2010

Ontological Services Using Crowdsourcing

Huairen Lin
University of Sydney, huairen.lin@sydney.edu.au

Joseph Davis
University of Sydney, joseph.davis@sydney.edu.au

Ying Zhou
University of Sydney, ying.zhou@sydney.edu.au

Ontological Services Using Crowdsourcing

Huairen Lin
Joseph Davis
Ying Zhou
School of Information Technologies
The University of Sydney
Sydney, Australia
Email: [firstname.lastname]@sydney.edu.au

Abstract

This paper develops a service for ontology evolution based on crowdsourcing. The approach is demonstrated using OntoAssist, a specially designed semantic search service that is capable of capturing and disambiguating user's search intent as well as automatically enabling ontology evolution. Successful and consistent ontology evolution often requires large amount of input data to specify new terms or changes in relationships. These inputs typically come mainly from domain experts or ontology professionals, which makes it hard to keep up with the change of open, dynamic World Wide Web environment. By integrating OntoAssist with an existing search engine, we show that users' search intent can be disambiguated and aggregated to help to evolve underlying ontology. The disambiguation feature helps the users to find desirable search results. OntoAssist has been implemented and tested by Turkers from Amazon Mechanical Turk in a live demonstration site. Promising results and analysis are reported.

Keywords

Service, Ontology Evolution, Crowdsourcing, Human Computation, Search Assist

INTRODUCTION

With the widespread use of ontology as an integral part of semantic web applications, ontology evolution has become an important problem in ontology research. New terms and changes in relationships between terms need to be identified and incorporated to reflect the web dynamics, such as change of domain knowledge and usage perspective (Noy et al. 2004). The ontology used by semantic search application needs to be constantly updated to ensure effective information retrieval.

The task of ontology evolution is primarily carried out by domain experts or ontology professionals. However, in the highly open, dynamic World Wide Web environment, it is difficult for them to track and respond to those changes in a timely manner (Braun et al. 2007). Moreover, the ontologies maintained by them may not fit the needs of online applications, since the actual online users are usually not able to participate in the evolution process and have no control over the resulting ontology. Thus, the efficacy and value of the ontology-based application will be limited. For instance, the keywords submitted by users may have poor match with terms in ontology and result in poor search recall and precision. The limited involvement of users in the process of semantic content creation and maintenance has been recognized as one of the key reasons that restrict the more widespread adoption of semantic technologies (Siorpaes et al. 2010).

While the most sophisticated computational techniques cannot substitute the participation of domain experts, the recently proposed crowdsourcing method provides new ways to resolve some of the difficult problems in ontology refinement and evolution. The crowdsourcing method has the potential to aggregate the knowledge and intelligence of a large number of online users through a mass collaboration technique. It harnesses the collective knowledge and intelligence of a vast number of individuals to offer solutions to the problem, and the winning ideas are typically awarded some form of reward (Brabham 2008).

In this paper, we present OntoAssist: a semantic navigation support service for evolving ontology through the integration with widely used web application, such as Yahoo! Search engine[1]. It is based on a crowdsourcing model that tackles the ontology evolution problems by utilizing knowledge from large number of actual web users. It allows users to refine their searches on all kind of web resources by a few simple clicks to specify relationships between the query keyword and relevant terms. The initial ontology can thus evolve and help provide better search results to users along with the evolving domain knowledge. We make following three primary contributions:

[1] http://search.yahoo.com

(i.) We propose a crowdsourcing service facilitating ontology evolution that emphasizes gaining knowledge on the strength of semantic search intent behind common query sessions.

(ii.) We propose a design that embeds ontology evolution seamlessly in public users' daily search activities. With this design, our proposed crowdsourcing approach can get actual users involved without necessarily offering monetary reward.

(iii.) We describe a crowdsourcing evaluation method utilizing crowdsourcing websites such as Amazon Mechanical Turk (MTurk). Recruiting experimental subjects in traditional experiment can be time-consuming and costly. We demonstrate an experimental design that is able to attract hundreds of users to work on the evaluation in only several hours with less cost.

The rest of the paper is organized as follows: In the next section, we describe the concept of crowdsourcing model, discuss other relevant genre of human computation approach such as "game with a purpose", and compare their advantage and weakness in detail. In section 3, we review related research, including both traditional method and recently proposed collaborative or crowd based efforts. In section 4 and 5, we present our crowdsourcing model for evolving ontology and discuss the detailed implementation with Yahoo! search engine. Then in section 6, we describe experiments that conducted on OntoAssist with users from MTurk. Results are presented and evaluated. Finally, we discuss our conclusion and future work.

HUMAN COMPUTATION AND CROWDSOURCING

Human computation is one of the main methods to channel public users for a specific purpose. It has proved its strength in distributed problem solving and it often performs better than traditional approaches due to the typically human capabilities such as acute visual perception and aesthetic judgment (Dawkins et al. 1991). It turns out that basic conceptual intelligence and perceptual capabilities that most human take for granted (Von Ahn 2007) are critically important and useful in speech recognition and natural language analysis (Gentry et al. 2005). Although humans have an innate ability to gather and analyse data, they tend to make decisions motivated by self-preservation and on their selfish needs. Therefore, the design of certain incentives is vital to attract human to be part of a collective computation process (Von Ahn 2007). A computer program that can attract human's interest, fulfil their needs, and collect, interpret human's solution is also important.

One genre of human computation applications, known as "game with a purpose", was proposed by Luis von Ahn to enable humans to solve problems within a gaming context where a game is designed as the platform and fun of gaming as the motivation process (Von Ahn 2007). Applications designed under this consideration are expected to collect knowledge from people playing computer games, such as Google image labeler[2] . It is a feature from Google which allows two randomly paired online users to provide many labels for the same image. They get points when two labels from them are matched. Both cases enrich the textual description of images which in turn improve the accuracy of image retrieval. People contribute to it not because they care or even know about image retrieval, but because they enjoy the game.

Another genre of human computation applications is known as the crowdsourcing model. Playing games can be a good incentive for some people, but not for the majority of the web users. Crowdsourcing model describes a process of organizing users and tapping their knowledge/intelligence to complete assigned tasks. The contributing users could be rewarded for their efforts (Howe 2006). It enables the aggregation of ideas and stimulates a certain kind of innovation from numbers of users (Brabham 2008). Not limited in monetary rewards, free service such as a successful login procedure can also be used as a motivation. For instance, reCAPTCHA project (Von Ahn et al. 2008) improved the process of digitize old printed material by challenging users to decipher scanned words from books that could not be read by OCR software. The segmented words were presented as a part of CAPTCHA test (a widespread security measures on the web that prevents automated programs from abusing online services). Users were required to transcript the words appears at the registration forms as a test solution to complete a login procedure. This text transcription method has achieved high word accuracy almost at the level of professional human transcribers, and illustrated it as a fast transcription technique.

RELATED RESEARCH

Ontology is the enabling technology in semantic web applications. It provides a specification of conceptualization of a domain of interest to facilitate successful information exchange, sharing, or communication between different agents (Gruber 1993). One of the most important issues in the development and maintenance of ontology is ontology evolution. Like any structure that holding information, ontology need to change in response to a certain change in the domain or its conceptualization simply because the world has changed (Flouris et al. 2006; Stojanovic et al. 2003). The relevant new domain vocabulary or changed relationships are usually collected by the

[2] http://images.google.com/imagelabeler/

informal and formal specification of competency questions (Siorpaes et al. 2010). Despite that current ontology evolution tools have provided many nice feature, such as collaborative editing, change propagation, most of them are based on the efforts of the expertise or their developers (Flouris et al. 2006). Actual users are usually not able to participate in the ontology maintenance process. This can lead to two problems, first, it is expensive to involve ontology engineering specialists and second, concepts may have already become obsolete by the time they are collected and incorporated into the ontology (Braun et al. 2007).

As part of the efforts to achieve the wide adoption of semantic web, there is pressing need and growing interest in increasing end users' involvement in ontology maintenance. Since ontology needs to reflect a common view on the domain of interest shared by many users, the most sustainable way to maintain ontology should be through nature community efforts (Siorpaes et al. 2008). In order to reduce the barriers for the participation of web users, Wiki has been proposed as a community environment for the creation and maintenance of ontology. With the use of wiki tool, users can add element to the ontology and refine or modify it based on their opinion (Hepp et al. 2006). Human agreements on the evaluation of terms and relationships can also be helpful to ontology maintenance. In InPhO project (Eckert et al. 2010), users rated the semantic relationships generated based on semantic similarity algorithm. The final ontology was built based on their assessment. Relationships with low agreements were excluded. In a bookmarks navigation system (Limpens et al. 2009), users also helped to validate, reject or correct the automatic suggestions of relatedness of tags.

Despite these impressive progresses, the question of why users should contribute remains to be investigated (Siorpaes et al. 2010). Ontogame (Siorpaes et al. 2008) employed the idea of "games with a purpose" and proposed a game for ontology building that asked users to check the structure and abstraction from random wiki pages. The fun of playing game was the motivation to attract users' participation. In the InPhO experiment (Eckert et al. 2010), users got monetary reward for their work. Braun et.al proposed an image-based navigation system that was able to manage domain specific ontology and allow it mature in the daily work of users (Braun et al. 2007). In the system, instead of tagging a new image with additional tags, users were able to pull one image over or under another image via drag and drop. The tags used to annotate the upper image were thus collected as more general terms. With the drag and drop operation, users were supported to get better organization of collections of images and enhance their work performance.

ONTOASSIST: TOWARD AN OPEN CROWDSOURCING APPROACH FOR ONTOLOGY EVOLUTION

In this section, we describe the design of OntoAssit, a semantic navigation support service and tool. We focus on three key features of OntoAssist: sustainable crowd attracting method, crowdsourcing based ontology evolution and complementary domain knowledge support.

Sustainable crowdsourcing motivation

The success of any crowdsourcing approach relies on strong and long lasting motivation to attract enough crowd. Fun of gaming satisfies both; however its application is limited to certain problem domains. In addition, it has the issue of biased participants as it attracts only people who play games. Monetary award is able to attract all sorts of participants. Yet it is only applicable for short term projects. We aim to design a method to attract all sorts of Internet users with strong and sustainable motivation.

We piggyback OntoAssist on a general purpose search engine. This immediately gives us nearly the whole Internet population as candidate participants. To ensure enough traffic, OntoAssist is designed to provide simple and intuitive semantic navigation on query results. Such semantic navigation helps a user locate the desirable result efficiently by filtering out tens of thousands unrelated entries. Moreover, OntoAssist continues evolving its underlying ontology based on user inputs. Users can feel the improvement of services provided by OntoAssist from time to time. This helps to retain existing users and to attract new users.

Crowdsourcing based ontology evolution

The power of semantic navigation comes from its backing ontology. The improved semantic navigation experience ties tightly with the evolution of that ontology. OntoAssist aggregates large amount of user inputs collected from the semantic navigation interface to evolve the base ontology. The ontology evolution model of OntoAssist consists of the following three components:

(i.) Semantic navigation

The design of the semantic navigation component in OntoAssist is based on general search assist tools most search engines provide. Queries submitted to search engine usually consists of very short keyword phrase. Search assist, such as *related terms suggestion*, is one of the useful ways to help to understand the query intent by adding additional related terms from certain background knowledge to the query. With *related term suggestion* enabled, any query submitted to the search engine will come back with a set of terms. Users then click one of them to filter the search result. Popular search assist applications include yahoo *search assist*, Google *related search suggestion*, Bing *related search* and so on. Those search assists are interested in a general association between terms. It is reasonable to assume that most users are aware of the semantic relationship between the query word and the suggested terms although there is no explicit way for them to express it. We attempt to collect both these related terms and their relationships for ontology evolution purpose. The semantic navigation component allows the users to express their search intent as a tuple (*keyword, relation, related term*). For instance, a user can refine an original query *python* by the tuple: (*python*, *is-a*, *programming language*).

(ii.) User inputs aggregation

We then aggregate these terms and relationships from different query sessions. We assume that one expression is correct if majority of the users agree on it. Furthermore, we do not treat all user inputs equally. Analysing query log helps us to distinguish users into trusted or untrusted users on the purpose of knowledge collection. We provide an option for users to register and login for the use of personalized services and record their behaviours. This makes it easy to distinguish registered users as trusted and untrusted. Anonymous users are ranked based on trustworthiness computed using query log. Inputs made by trusted usershave high impact in the assessment of the collections.

(iii.) Versioning control and automatic update

Noy et al. have presented a framework for collaborative ontology development, designed for domain experts (Noy et al. 2006). We adapt the framework to use in Internet environment where large numbers of non-experts are able to contribute. The adapted framework has the following features. *Asynchronous*: every user checks out a part of concept related to his/her own query, edit, and submit back to the system. *Monitored*: the system records all the changes and other metadata such as time or IP address. In fact, users do not change ontology directly but only submit proposed changes to a separate log database. The system will apply the *changes periodically* to the old version and then release a new one for further editing. Change conflicts will be resolved during the aggregation under majority rule adjusted by user impact.

Domain knowledge support

The semantic representation of user's search intent is expressed as a list of related terms and a set of possible relations. The construction of semantic representation follows two simple guidelines: it should be understandable to the users and be able to distinguish the intent of the original query well (Hu et al. 2009). In terms of ontology evolution, the domain concept should cover the domain well and should be able to reflect new and emergent terms.

We attempt to leverage both the existing ontology and Wikipedia category for related terms generation. Obviously, there is always a gap between the number of terms representing users' search intent and the amount of existing domain terms. Wikipedia, one of the best and biggest online knowledge base, can help us infer a user's query intent when certain keywords may not be covered or correctly interpreted in existing ontology. The article link and category link provided in Wikipedia shows a kind of semantic connection to each connected nodes. Initially, we map the query into the WordNet space and get related terms and relationships. We map the query into Wikipedia link graph space as well for additional, relevant terms. Thus, a comprehensive set of candidate conceptual terms and relationships can be developed.

In short, we show how users' search intent can be captured to help to evolve ontology while helping search results. For example, we analyze the query log and find out that a number of users query for "python" agree on these inputs: "*python*, *is-a*, *programming language*". "c*python*, *is-a*, *python*" and "*jython, is-a, python*". We then incorporate them into initial computer ontology. This enables the system to expand the query for "python" to "cpython" and "jython". It also removes search results other than "programming language", such as "snake" or "animal".

IMPLEMENTATION

In this section, we present an implementation of the semantic navigation support tool on which the service is based. OntoAssist is integrated with Yahoo search engine and to form a new web system, www.hahia.com. We show how this system can be used to assist users in performing disambiguation of search intent and contributing to ontology evolution.

The hahia.com website is built on browser/server model. The user interface is developed with PHP and AJAX and run on Apache 2.2 web server. We choose WordNet as our ontology knowledge base since it is an upper ontology that crosses many domains, which is more suitable than other specific domain ontology in general search engine. The backend of our platform is a web service that generates related terms from WordNet and Wikipedia website.

It is developed using java language and JAWS[3] API, and return related terms in the format of XML. The platform is also a web search engine based on Yahoo! Search BOSS Framework[4] which utilizes the entire Yahoo! Search Index, ranking and relevance algorithm. MySQL is used as a database to store the entire user and other log information. All the backend run on a Centos 5.3 server.

Overall structure

Figure1 describes an overview model of the website. It consists of three core modules: user search interface, ontology evolution module, and semantic search module.

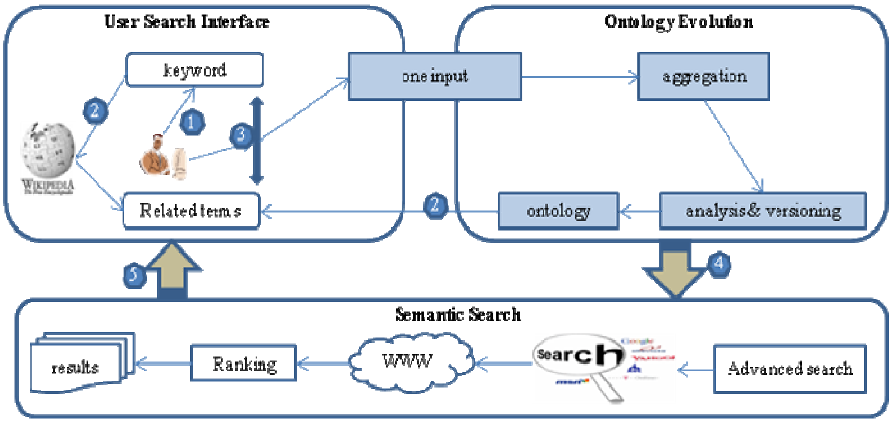

Figure.1. An open crowdsourcing model for ontology evolution with semantic search engine

(i.) User Search Interface

User search interface provides an interface that allows users to pick up terms and assign the semantic relationship between them. There are three main parts in user search interface: *Keyword interface* where user can input the keyword. *Related terms generator* provides related terms which are generated from WordNet and Wikipedia. *Relationship generator* provides relationships corresponding to the keyword from WordNet. Users can make explicit the semantics of her(his) query by simply selecting one related term and assigning the relationship between the query keyword and related term.

(ii.) Semantic navigation module

The aim of this module is to improve the search precision and recall and to provide user with better navigation based on domain knowledge. By assigning relationship between the query keyword and one of the related terms, a user is able to express his/her query intent in a machine understandable format. Thus, the precision can be improved by performing an advanced search with the match of additional related term. It also removes pages that have unwanted terms from other domain. The query can also be expanded to other related terms, for example, synonyms in the same domain of interest. With the use of JSON and AJAX technique, the refined result can be pushed to user automatically without the need to refresh the web page. Semantic navigation is a plus to improve user satisfaction by enabling them quickly explore the concept in the relevant domain.

(iii.)Ontology Evolution Module

The backend is the ontology evolution module. It has two purposes. First, it generates a part of related term based on WordNet and returns them to user interface. Second, it records all the relationships and terms selected by users. Users' query logs are recorded as well. All these term-term relationships remain unreleased status until they are validated by the system.

Demonstration

The alpha version of OntoAssist platform has been release and can be accessed online via http://www.hahia.com. Figure 2 illustrates a snapshot of the OntoAssist platform. On top of the web page is the main user interface, including search box and disambiguation assist. There are two separate columns under the disambiguation box. On the left hand column, terms from ontology base has been provided and grouped into different domains. By clicking one of the related terms and one of the relationships, the system refines the search result and returns the new result on the right column. Terms from other domains are also removed on the left column.

[3] http://lyle.smu.edu/~tspell/jaws/index.html
[4] Yahoo search BOSS Framework is a Yahoo!'s open search web service that allow developers use it to build web-scale search product. http://developer.yahoo.com/search/boss/

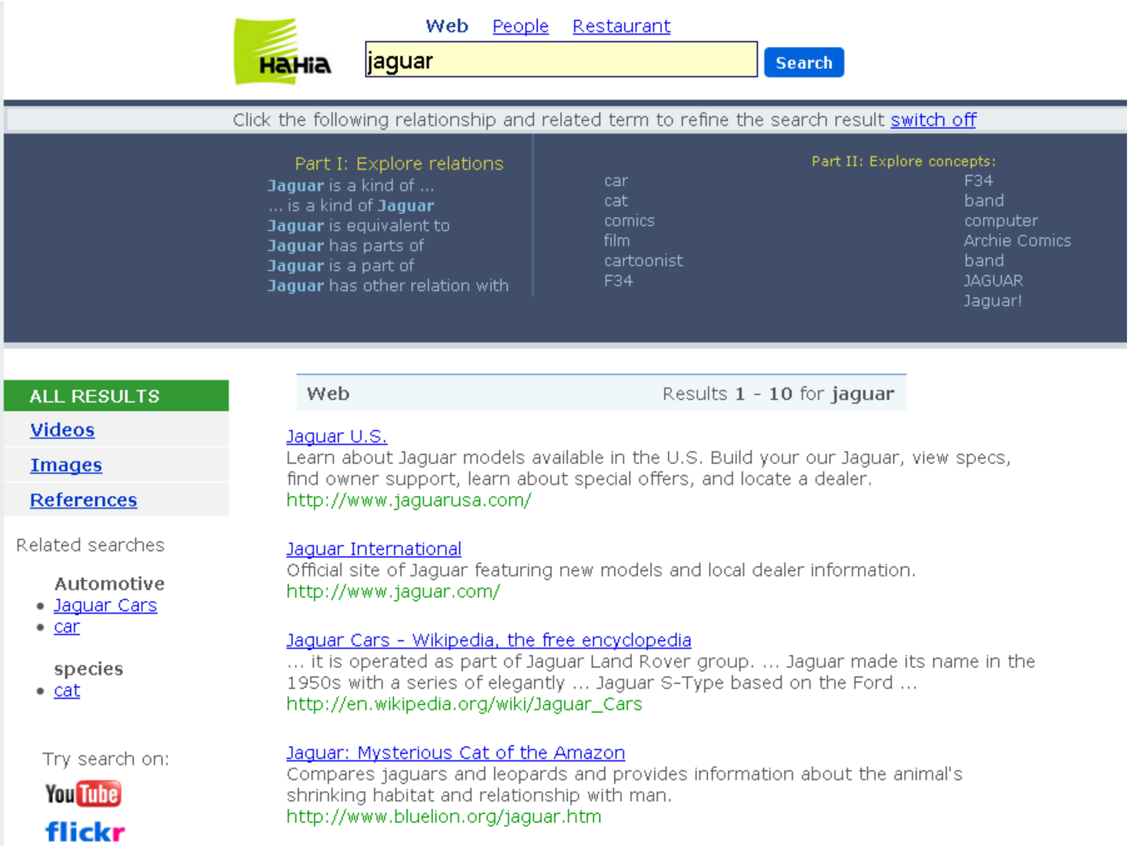

Figure 2, Snapshot of the OntoAssist platform, hahia.com

Figure 3 illustrates a typical data flow example. In this scenario, a user is looking for "jaguar":

i. A User inputs a keyword "jaguar" and submits it to the system.
ii. Related terms including "cat, car, band" are generated and presented to the user.
iii. The User clicks one of the related terms "car" and then clicks one of the relationship "is a kind of" to express that s/he is looking for jaguar which is a kind of car, but not a band or anything else.
iv. The input tuple (jaguar, is a kind of, car) is captured by the system and stored in the log database for future analysis and ontology maintenance.
v. The system refreshes the navigation bar in the left column of the webpage and only shows related terms in the automotive domain, such as "jaguar cars", "Benz".
vi. The system removes all the results that are in other domain such as species or band; it also expands search with models of jaguar cars, ranks the result and returns them to the user.

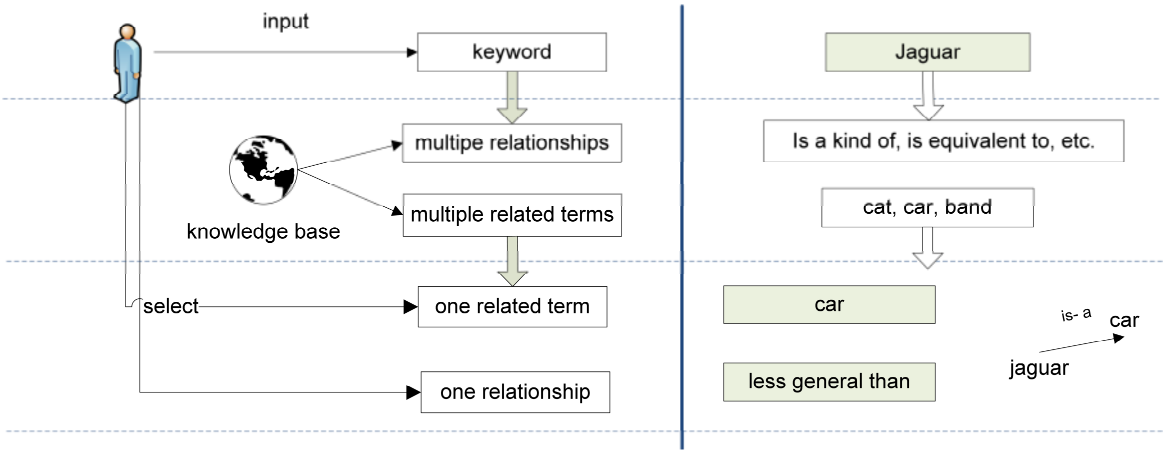

Figure 3, An example search of jaguar (car)

EXPERIMENT

In this section, we want to use the experiment to show that OntoAssist and crowdsourcing can discover new terms and facilitate rapid ontology evolution.

Experimental setup

Test ontology

To validate the model and our approach, we chose a part of ontology in the computer domain from WordNet as our test ontology. We manually query "computer" in WordNet 2.1 and got 87 terms, including 6 synonyms terms, 4 hypernyms terms, one term in "is a part of" relationship, 24 terms in "has parts of" relationship, and another 50 terms "is a kind of" computer. In our experiment, users are only allowed to issue queries with one of these terms.

The source of online users

Evaluation of the tool needs the participation of public users. Practically, we need to organize certain number of users to try the new prototype in short time. Traditional evaluation approaches can be quite time-consuming since they involve recruiting a large number of volunteer subjects. It is not easy to organize and usually need weeks to prepare the application if hundreds of subjects are needed. In our experiment, Amazon Mechanical Turk (MTurk) is introduced as a tool to source public users. MTurk is a web-based service that enables developers to outsource certain task to human across the world. All online users can apply and become Turkers for MTurk. Each unit of work is called HIT or human intelligence task. It typically costs only few cents to complete. Based on this service, we ask users from MTurk to complete specific tasks and promise some payment to each complete task. To simulate the public users, Turkers were not aware of the purpose of this experiment. They only knew that they were performing normal queries using a search engine with additional search assistance plug-in that was developed as part of this project

Design of tasks

The design of MTurk tasks needs to satisfy three main goals. First, we have to make sure that each HIT result came from the real use of our search engine. Second, the design should be simple enough. Finally, we need a way to measure the quality of their work.

Our task titled "select a related term and specify a type of relationship" was designed to get human's knowledge on a specific term. With further click on the task link, user could see the full description of the job:

i. A term x was selected from the test ontology. Turkers were asked to click a given hyperlink, which led them to start a query of term x on hahia.com.

ii. Top 12 related terms were generated and presented to Turkers. They need to review them and select the most relevant term.

iii. Turkers specified a type of relationship between the given term and selected related term. The system will then start a new query based on the specification. A refined search result will be presented to Turkers.

iv. Turkers were requested to go back to MTurk website and submit the selection by pasting the selected term and relationship in the field provided.

For example, given a keyword "redhat", if a Turker clicks the given web URL, s/he will find a list of candidate terms. S/he might select the term "operating system" and assign a relationship "is a kind of" to them. These inputs from the Turker make the following assertion: (*redhat, is a kind of, operating system*), we call it a judgment. There was also an optional field that let Turkers write their comments/suggestion on the use of our OntoAssist platform.

Results and evaluations

The experiment completed in about 3 hours. In our experiment, each participant was required to have an MTurk account and he/she can only participate once with each unique term. We have collected 1935 judgements/HITs from 225 individual Turkers, with the total cost of $34.74. The price was calculated by estimating the time users need to complete a HIT and Turker's hourly pay, $3.6/hour (usual price in MTurk). The Turkers came from 8 countries. The majority was from India, and the others were from USA, Romania, United Arab Emirates, Macedonia, etc. Figure 4 shows the top 100 contributing Turkers where each bar represents an individual Turker. The numbers indicates how many judgements have been submitted in the experiment.

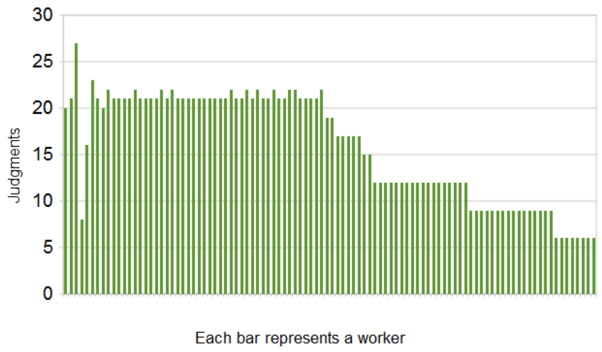

Figure 4, Judgement per Turker

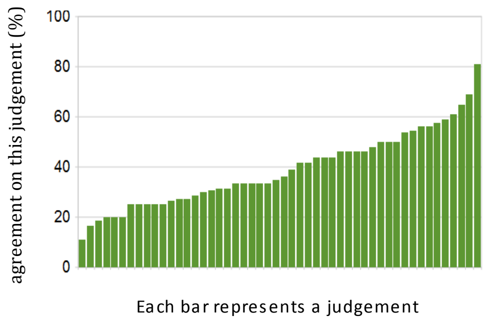

Figure7, Aggrement per unit

As we described in the design section, it is important that we assess the quality of inputs and only collect the meaningful ones because we are dealing with a diverse population of users of the open search engine. We have employed two strategies to assess the quality. First, we introduced 5 golden standard tasks (Queries with five special keywords, i.e. PC, dedicated file server, bulletin board system, analog computer and CRT) in all the work pool to be used to assess the quality of Turkers' work. These golden standards randomly appeared in the queries that were pushed to the Turkers. For each query, we have a complete set of correct relationships and if a user chooses one of the correct relations, that is considered accurate. With this setting, we were able to identify the untrusted Turkers and then excluded the inputs from the entire collection. Turkers who have less than 40% accuracy were recognized as untrusted Turkers. Figure 5 shows that all Turkers completed the jobs at an average of 68% accuracy against the gold standards. It also shows that the Turkers who were classified as trusted Turkers have a significantly higher accuracy of 96% on the average, while the Turkers who were classified as untrusted Trukers only have 22% accuracy on the average. In Figure 6, we have a further look at each of the 5 golden standard units. It shows that they have almost same accuracy. Finally, 777 judgements made by untrusted Turkers were excluded from the result. 1158 inputs remain as trusted inputs.

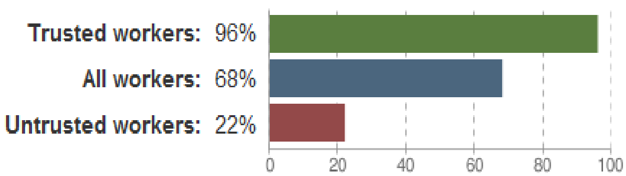

Figure 5, Average golden accuracy

Figure 6, Percent correct on golden data

While golden standards are helpful in removing untrusted judgments from the collection, agreement is an important parameter to aggregate the trusted judgments and work out a common concept from them. In our experiment, each term were queried by at least 9 distinct Turkers. Figure7 shows that there are agreements from different users over most of the judgements. Some judgements may have lower number of agreements simply because those terms have different kinds of relationship with more than one related terms. For example, in Table 1, three users (Turker ID 235670, 248921, and 245099) have made judgements that (*software platform, is a kind of, platform*), while there are also two users (Turker ID 305701 and 169892) declared that (*computing platform, is a kind of, platform*). Indeed, both of them are correct.

Table 1, An example inputs from Turkers, collected with the keyword "platform"

workerid	termA	relationship	termB
235670	software platform	is a kind of	***platform***
251210	***platform***	is a kind of	construction
46441	***platform***	has parts of	Software Framework
248738	***platform***	has other relations with	construction
305701	Computing platform	is a kind of	***platform***
248921	software platform	is a kind of	***platform***
169892	Computing platform	is a kind of	***platform***
245099	software platform	is a kind of	***platform***
256280	***platform***	is equivalent with	Computing platform

We then aggregated the results by applying the rule of majority agreement. With this aggregation, we finally get a common view from Turkers for each HIT. By comparing the aggregated result with the original ontology, we find out that 173 additional domain terms from Wikipedia were collected, together with relationships among them. These terms reveal the new emergent concepts in the domain of computer such as *"Logitech G51, flash memory, NAS"*. Furthermore, the relationships there show their connections to existing terms in WordNet. Some explain themselves with "is equivalent to" relationships. For instance, *(network-attached storage, is equivalent to, NAS), (dynamic random-access memory, is a kind of, memory), (floppy disk, is a kind of, removable storage device)*. Figure 8 shows a part of resulting ontological structure. Terms and relationships with dashed line are new collections from users' inputs.

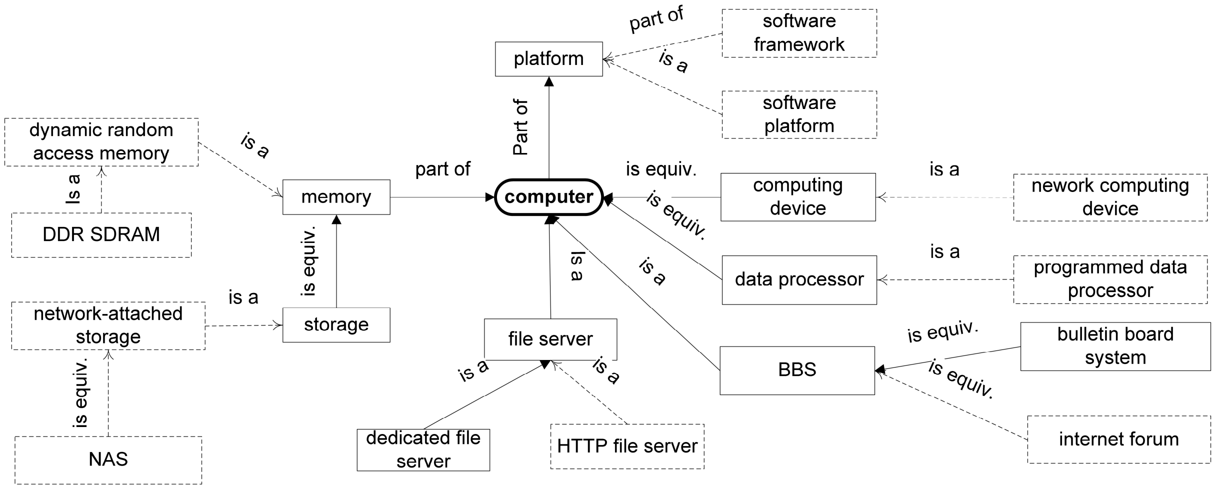

Figure 8, A part of resulting ontological structure.

We manually reviewed those terms and relationships among them. We discovered that 89% of the new terms were relevant to computer domain. The accuracy of the relationship among them was much lower, about 62%. We found out that some users got confused with two different description of relationships in the experiment: *"… is a kind of term x"* and *"term x is a kind of …"* The accuracy of relationship might be improved if we change these two type of relationships to "less general" and "more general".

CONCLUSION AND FUTUREWORK

In this paper, we have attempted to develop a service for addressing the challenge of ontology evolution by enabling the extraction of new concepts and the semantic relationships among them. A novel crowdsourcing approach has been presented to attract Internet-scale distributed users to participate in this work while they go about doing normal search activities. An implementation of the OntoAssist semantic navigation tool has been presented. We have conducted an experiment using our platform with more than 200 Turkers from Amazon Mechanical Turk. The work has produced promising results in terms of helping evolve a test ontology. Considering that hundreds of billions searches conducted each month, this has the potential to bring a significant change in the way we approach ontology maintenance and evolution, and thus accelerate the maturity of semantic web.

To improve the accuracy of aggregation of terms, we plan to develop newer strategies to distinguish usage from multiple domains, and to handle the conflicts and disagreement among users by improving the aggregation techniques. Experiments with more participants are also necessary to collect larger usage datasets for analysis.

REFERENCES

Brabham, D.C. "Crowdsourcing as a model for problem solving: An introduction and cases," *Convergence* (14:1) 2008, pp 75-75.

Braun, S., Schmidt, A., and Walter, A. "Ontology maturing: a collaborative web 2.0 approach to ontology engineering," CKC workshop at WWW 07, Banff,Canda, 2007.

Dawkins, R., and Pyle, L. *The Blind Watchmaker* Penguin Harmondsworth, 1991.

Eckert, K., Niepert, M., Niemann, C., Buckner, C., Allen, C., and Stuckenschmidt, H. "Crowdsourcing the Assembly of Concept Hierarchies," the 10th ACM/IEEE Joint Conference on Digital Libraries, 2010.

Flouris, G., Plexousakis, D., and Antoniou, G. "Evolving ontology evolution," *SOFSEM 2006: Theory and Practice of Computer Science*) 2006, pp 14-29.

Gentry, C., Ramzan, Z., and Stubblebine, S. "Secure Distributed Human Computation," the 6th ACM conference on Electronic commerce, 2005, p. 164.

Gruber, T.R. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal of Human-Computer Studies*) 1993.

Hepp, M., Bachlechner, D., and Siorpaes, K. "OntoWiki: community-driven ontology engineering and ontology usage based on Wikis," 2006 international symposium on Wikis ACM, Odense, Denmark, 2006, pp. 143-144.

Howe, J. "The rise of crowdsourcing," *Wired Magazine* (14:6) 2006, pp 1-4.

Hu, J., Wang, G., Lochovsky, F., Sun, J.-t., and Chen, Z. "Understanding user's query intent with wikipedia," the 18th international conference on World wide web, ACM, Madrid, Spain, 2009, pp. 471-480.

Limpens, F., Gandon, F., and Buffa, M. "Collaborative Semantic Structuring of Folksonomies," IEEE Computer Society, 2009, pp. 132-135.

Noy, N.F., Chugh, A., Liu, W., and Musen, M.A. "A framework for ontology evolution in collaborative environments," *The Semantic Web-ISWC*) 2006, pp 544-558.

Noy, N.F., and Klein, M. "Ontology Evolution: Not the Same as Schema Evolution," *Knowledge and Information Systems* (6:4) 2004, pp 428-440.

Siorpaes, K., and Hepp, M. "Games with a Purpose for the Semantic Web," *IEEE Intelligent Systems* (23:3) 2008, pp 50-60.

Siorpaes, K., and Simperl, E. "Human Intelligence in the Process of Semantic Content Creation," *World Wide Web* (13:1) 2010, pp 33-59.

Stojanovic, L., Maedche, A., Stojanovic, N., and Studer, R. "Ontology evolution as reconfiguration-design problem solving," the 2nd international conference on Knowledge capture, ACM, 2003, p. 171.

Von Ahn, L. "Human computation," the 4th international conference on Knowledge capture, ACM, Whistler, BC, Canada, 2007, pp. 5-6.

Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., and Blum, M. "reCAPTCHA: Human-Based Character Recognition via Web Security Measures," *Science* (321:5895) 2008, pp 1465-1465.

COPYRIGHT