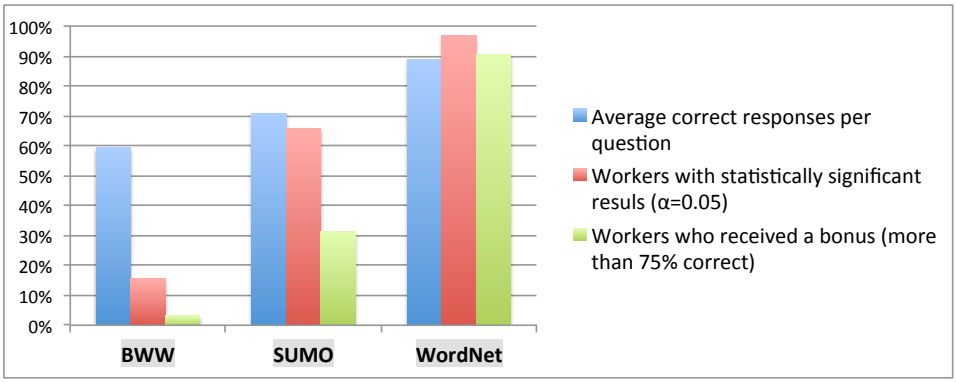

Figure 3. Comparison of the results for the three ontologies in the study. The data is for 32 qualified respondents.

Ontology: CARO	
Child	Parent
TRUE statements	
extraembryonic structure	anatomical structure
simple cuboidal epithelium	unilaminar epithelium
portion of tissue	anatomical structure
anatomical structure	material anatomical entity
multi-cell-component structure	anatomical structure
unilaminar epithelium	epithelium
hermaphroditic organism	multi-cellular organism
protandrous hermaphroditic organism	sequential hermaphroditic organism
sequential hermaphroditic organism	hermaphroditic organism
anatomical point	immaterial anatomical entity
anatomical line	immaterial anatomical entity
acellular anatomical structure	anatomical structure
compound organ component	multi-tissue structure
male organism	gonochoristic organism
FALSE statements	
organism subdivision	female organism
asexual organism	multi-cell-component structure
portion of tissue	anatomical space
acellular anatomical structure	simple organ
single cell organism	epithelial cell
compound organ	cell component
male organism	acellular anatomical structure
female organism	cavitated compound organ
portion of cell substance	simple columnar epithelium
basal lamina	anatomical surface
anatomical cluster	sequential hermaphroditic organism
anatomical point	material anatomical entity
neuron projection bundle segment	solid compound organ
extraembryonic structure	hermaphroditic organism

Table 3. The pairs of term for the sentence verification tasks for CARO with definitions. The table shows the data both for TRUE and FALSE statements.

the *obo-anatomy* mailing lists. These channels mostly reach those who are interested in biomedical ontologies.

We created a standard online web survey using SurveyMonkey. We asked each domain expert to answer the same 28 questions that the turkers did and we presented them with a very similar form. To provide incentive, we awarded a randomly selected participant with a gift card. We randomly directed the domain experts either to the survey that presented term definitions along with the term names or to the one that displayed only the names of the terms. We collected 11 and 14 responses for these tasks, respectively. Due to time constraints, it was unreasonable to obtain 32 responses for each task. To evaluate differences between the turkers and experts, we again use a Student's t-test. In addition, we asked those who participated in the survey what their level of expertise in biomedical ontologies was.

Mean Accuracy of Turkers vs Experts			
	Turkers	Experts	p-value
No term definitions	0.667	0.812	0.0008*
With term definitions	0.818	0.885	0.1858

Table 4. Average accuracy of turkers and experts when validating hierarchical relations in CARO, an anatomy ontology. *significance at $p < 0.05$ via Student's t-test

Turkers vs Domain experts: Results

All the participants in the domain-expert survey self-identified as experts in at least two of the following three categories: ontologies, anatomy ontologies, biomedicine.

Table 4 compares the accuracy of domain experts and turkers completing a hierarchy-verification task for CARO. We present results both for the task that contained definitions and the task that did not. Recall that turkers needed to pass a qualification test by answering correctly 8 out of 12 high-school-level biology questions. Turkers, even with a qualification test, perform with an accuracy of 66% when they did not see the textual definitions of the terms. In comparison, experts achieve an accuracy of 81%, significantly better than turkers ($p < 0.05$). When given definitions, turkers are correct 82% of the time. However, experts again perform better, with an accuracy of 89%. The difference between turkers and domain experts in the second task was not statistically significant. Thus, we cannot reject the null-hypothesis that turkers perform as well as experts ($p > 0.05$).

Turkers vs Domain experts: Discussion

In our experiments, domain experts performed better than the turkers in verifying ontology statements. However, the difference was significant only when we did not provide the context in the form of textual class definitions. It is likely that domain experts have general domain knowledge and understand what the ontology might be about (they were aware that they were invited because of their expertise in biomedical ontology). Turkers did not have such context.

Our results demonstrate a significant effect of providing context, even for the experts in biomedical ontologies.

We believe that respondents may have had general domain knowledge but may not have known the specific terms in the ontology.

We analyzed the correlation between experts and turkers on specific questions for the task where we provided definitions. On average, experts with provided definitions had 0.07 higher accuracy than qualified turkers who were also provided definitions (Table 4, last row). In order to determine between turkers and experts significantly deviated from this mean, we looked for questions where this difference was more than one standard deviation from the mean. On 23 of the 28 questions, experts had accuracy difference within one standard deviation (-0.02,0.16) of the mean. On 4 questions, turker accuracy was more than 0.16 lower than experts. The turkers' accuracy was greater than experts by at least 0.02 on one question, *"Compound organ component is a kind of multi-tissue structure."* All turkers answered this question correctly, whereas some experts answered the questions incorrectly (TRUE). We found that the definitions for the concepts in this question clearly described that the relationship between them is indeed true. While turkers likely used these definitions alone, experts may have used their background knowledge when evaluating this relationship and disagreed with it, thus accounting for their lower performance.

The fact that the difference between turkers and domain experts, provided they both saw the definitions, was not statistically significant provides initial evidence that the use of turkers even in specialized domains may be feasible. Indeed, in our study, we found it difficult to obtain a reasonable number of experts to perform verification. In a large-scale ontology-engineering project, it is even more difficult to have multiple experts verify the ontology. Therefore, while the turkers did not perform as well as experts, they could still provide a scalable resource that would likely assist in identifying "hot-spots" or problem area in an ontology hierarchy.

GENERAL DISCUSSION AND LESSONS LEARNED

The set of experiments that we described in this paper provided evidence and analysis on the use of microtask crowdsourcing platform such as Amazon Mechanical Turk for ontology-verification tasks. Our results suggest that crowdsourcing ontology verification not only is feasible but also can be useful in specialized domains. In this section, we discuss two aspects of our experiments: our vision for integrating this type of crowdsourcing in an ontology-engineering and management process and the lessons learned on applying crowdsourcing to ontology-engineering tasks. We start with the discussion of related work.

Related Work

Researchers successfully used human computation in a number of complex tasks [22]. For example, they used so-called "games with a purpose" for tasks ranging from image tagging [31] to ontology alignment [29] and identity resolution [14]. Researchers developed special-purpose platforms such as Zooniverse [23] to involve citizen scientists in solving complex scientific problems and fold.it [4] to allow anybody to predict structures of proteins. Scientists successfully deployed microtask crowdsourcing using such platforms as Amazon Mechanical Turk for user studies [12] and text editing [3].

Recent studies considered crowdsourcing for management of structured and linked data. ZenCrowd, for example, combines the results of automatically generated answers with the answers by turkers in order to link entities recognized in a text with entities in the Linked Open Data cloud [5]. Simperl and colleagues discuss the use of crowdsourcing for querying semantic data [27]. Sarasua and colleagues studied the use of microtask crowdsourcing to improve ontology alignment [24].

To our knowledge, the work reported in this paper is the first attempt to use crowdsourcing in a subtask in ontology evaluation and engineering, such as hierarchy verification. Thus, this work complements both other studies on using collaboration in ontology engineering and in applying crowdsourcing to enhancing structured and linked data.

Using microtask crowdsourcing in ontology management

Imagine an ontology engineer uses a tool, such as Protégé [30], to develop an ontology or to explore an ontology that her collaborators developed. There is a portion of the hierarchy that she needs verified and does not have either the time or the knowledge to perform the verification. Our experiments demonstrate that it is feasible to invoke microtask crowdsourcing for this task. We envision a workflow that would enable the user to invoke such a module by identifying (1) a portion of her ontology that needs to be verified; (2) the field of expertise that she expects the turkers to have (or a set of qualification questions if her domain is specialized); (3) the amount that she is willing to pay; (4) the level of agreement or certainty in the responses (e.g., she may want the turkers only to flag problematic areas rather than to suggest correct modeling solutions); and (5) other configuration parameters if necessary. The crowdsourcing component of a tool like Protégé would then create and post the corresponding tasks on a microtasking platform, collect and analyze results and present the results to the requester.

There are many research questions that we still need to answer to transform this vision into a pragmatic versatile solution. For example, what are other tasks that are amenable to microtask crowdsourcing. Some of the examples that we considered include ontology alignment, generation of natural-language class definition from automatically generated ones, verification of ontology statements that we can learn from data or through information extraction from text. For each of these tasks, we will need to determine the appropriate granularity of questions, formulation of the questions,

and ways to combine the results. For instance, in using crowdsourcing to verify results of automatic methods such as ontology learning from data, we may want to factor in the certainty factors provided by the learning method: if we are using the crowd to verify a statement to which an automatic method has assigned high certainty factor, we may need less redundancy than when verifying a fact that has less support in the data. Furthermore, in order to enable the user to decide which resources and constraints are important to her, we can incorporate models for balancing time and budget constrains and for optimizing the trade-off between these resources proposed by other researchers [15, 17].

Based on our experiments so far, we believe that introducing crowdsourcing into an ontology-engineering workflow in a principled way, however, may considerably increase productivity and scalability of ontology-engineering.

Designing experiments on Amazon Mechanical Turk

Kittur and colleagues describe the significant change in the performance of turkers with only minor modifications in experimental configuration [12]. Similarly, in psychology studies researchers consistently demonstrated that participant selection, priming, question tone, and context all affect performance and reliability [25, 28]. To explore this issue, we performed additional experiments to evaluate the effect of two key parameters in our microtask configuration: (1) qualification questions that turkers must answer in order to gain access to the task; and (2) the formulation of the hierarchy-verification question [21].

Qualification questions Qualification questions can serve three purposes. First, they help filter out spammers, by requiring turkers to answer "free" questions before gaining access to the paid questions. Second, they determine wether a turker is capable of answering hierarchy-verification questions. And finally, they provide the turkers with initial training, familiarizing them with the type of questions that they will need to answer. We found that qualification questions do indeed improve the quality of responses, although not significantly (by 8 to 17%, depending on the task). However, it appears that having more difficult questions does not have a measurable effect on performance (and in some cases, we observed that the accuracy was lower). However, whether or not we had qualification questions had a dramatic effect on the time that it took us to collect the responses. In all cases, not having qualification questions produced the required number of responses in a matter of minutes. With qualification questions, regardless of whether they were simple or not, we had to wait between 3-4 days to a week or two before we obtained the required number of responses.

Question formulation We performed a number of experiments to determine the effect of question formulation on the accuracy of the results. We varied the question grammatical polarity as positive or negative (e.g., *Every computer is a machine* vs *Not every computer is a machine*) and posed questions in the mood of either interrogative (YES or NO) or indicative (TRUE or FALSE). We observed statistically significant difference in accuracy of responses based on the formulation of the questions. We achieves the highest accuracy with a question of a positive polarity in the indicative mood (e.g., *Computer is a kind of machine*, *Every computer is a kind of machine*). We introduced negative questions (e.g., *Not every computer is a machine*) in an attempt to direct the turkers to think of exceptions to a hierarchical relation. However, it appears that the additional cognitive load of a complex question reduced performance. These observations indicate that it is worthwhile to perform pilot experiments for new types of tasks to determine the most optimal configuration.

Amount paid For our first experiment, recapitulating the Evermann and Fang experiment, we paid a total $16.20 for completing the assignments for the first experiment (in some cases, we got more responses than we needed) and $5.50 in bonus payments. Thus, the experiment cost us $21.70, much less than the original experiment cost Evermann and Fang (they paid a $50 bonus to the best-performing student). The relatively low cost of these experiments indicates the potential scalability of our method.

Redundancy in responses In the experiments that we describe in this paper, we compare our data to the Evermann and Fang experiment. Thus, we used the same number of responses in each of the experiment as they did: 32 qualified responses. In practice, you do not need that much redundancy to achieve the accuracy that we achieved. Indeed, we achieve essentially identical accuracy of responses for all three ontologies by considering only the first 8 (25%) of the 32 responses. Several researchers have recently studied the optimal number of responses in crowdsourced tasks [33, 13]. We plan to use the results of these studies as we look at integrating crowdsourcing steps in ontology-enginerring workflows.

Disqualifying responses. As we noted earlier, we disqualified the responses for each set, where the turker had the same response for 23 out of 28 questions, assuming that these were spammers. We disqualified 10 responses for BWW; 4 responses for SUMO; and 9 for WordNet. In other words, we needed between 12% and 31% more responses in order to receive the needed number of qualified responses. Note that for BWW, some of the 10 turkers that we identified as spammers might simply have had trouble answering the questions correctly, given the generally poor performance of both the students and the turkers on this task. We chose the threshold of 23 out of 28 empirically by analyzing responses for WordNet: there was a significant difference in the quality of responses that were above this threshold. In future work, we plan to analyze the effects of different thresholds on the results.

Identifying spam As other researchers who have used the Amazon Mechanical Turk platform to perform user studies have observed, a certain amount of spam is inevitable [12]. We controlled for malicious behavior in several ways: First, turkers had to obtain the qualification to access our tasks by answering correctly 8 out of 12 training questions. Second, we have requested that turkers have at least a 90% approval rating from other requesters, thus selecting the turkers with good reputations. Third, as other researchers have shown, making a correct answer as time consuming as a wrong answer reduces the spam level significantly [12]. In our case, the turkers had to select 28 true or false answers. Thus, they had to spend the time to click 28 times, which required sufficient amount of effort. The incremental effort to read the sentences as well was not large. Finally, we have filtered out the responses that appeared to have been done randomly, with 23 out of 28 identical responses.

While qualification tests serve as one barrier for filtering out unqualified turkers, our experience shows that these tests alone are not sufficient. For instance, we observed that, even after passing the qualification tests, 10% of turkers still provided identical responses to all the questions in their task—a clear indication of unusable responses. We need to analyze a number of ways to filter spam such as presenting "golden questions"—easy questions for which we know the answer in advance—inconspicuously among the microtasks presented to the same turker; filtering questions from a turker who fails to meet a specific threshold (e.g., certain diversity of answers); requiring textual answers to ascertain that the turker has analyzed the input data before requesting an answer to a multiple-choice question; and so on.

CONCLUSIONS

Our results demonstrate that microtask crowdsourcing can serve as an effective step in an ontology-engineering workflow. The performance of the turkers did not have any statistically significant difference from the performance of students in a traditional user study. Furthermore, our results showed that turkers can achieve very high levels of accuracy (close to 90%) on verifying hierarchy statements for common-sense ontologies, such as WordNet. In assessing the turkers' performance on domain-specific ontologies, such as CARO, we have demonstrated that turkers who pass qualification tests can answer correctly up to 81% of the questions—fewer questions than domain experts. However, for the applications where such performance is acceptable, micro tasking provides a far more scalable alternative than relying on domain experts. In summary, our experiments demonstrate that crowdsourcing approach can become a feasible scalable component in an ontology-engineering workflow.

Acknowledgements

We are grateful to Matthew Horridge, Jamie Taylor, Dimitri Vaynblat and Anna Bauer-Mehren for their suggestions on the experiment help with statistical analysis. Joerg Evermann was extremely helpful in providing the details of the original experiment.

REFERENCES

1. Alexander, P. R., Nyulas, C. I., Tudorache, T., Whetzel, T., Noy, N. F., and Musen, M. A. Semantic infrastructure to enable collaboration in ontology development. In *International Workshop on Semantic Technologies for Information-Integrated Collaboration (STIIC 2011)* (Philadelphia, PA, USA, 2011).
2. Auer, S., Dietzold, S., and Riechert, T. OntoWiki–a tool for social, semantic collaboration. In *Fifth International Semantic Web Conference, ISWC*, vol. LNCS 4273, Springer (Athens, GA, 2006).
3. Bernstein, M., Little, G., Miller, R., Hartmann, B., Ackerman, M., Karger, D., Crowell, D., and Panovich, K. Soylent: a word processor with a crowd inside. In *The 23d annual ACM symposium on user interface software and technology*, ACM (2010), 313–322.
4. Cooper, S., Khatib, F., Treuille, A., Barbero, J., Lee, J., Beenen, M., Leaver-Fay, A., Baker, D., and Popovi?, Z. Predicting protein structures with a multiplayer online game. *Nature 466*, 7307 (2010), 756–760.
5. Demartini, G., Difallah, D. E., and Cudr-Mauroux, P. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *21st World Wide Web Conference WWW2012* (Lyon, France, 2012), 469–478.
6. Evermann, J., and Fang, J. Evaluating ontologies: Towards a cognitive measure of quality. *Information Systems 35* (2010), 391403.
7. Ghidini, C., Kump, B., Lindstaedt, S., Mahbub, N., Pammer, V., Rospocher, M., and Serafini, L. Moki: The enterprise modelling wiki. In *European Semantic Web Conference (ESWC-2009)*, Springer Berlin / Heidelberg (Heraklion, Greece, 2009), 831835.
8. GOConsortium. Creating the Gene Ontology resource: design and implementation. *Genome Res 11*, 8 (2001), 1425–33.
9. Green, P., and Rosemann, M. Integrated process modeling: An ontological evaluation. *Information Systems 25*, 2 (2000), 73–87.
10. Haendel, M., Neuhaus, F., Osumi-Sutherland, D., Mabee, P., Mejino, J., Mungall, C., and Smith, B. Carothe common anatomy reference ontology. *Anatomy Ontologies for Bioinformatics* (2008), 327–349.
11. Hausenblas, M., Troncy, R., Raimond, Y., and Brger, T. Interlinking multimedia: How to apply linked data principles to multimedia fragments. In *WWW 2009 Workshop: Linked Data on the Web* (2009).
12. Kittur, A., Chi, E., and Suh, B. Crowdsourcing user studies with Mechanical Turk. In *26th annual SIGCHI conference on human factors in computing systems* (2008), 453–456.
13. Lin, C. H., Mausam, and Weld, D. S. Dynamically switching between synergistic workows for crowdsourcing. In *Twenty-Sixth AAAI Conference on Artificial Intelligence* (2012).
14. Markotschi, T., and Völker, J. GuessWhat?! - Human Intelligence for Mining Linked Data. In *Proceedings of the Workshop on Knowledge Injection into and Extraction from Linked Data at EKAW* (2010).
15. Mason, W., and Watts, D. Financial incentives and the "Performance of Crowds". In *ACM SIGKDD workshop on human computation*, ACM (2009), 77–85.
16. McCann, R., Shen, W., and Doan, A. Matching schemas in online communities: A Web 2.0 approach. In *The 24th International Conference on Data Engineering (ICDE-08)* (Cancun, Mexico, 2008).

17. Minder, P., Seuken, S., Bernstein, A., and Zollinger, M. Crowdmanager-combinatorial allocation and pricing of crowdsourcing tasks with time constraints. In *Workshop on Social Computing and User Generated Content in conjunction with ACM Conference on Electronic Commerce (ACM-EC 2012)* (Valencia, Spain, 2012), 1–18.
18. Musen, M. A., Noy, N. F., Shah, N. H., Whetzel, P. L., Chute, C. G., Storey, M.-A., Smith, B., and team, T. N. The national center for biomedical ontology. *Journal of American Medical Informatics Association 19* (2012), 190–195.
19. Niles, I., and Pease, A. Towards a standard upper ontology. In *The 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)* (Ogunquit, Maine, 2001).
20. Noy, N. F., Griffith, N., and Musen, M. A. Collecting community-based mappings in an ontology repository. In *7th International Semantic Web Conference (ISWC 2008)* (Karlsruhe, Germany, 2008).
21. Noy, N. F., Mortensen, J., Alexander, P. R., and Musen, M. A. Ontology engineering through microtask crowdsourcing. *Under review* (2013).
22. Quinn, A., and Bederson, B. Human computation: a survey and taxonomy of a growing field. In *Annual Conference on Human Factors in Computing Systems (CHI 2011)*, ACM (Vancouver, BC, 2011), 1403–1412.
23. Raddick, M., Bracey, G., Gay, P., Lintott, C., Murray, P., Schawinski, K., Szalay, A., and Vandenberg, J. Galaxy zoo: exploring the motivations of citizen science volunteers. *arXiv preprint arXiv:0909.2925* (2009).
24. Sarasua, C., Simperl, E., and Noy, N. F. Crowdmap: Crowdsourcing ontology alignment with microtasks. In *11th International Semantic Web Conference (ISWC)*, Springer (Boston, MA, 2012).
25. Schwarz, N. Self-reports: How the questions shape the answers. *American Psychologist 54*, 2 (1999), 93–105.
26. Sebastian, A., Noy, N. F., Tudorache, T., and Musen, M. A. A generic ontology for collaborative ontology-development workflows. In *16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008)*, Springer (Catania, Italy, 2008).
27. Simperl, E., Norton, B., and Vrandecic, D. Crowdsourcing tasks in linked data management. In *2nd workshop on consuming Linked Data COLD2011 co-located with the 10th International Semantic Web Conference ISWC 2011* (Bonn, Germany, 2011).
28. Tanur, J. M. *Questions about Questions: Inquiries Into the Coqnitive Bases of Surveys*. Russell Sage Foundation Publications, 1992.
29. Thaler, S., Siorpaes, K., and Simperl, E. SpotTheLink: A Game for Ontology Alignment. In *6th Conference for Professional Knowledge Management* (2011).
30. Tudorache, T., Nyulas, C., Noy, N. F., and Musen, M. A. Webprotégé: A distributed ontology editor and knowledge acquisition tool for the web. *Semantic Web Journal 11-165* (2011).
31. von Ahn, L., and Dabbish, L. Labeling images with a computer game. In *SIGCHI conference on Human factors in computing systems*, ACM Press New York, NY, USA (2004), 319–326.
32. Wang, J., Ghose, A., and Ipeirotis, P. Bonus, disclosure, and choice: What motivates the creation of high-quality paid reviews? In *Thirty Third International Conference on Information Systems (ICIS)* (Orlando, FL, 2012).
33. Waterhouse, T. P. Pay by the bit: an information-theoretic metric for collective human judgment. In *Conference on Computer supported cooperative work (CSCW)*, ACM (2013), 623–638.
34. Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyulas, C. I., Tudorache, T., and Musen, M. A. Bioportal: Enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic Acids Research (NAR) 39*, Web Server issue (2011), W541–5.
35. Zhdanova, A., and Shvaiko, P. Community-driven ontology matching. In *3rd European Semantic Web Conference* (Budva, Montenegro, 2006), 3449.