8-6-2011

Web 3.0 and Crowdservicing

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

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

Web 3.0 and Crowdservicing

Joseph G. Davis
School of Information Technologies
The University of Sydney
Sydney NSW 2006 Australia
joseph.davis@sydney.edu.au

Winston H. Lin
School of Information Technologies
The University of Sydney
Sydney NSW 2006 Australia
huairen.lin@sydney.edu.au

ABSTRACT

The World Wide Web (WWW) has undergone significant evolution in the past decade. The emerging web 3.0 is characterized by the vision of achieving a balanced integration of services provided by machines and human agents. This is also the logic of 'crowdservicing' which has led to the creation of platforms on which new applications and even enterprises can be created, and complex, web-scale problem solving endeavors undertaken by flexibly connecting billions of loosely coupled computational agents or web services as well as human, service provider agents. In this paper, we build on research and development in the growing area of crowdsourcing to develop the concept of crowdservicing. We also present a novel crowdservicing application prototype, OntoAssist, to facilitate ontology evolution as an illustration of the concept. OntoAssist integrates the computational features of an existing search engine with the human computation provided by the crowd of users to find desirable search results.

Keywords

Web 3.0, crowdsourcing, ontology, Mechanical Turk.

INTRODUCTION

There is considerable interest in recent years in the possibilities offered by what has come to be known as web 3.0. Also referred to as the service web or the semantic web, it builds on the developments achieved with web 2.0. Common descriptions of web 3.0 have included the combination of semantic technologies and service oriented architecture (SOA) to create an infrastructure that allows numerous parties to expose and consume large number of services in a seamless and transparent manner. The stakeholders could range from individual end users to small and large enterprises (Domingue et al. 2009). It is essentially read-write-execute web and it subsumes both web 2.0 and web 1.0.

Web 2.0 introduced the critical feature of user contribution and its impact has been massive in the rise of a vast array of social media sites and applications. We are still living through the reverberations and implications of web 2.0 applications which have changed the way we live, learn, educate, entertain, deal with our governments etc. Also, known as the read-write web or the participative web, it represented a major shift from the largely exposure-based features of web 1.0.

Web 3.0 also represents a significant transition in that it enables us to develop technologies and platforms based on which complex applications and even new enterprises can be created by flexibly linking millions of loosely coupled services with a large number of human service provider agents. It is envisaged to be a marriage between the rapidly improving semantic services provided by the machine with the semantically rich albeit somewhat less predictable services provided by the 'crowd'. We argue that the capabilities provided by web 3.0 and the evolving conceptual development and practice of crowdservicing has the potential to lead to a fuller realization of the promise of 'augmentation' that Douglas Engelbart (Engelbart et al. 1962) has consistently championed for nearly 50 years.

Crowdservicing can be viewed as the next stage in the evolution of crowdsourcing which has received much attention in recent years. The term 'crowdsourcing' was first coined by Jeff Howe, the author of 'Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business' (Howe 2009) as an approach to draw large number of people into solving complex problems and completing certain tasks by tapping in to their vast expertise and knowledge that was previously not available. It originated as means to facilitate the aggregation and/or selection of information and knowledge from a large number of interested people connected to the internet. Wikipedia and Creativechallenge are good examples of distributed information and knowledge sharing and organization. It involves directing tasks traditionally performed by experts to non-experts, typically a large group of people through open forums and sites on the web. It can harnesses the collective knowledge and intelligence of a vast number of individuals to offer solutions to open problems and very often, winning ideas are rewarded.

However, crowdsourcing has evolved rapidly in recent years. It has now become a generic expression for a wide range of endeavors on the internet including prediction markets, distributed problem solving over the internet, open innovation, mass collaboration, cheap and efficient human computation and problem solving using Mechanical Turk, among others. The aspects that seem to be common across these are (a) the assignment of a problem or the distribution of some work to a large number of independent (volunteer or paid) individuals or groups through the internet, (b) some mechanism to aggregate or select from the submissions, (c) optional offer of rewards or payment. This is seen as more robust alternative to the use of in-house teams of experts or a chosen group of contributors for a wide array of problems. The basic assumption is that the crowd can bring interesting, non-trivial, and non-overlapping information, insights, or skills which can add to the quality of the solutions when harnessed through appropriate aggregation and selection mechanisms.

Crowdservicing is more consonant with Web 3.0. It is making possible the aggregation of actual services and human computation by the crowd and its integration with the computational power of the machine. The contributing agents in the crowd are more than just creators, annotators, exposers, and consumers of content and services. There are many situations in which the human agents can provide whole or part of a badly needed service which in combination with one or more computational services can achieve a level of service quality that cannot be achieved under each regime separately. This is the potential that crowdservicing offers.

In this paper, we define and develop the concept of crowdservicing as a distinct class of web-scale problem solving approaches as compared to crowdsourcing. We also discuss the technical and other infrastructure that has evolved to support scalable crowdservicing applications. We proceed to develop an application prototype in the domain of ontology evolution as a demonstration of crowdservicing and report on experiments using it using the design research methods.

Ontology domain is appropriate since maintaining ontologies over time has proven to be a task that is difficult for the experts alone and the contributions are needed from a large number of participants. The task also needs to be complemented with computational support provided by the machine. Ontologies provide formal representation of knowledge in a domain as a set of concepts and their relationships. They play a crucial role in the evolving semantic web (Fensel et al. 2005). For instance, ontologies can be used by search engines to improve the precision and accuracy of web searches by searching for web pages that refer to specific concepts and related ones, instead of relying on ambiguous keywords (Berners-Lee et al. 2001).

The rest of the paper is organized as follows: In the next section, we review the related work. In section 3, we provide a conceptual development of crowdsourcing followed by a discussion of the design research method employed in section 4. In section 5, we present our crowdservicing model for ontology evolution and discuss the detailed implementation of the prototype illustration based on Yahoo! search engine. Experimental results are also presented and evaluated. Finally, we present our conclusions and future work.

RELATED WORK

Ontology is perhaps the most critical enabling technology for 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). Most of the ontologies developed are based on the efforts of domain experts. This can lead to several issues. First, it is expensive to employ 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). Third, 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 and have no control over the resulting ontology. Thus, the efficacy and value of the ontology-based application can be limited.

Ease of availability of non-experts and the motivation for them to participate are two key factors in the growing interest in crowdsourcing for ontology-related tasks. As well, the logic of crowdsourcing suggests that the aggregated inputs from a sufficiently large number non-experts can provide better solutions than individual experts for a range of tasks. Online on-demand labor markets such as Amazon Mechanical Turk (MTurk) have opened the door for exploration of this shift towards crowdsourcing for innovation (Little 2010). MTurk is a web-based service that enables developers to outsource certain tasks to human agents all over the world. Any online user can apply and become workers for MTurk, a.k.a., Turker. Each unit of work is referred to as a human intelligence task (HIT). Given the narrow scope of each HIT, it the cost per HIT is relatively small. MTurk has been adopted in natural language processing tasks and proven to be a significantly cheap and fast method. Examples include experiments involving the collection of a large number of data annotations (Snow et al. 2008; Sorokin and

Forsyth 2008), descriptions of image (Little 2010), learning and populating a taxonomy (Eckert et al. 2010), and assessing document relevance (Grady and Lease 2010).

In additional to monetary rewards, the spirit of service and other social-psychological incentives can also promote users' contribution behavior in the design of crowdsourcing task (Antin and Cheshire 2008; Gallaugher 2010). For instance, Facebook[1] leverages its members' knowledge to develop localized versions in various languages. Facebook engineers have collected thousands of English words and phrases throughout its website and designated each of them as a translation objective. Members were invited to translate the individual terms and rate them to select the best translation. Using this form of crowdsourcing , Facebook has attracted thousands of volunteers and completed several location tasks within a few days (Gallaugher 2010; Kirkpatrick 2008). Braun et al proposed an image-based navigation system that was able to manage a domain specific ontology and allow it to mature as a by-product of the daily work of users (Braun et al. 2007). In this 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 the more general terms. With the drag and drop operation, users' collective knowledge was harnessed to obtain better organization of collections of images while simultaneously enhancing their work performance.

CROWDSERVICING

Crowdservicing represents the next stage in the evolution of the models of crowdsourcing. It highlights the role of actual service provision including human computational service which is can be viewed as a generalization of contributing information, knowledge or ideas. The key challenge in developing effective crowdservicing model is the innovative integration of human contributions with computational models and services processed by the machine. In this perspective, human agents are viewed as more than contributors of content or source of information, knowledge or ideas. There are many problems for which human agents can actually offer whole or part of a badly needed service (eg. evaluation of the output of a language translation job; human query processing to complement database query processing) which in combination with one or more computational service can achieve a level of service quality that cannot be achieved under each regime (Davis 2011).

Crowdservicing represents the full flowering of the augmentation concept developed by Engelbart with the goal of augmenting the human intellect by developing technologies and systems for manipulating information to improve individual and group processes and knowledge work (Engelbart et al. 1962). The technical infrastructure for crowdservicing includes the evolving internet and web 3.0, the emerging cloud computing platforms, web-scale data management and semantic technologies, service oriented computing and web services and their orchestration, computational agents, various interfaces to achieve programmatic access for availing crowd-based services, and the diverse devices for user access. In order to genuinely support crowdservicing needs, the infrastructure needs to demonstrate the requisite variety and seamlessness demanded by the project. Many a research group and start-up company is presently engaged in researching, developing, and refining parts of this infrastructure to make it better suited to the needs of large-scale crowdservicing. The softer side of the infrastructure comprises potentially large number of human agents with internet access possessing diverse information and human computation capabilities and the willingness to contribute these at appropriate terms. This valuable resource can only grow with time as more people around the world get connected and the digital divide is narrowed.

Crowdservicing has been discussed in the context of distributed human computation (Schall et al. 2010). Useful examples include the Google image labeler which utilizes the power of human computation to label images with relatively high levels of accuracy to improve the quality of image search (Von Ahn and Dabbish 2008) and the Stanford project attempting to enable database querying to scale up to be able to address the complexity of contemporary intra- and inter-enterprise information needs by developing declarative approaches to answering queries by adding crowd service inputs to the databases and algorithms (Parameswaran and Polyzotis 2011). It clearly recognizes the distinctive aspect of human computation that enables human agents to analyze contexts effectively, disambiguate concepts, and to infer certain facts not included in the database while acknowledging the probability of errors and failures.

Crowdservicing has the potential to radically alter the landscape of service delivery. It can lead to a scenario for the future of computing in which 'everyone is a service' (Petrie 2010) and many complex problem solving and task execution can take place outside the boundaries of business firms and other institutions. As cognate technologies such as cloud computing develops further, it also offers the possibility of new startups and other enterprises to scale up very rapidly and achieve striking results in short order, referred to as 'flash companies' by Woods (Woods 2010).

[1] http://www.facebook.com

RESEARCH METHOD

Having presented the crowdsourcing concept, we present a sketch of the research method employed in this paper as partial demonstration and validation of the concept. The approach we adopt is to develop a prototype crowdservicing application in the area of ontology evolution. The task is eminently amenable to crowdservicing. Many of the ontologies are developed by teams of experts but the task of maintaining and evolving them over time has proven to be difficult for the experts (Braun et al. 2007). Several purely computational approaches to the problem have also been proposed but have been found to be wanting (Stojanovic et al. 2003). There are good reasons to believe that the crowdservicing approach offers greater potential.

The design research method elaborated by (Hevner et al. 2004) provides the basis for our methodological stance. The first stage in this approach is the design of the artifact in the form of a prototype that serves as a validation of the crowdservicing concept developed in the foregoing. The problem relevance and the need for a socio-technical solution and appropriate platform has already been established. After building the demonstration prototype, OntoAssist, we proceed to perform systematic experimental evaluation in relation to the key features and capabilities of the prototype. The technical and managerial dimensions of the research based on the prototype and its deployment are articulated.

PROTOTYPE DEVELOPMENT: ONTOASSIST

In this section, we describe OntoAssist, a prototype implementation of a crowdservicing application. It is a semantic navigation support system designed to address the ontology evolution problem. We focus on three key features of OntoAssist: sustainable crowd attracting method, crowdsourcing based ontology evolution and complementary domain knowledge support. The computational component is provided by Yahoo search engine. Experimental results and analysis are also presented.

Model

Sustainable crowdservicing

The success of any crowdservicing approach relies on strong and sustainable motivation to attract a sufficient number of human agents. Monetary award is able to attract all sorts of participants. Yet it is only applicable for short term projects, such as the building of ontology at the early stage. In this section, we present a method to attract a wide range of Internet users with strong and sustainable motivation.

We piggyback OntoAssist on a general purpose search engine which provides the computational component. This also gives us access to a large number of users as candidate participants. To ensure enough traffic, OntoAssist was designed to provide simple and intuitive semantic navigation over 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.

Crowdservicing-based ontology evolution

The power of semantic navigation comes from the underlying ontology. The improved semantic navigation experience is linked closely to the evolution of that ontology. OntoAssist aggregates a 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 four components:

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 kind of, programming language*).

User input 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 users have high impact in the assessment of the collections.

Version 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 the 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 candidate 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 related terms generated using the category feature in Wikipedia. Clearly, 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 the existing ontology. The article link and category link provided in Wikipedia show a kind of semantic connection to each connected node. Initially, we map the query into WordNet and get related terms and relationships. We map the query into Wikipedia link graph as well for additional, relevant terms. Thus, a comprehensive set of candidate conceptual terms and relationships can be developed.

We show how users' search intent can be captured to help evolve the ontology while helping to refine 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 kind of, programming language*". "c*python, is a kind of, python*" and "*jython, is a kind of, 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 our implementation of the crowdservicing tool to support semantic navigation. 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 based on browser/server model. The user interface is developed with PHP and AJAX and run on Apache 2.2 web server. We choose WorldNet as our ontology knowledge base since it is an upper ontology that crosses many domains, which is more suitable than other domain-specific ontologies. The backend of our platform is a web service that generates related terms from WorldNet and Wikipedia website. It is developed using java language and JAWS[2] API, and return related terms in the format of XML. The platform is also a web search engine based on Yahoo! Search BOSS

[2] http://lyle.smu.edu/~tspell/jaws/index.html

Framework[3] which utilizes the entire Yahoo! Search Index, ranking and relevance algorithm. A MySQL database is used to store the entire user and other log information.

Overall structure

Figure 1 describes an overview model of the website. It consists of three core modules: user search interface, ontology evolution module, and semantic search module.

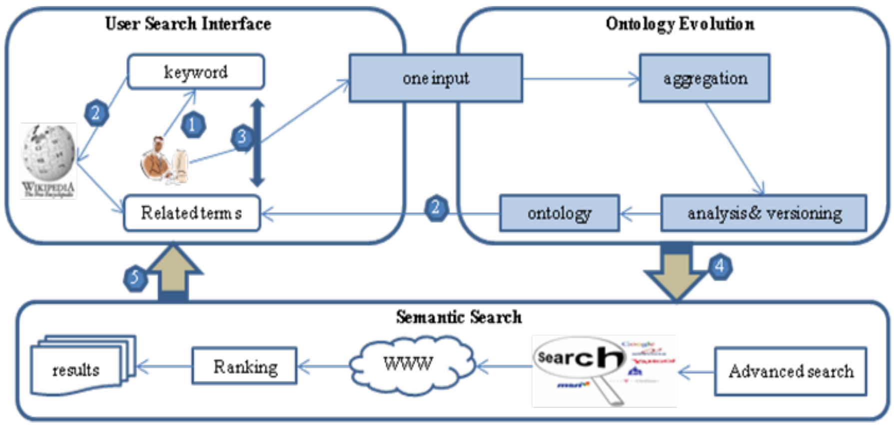

Figure 1, An open crowdsourcing model for ontology evolution with semantic search engine

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 selector* provides relationships 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.

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.

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 illustrate 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

---

[3] 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/

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.

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

a) A User inputs a keyword "jaguar" and submits it to the system.
b) Related terms including "cat, car, band" are generated and presented to the user.
c) 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.
d) 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.
e) 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".
f) 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.

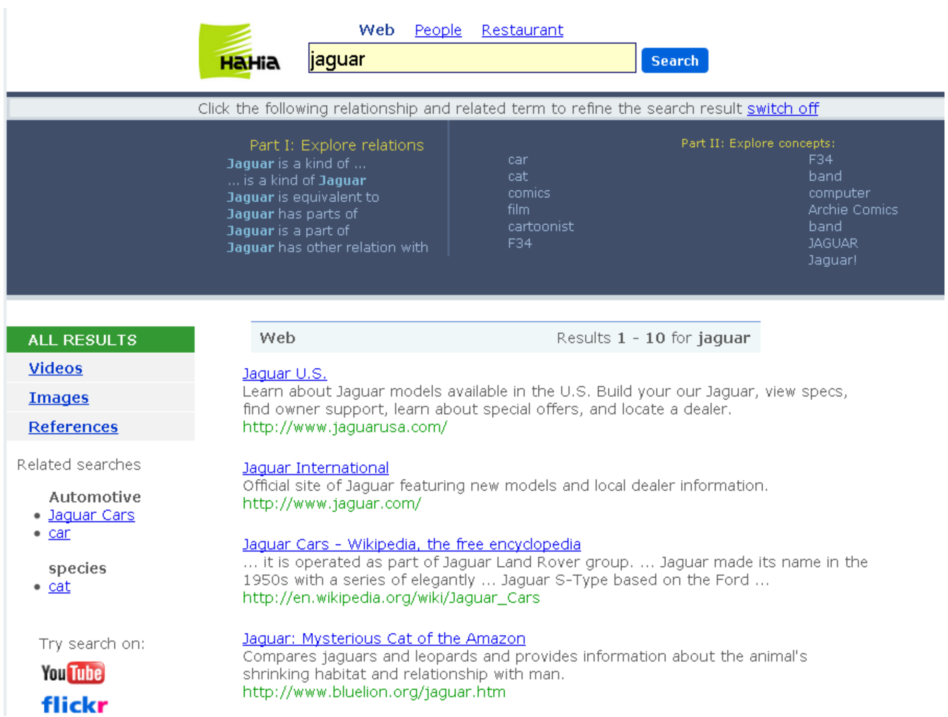

Figure 2, Screenshot of the OntoAssist platform, hahia.com

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

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.

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. In our experiment, MTurk is introduced as a tool to source public users.

The design of 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:

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

The experiment was completed in about 3 hours. Each participant was required to have an MTurk account and he/she can only participate once with each unique term. We have collected 1935 judgments/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 3 shows the top 100 contributing Turkers where each bar represents an individual Turker. The numbers indicates how many judgments have been submitted in the experiment.

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 five 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 4 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 5, we have a further look at each of the 5 golden standard units. It shows that they have almost same accuracy. Finally, 777 judgments made by untrusted Turkers were excluded from the result. 1158 inputs remain as trusted inputs.

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. Figure 6 shows that there are agreements from different users over most of the judgments.

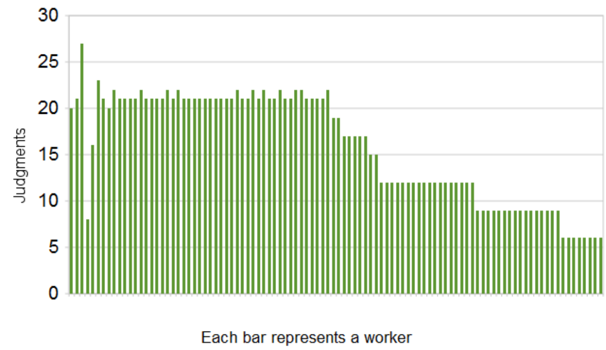

Figure 3, Judgment per worker

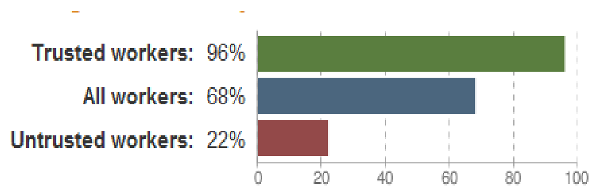

Figure 4, Average golden accuracy

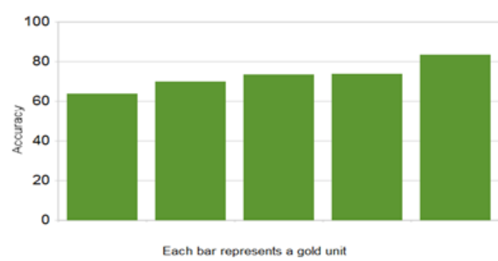

Figure 5, Percent correct on golden data

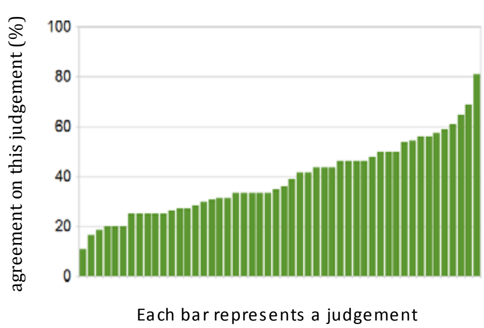

Figure 6, Agreement per unit

We then aggregated the results by applying the rule of majority agreement. With this aggregation, we finally get an aggregate 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 the 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 7 shows a part of resulting ontological structure. Terms and relationships with dashed line are new collections from users' inputs.

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".

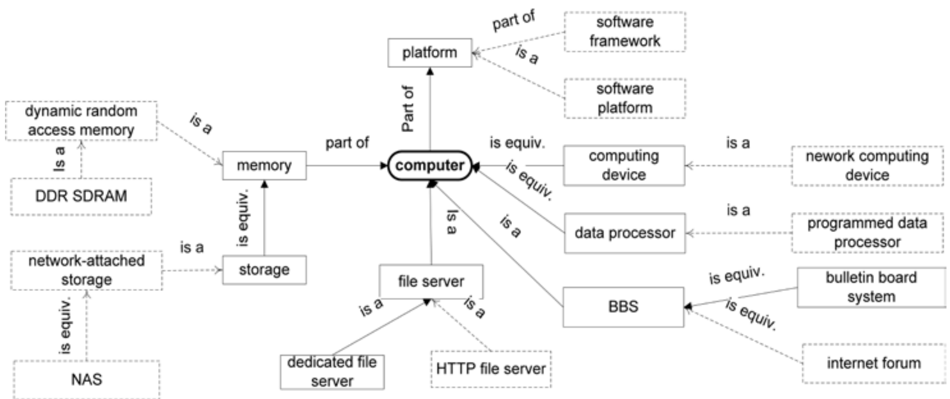

Figure 7, A part of resulting ontological structure

CONCLUSION, AND FUTURE WORK

We developed and amplified the concept of crowdservicing as the next stage in the evolution of crowd-based solutions to complex tasks and problems. It coheres well with the foundations of the emerging web 3.0 and the potential to radically transform web-scale problem solving and collaboration. There is considerable research and development activity in this space and the infrastructure for largescale crowdservicing is emerging at a rapid pace.

We also proposed a novel crowdservicing model in the domain of ontology evolution which attempts to use semantic search navigation tool to attract a large number of distributed users to participate in this work while they go about doing normal search activities. A prototype based on the model, OntoAssist, is developed as a demonstration and part validation of the crowdservicing concept. We elicit their knowledge with related terms generated from Wikipedia, and then aggregate their inputs for the purpose of enabling ontologies to evolve. Results of experiments using the prototype are presented as evidence of the value and efficacy of the concept.

In the future, experiments with more participants are necessary to collect larger datasets for analysis. We plan to develop newer strategies to handle the conflicts and disagreement among users by improving the aggregation techniques. We also would like to integrate the OntoAssist application with the API provided by CrowdFlower. Thus, we will have two sources of labor, both paid worker and unpaid users. We can employ paid workers to build the ontology from scratch and perhaps, rely on unpaid users to evolve it in the long run.

REFERENCES

1. Antin, J., and Cheshire, C. (2008) Designing Social Psychological Incentives for Online Collective Action, *Proceeding of the Conference on Online Deliberation*.
2. Berners-Lee, T., Hendler, J., and Lassila, O. (2001) The semantic web, *Scientific american* (284:5), pp 28-37.
3. Braun, S., Schmidt, A., and Walter, A. (2007) Ontology maturing: a collaborative web 2.0 approach to ontology engineering, *Proceeding of the CKC workshop at WWW 07*, Banff,Canda.
4. Davis, J.G. (2011) From Crowdsourcing to Crowdservicing, IEEE Internet Computing, May-June.
5. Domingue, J., Fensel, D., Davies, J., González-Cabero, R., and Pedrinaci, C. (2009) The Service Web: a Web of Billions of Services, *Towards the Future Internet*), p 203.
6. Eckert, K., Niepert, M., Niemann, C., Buckner, C., Allen, C., and Stuckenschmidt, H. (2010) Crowdsourcing the Assembly of Concept Hierarchies, *Proceeding of the the 10th ACM/IEEE Joint Conference on Digital Libraries*.

7. Engelbart, D.C., Friedewald, M., Institute, S.R., and Alliance, B. (1962), *Augmenting human intellect: A conceptual framework* Stanford Research Institute Menlo Park, CA.
8. Fensel, D., Van Harmelen, F., Horrocks, I., McGuinness, D., and Patel-Schneider, P. (2005) OIL: An ontology infrastructure for the semantic web, *Intelligent Systems, IEEE* (16:2), pp 38-45.
9. Gallaugher, J. (2010), Crowdsourcing, in: *Information Systems: A Manager's Guide to Harnessing Technology*, Flat World Knowledge.
10. Grady, C., and Lease, M. (2010) Crowdsourcing document relevance assessment with Mechanical Turk, *Proceeding of the NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk*.
11. Gruber, T.R. (1993) Toward Principles for the Design of Ontologies Used for Knowledge Sharing, *International Journal of Human-Computer Studies*).
12. Hevner, A.R., March, S.T., Park, J., and Ram, S. (2004) Design science in information systems research, *MIS Quarterly* (28:1), pp 75-105.
13. Howe, J. (2009), *Crowdsourcing: Why the power of the crowd is driving the future of business* Three Rivers Pr.
14. Hu, J., Wang, G., Lochovsky, F., Sun, J.-t., and Chen, Z. (2009) Understanding user's query intent with wikipedia, *Proceeding of the the 18th international conference on World wide web*, ACM, Madrid, Spain, pp. 471-480.
15. Kirkpatrick, D. (2008) Help Wanted: Adults on Facebook, Fortune.
16. Little, G. (2010) Exploring iterative and parallel human computation processes, *Proceeding of the*, ACM, pp. 4309-4314.
17. Noy, N.F., Chugh, A., Liu, W., and Musen, M.A. (2006) A framework for ontology evolution in collaborative environments, *The Semantic Web-ISWC*), pp 544-558.
18. Parameswaran, A., and Polyzotis, N. (2011) Answering Queries using Humans, Algorithms and Databases, *Proceeding of the*, CIDR.
19. Petrie, C. (2010) Plenty of room outside the firm, *IEEE Internet Computing*, pp 92-96.
20. Schall, D., Dustdar, S., and Blake, M.B. (2010) Programming Human and Software-Based Web Services, *IEEE Computer* (July).
21. Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. (2008) Cheap and fast---but is it good?: evaluating non-expert annotations for natural language tasks, *Proceeding of the*, Association for Computational Linguistics, pp. 254-263.
22. Sorokin, A., and Forsyth, D. (2008) Utility data annotation with amazon mechanical turk, *Proceeding of the Computer Vision and Pattern Recognition Workshops*, pp. 1-8.
23. Stojanovic, L., Maedche, A., Stojanovic, N., and Studer, R. (2003) Ontology evolution as reconfiguration-design problem solving, *Proceeding of the the 2nd international conference on Knowledge capture*, ACM, p. 171.
24. Von Ahn, L., and Dabbish, L. (2008) Designing games with a purpose, *Communications of the ACM* (51:8), pp 58-67.
25. Woods, D. (2010) Steps To Creating A Flash Company, Forbes Magazine.