EXTRACTING ONTOLOGICAL STRUCTURES FROM COLLABORATIVE TAGGING SYSTEMS

THE UNIVERSITY OF
SYDNEY

A thesis submitted in fulfillment of the requirements for the
degree of Doctor of Philosophy in the School of Information Technologies at
The University of Sydney

Winston Huairen Lin

August 2012

I have examined this thesis and attest that it is in a form suitable for examination for the degree of Doctor of Philosophy.

(Joseph Davis) Principal Adviser

Preface

Where Dreams Meet Reality.

Acknowledgements

To begin with, let me say that completing my PhD was challenging in and of itself. After all, if it were not, there would not be much value in the education and this education brought me great value. In addition to the scholastic challenges, I experienced life challenges. To pursue my PhD I moved to a country foreign to me and my wife and I had our first child during my studies. I cannot imagine how much more difficult my education would have been were it not for your assistance and support.

My gracious thanks also go out to my supervisor, Prof. Joseph Davis for his wisdom and guidance. Without his encouragement and constant engagement, I would never have completed my PhD studies. His critical and insightful feedback considerably improved the quality of my work. Special thanks to Dr.Ying Zhou for her useful comments and inspiring discussions throughout my thesis journey.

Thanks to all the past and present members of the "Knowledge Discovery and Management Research Group" at the University of Sydney for their contributions. I would also like to thank Dr. Simon Poon for his helpful input. I feel incredibly honored and humbled to be part of this dynamic and effective team.

Last but certainly not least, I thank my family. My heartfelt gratitude goes to my parents for all the love and support they've given me. A special tribute to my understanding wife Leily, without her generosity, encouragement, and loving support, I would not have had the peace of mind or focus necessary for my thesis accomplishments.

Publications

1. **W.H.Lin** and J. Davis, "OntoAssist: leveraging crowds for ontology-based search" (Demo paper) in the Proceedings of The 12th International Conference on Web Information System Engineering (WISE 2011), October 13-14, 2011, Sydney, Australia (ERA A)

2. J. Davis and **W.H.Lin**, "Web 3.0 and Crowdservicing" in the Proceedings of the Seventeenth Americas Conference on Information Systems (AMCIS 2011), August 4th-7th 2011, Detroit, Michigan, USA (ERA A)

3. **H.Lin**, J. Davis and Y.Zhou, "Ontological Service Using Crowdsourcing" Proceedings of the 21st Australasian Conference on Information Systems (ACIS2010), December 1-3,2010, Brisbane, Queensland, Australia (ERA A)

4. **H.Lin** and J. Davis, "Computational and Crowdsourcing Methods for Extracting Ontological Structure from Folksonomy" in the proceeding of 7th Extended Semantic Web Conference (ESWC 2010), May 30-June 3, 2010, Heraklion, Crete, Greece (ERA A)

5. **H.Lin** and J. Davis, "Exploiting Tag Relations to Improve Search in Collaborative Tagging Systems" (Demo paper) in the proceedings of the 4th IEEE International Conference on Semantic Computing (ICSC 2010), September 22-24, 2010, Carnegie Mellon University, Pittsburgh, PA, USA

6. **H.Lin**, J. Davis, and Y. Zhou, "Integration of Computational and Crowd-Sourcing Methods for Ontology Extraction" in Proceedings of the 5th International Conference on Semantics, Knowledge and Grids (SKG 2009), 2009 Zhuhai, China (ERA C)

7. **H.Lin**, J. Davis, and Y. Zhou, "An Integrated Approach to Extracting Ontological Structures from Folksonomies" in the proceeding of 6th European Semantic Web Conference (ESWC 2009), May 31-June 4, 2009, Heraklion, Greece (ERA A)

Abstract

The World Wide Web has undergone significant evolution in the past decade. The Web in its present form (often referred to as Web 2.0) is a major shift from the largely exposure-based features of Web 1.0. Also known as the social web or the read-write web, Web 2.0 introduced the critical feature of user contribution. Its impact has been massive in the rise of a vast array of social media sites and applications. However, our ability to access and use such content is somewhat limited. There is a need for new and innovative approaches to organising and retrieving online information in general and user-contributed content in particular.

Recently, folksonomy has emerged to help users share web-based information created by users, allowing users to organise resources using their own tags. However, our ability to search for information based on folksonomies is somewhat limited. This is largely because of its flat, non-hierarchical structure combined with tag vocabulary that largely consists of terms that are typically not found in dictionaries or thesauri. A promising solution that can transform a collection of tags into a queryable semantic web knowledge base is to build ontologies from the folksonomies. Our goal is to extract an ontological structure

from a folksonomy and facilitate its ability to evolve automatically as usage patterns change. We demonstrate that the resulting structure is significantly more efficient at supporting semantic-based exploration and search of online resources.

This thesis explores two questions. First, can knowledge be discovered in folksonomies and transferred into lightweight ontological structures using traditional automated computation? Second, how can ontological structures evolve and improve with end-user knowledge that has been solicited through crowdsourcing activities?

To address these two questions, we developed a new framework, termed "Ontological Structures Extraction 2.0". Our goal is to merge the useful aspects of ontologies and folksonomies. By extracting an ontological structure from the tags collected in a folksonomy, we can add explicit semantics to Web 2.0 applications, and use the knowledge of search engine users to help build semantic web structures. Specifically, our model does an initial automated extraction by exploiting the power of low support association rules mining supplemented by an upper ontology such as WordNet. Also, it integrates the knowledge of search engine users to help evolve the extracted ontology with the employment of crowdsourcing.

We implemented a semantic search application called SmartFolks to test semantic searches done on the extracted ontological structure. We also developed and tested a prototype hybrid human-machine system, OntoAssist. By piggybacking OntoAssist with an existing search engine, users can refine their online searches by choosing the relationships between query keywords and relevant

terms presented in the search results. This helps the initial ontology to evolve as well as providing better search results.

The automated algorithm returned promising initial results using two datasets from Flickr and CiteULike. We evaluated SmartFolks with a test dataset of 25,000 images from MIR Flickr. Comparing SmartFolks with benchmarks from MIR shows that semantic web technology improves user search experience and information retrieval. Two important, labour intensive tasks in ontology development are domain term selection and relationship assignment. We assessed the ability of non-experts to contribute to the ontology by engaging workers from Amazon Mechanical Turk (MTurk) to use our OntoAssist search tool. The experiments were completed in a short time at low cost with more than 90 percent accuracy. The OntoAssist tool is based on Yahoo! Search BOSS API and is available at the demonstration website www.hahia.com.

The evidence we submit indicates that knowledge from flat folksonomy structures can be extracted and enriched. This is a sound approach for solving the semantic search problems in collaborative tagging systems and for improving the precision and quality of information retrieved from the World Wide Web.

Contents

List of Tables

List of Figures

List of Acronyms

ACM The Association for Computing Machinery

AJAX Asynchronous JavaScript and XML

AMT Amazon Mechical Turk

API Application Programming Interface

ARM Association Rule Mining

CCS Computing Classification System

CTS Collaborative Tagging System

ESWC The Extended Semantic Web Conference

HIT Human Intelligence Task

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

IEEE The Institute of Electrical and Electronics Engineers

IR Information Retrieval

JSON JavaScript Object Notation

NLP Natural Language Processing

OSE Ontological Structures Extraction

OWL Web Ontology Language

RDF Resource Description Framework

SNA Social network analysis

SPARQL SPARQL Protocol and RDF Query Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

WWW World Wide Web

Chapter 1

Introduction

The Internet and the World Wide Web (abbreviated as WWW and commonly known as the Web) have grown rapidly in past decades. The web provides a rich medium to publish information, going beyond the traditional communications media of radio, television, and newspapers. It has revolutionized the way in which information is gathered, stored, processed, shared, and used (Zhong et al. 2002). Online information has become an ingrained part of our lives.

A well-organised framework for organising and retrieving information on the World Wide Web is essential if online users want to easily and quickly retrieve knowledge. Folksonomies and ontologies are two different ways to organize the knowledge present in the current Web (Echarte et al. 2007). Folksonomies describe the kind of informal social classification employed in CTS, where users describe and classify content with their own language or terminology Vander Wal (2007). Ontologies are formal structures for knowledge sharing and reuse, providing a common understanding between humans and

machine applications (Fensel et al. 2005). Each has its strengths and weaknesses. This introductory chapter outlines the progress made in the past three decades to categorise information using folksonomies and ontologies. A new approach is needed to overcome the problems found in these two ways of structuring knowledge. We discuss questions underlying the overall objectives of this research, and present an outline of the organisation of this thesis.

1.1 Why Does Information Classification Matter?

The World Wide Web has undergone significant evolution in the past decade. We are entering an era where people are increasingly connected on the Internet through Web 2.0 and related applications. The current Web 2.0 represents a major shift from the largely exposure-based features of Web 1.0. Also known as the Social Web or the read-write web, 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. According to The International Telecommunications Union, the total number of Internet users in the world reached 2 billion in 2010. As of July 2011, over 131 million websites operated, reported by domaintools.com. [1] (See Figure 1.1, an image adapted from: http://wemtech.wikispaces.com/) There is a wealth of online content that is generated by users of applications to create and manage videos, images, music, and other information. However, our ability to access such user-generated content is somewhat limited. There is a need for new approaches to the organisation

[1] http://www.nasdaq.com/markets/ipos/filing.ashx?filingid=7161599

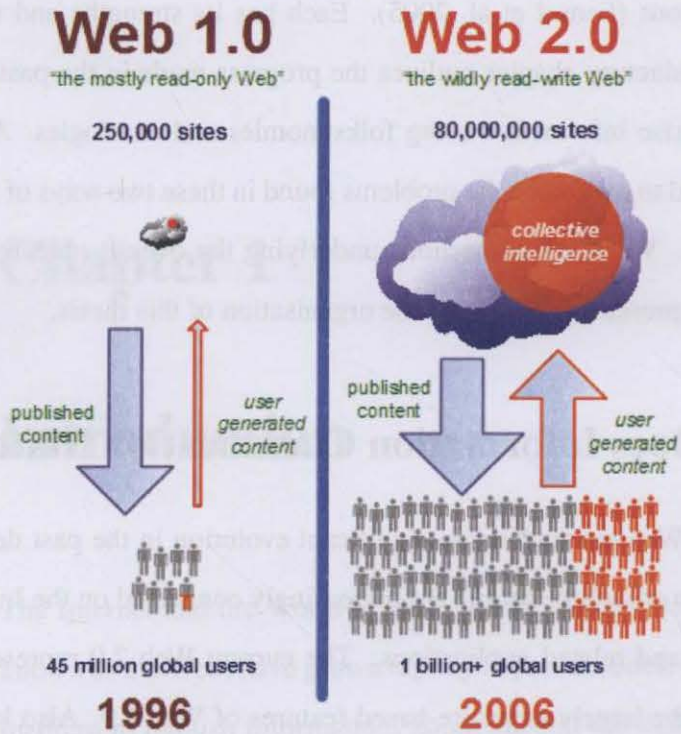


Figure 1.1: Web 1.0 vs. Web 2.0, adapted from http://wemtech.wikispaces.com/

of information online.

Information retrieval (IR) is the area of study in computer science that deals with searching for information within documents, relational databases, storage, and the World Wide Web (Salton and McGill 1986; Davies et al. 2009a). The essential challenge in information retrieval is the design and consistent update of a meaningful classification mechanism which provides a systematic organisation of knowledge (Quintarelli 2005). When creating meaningful representations of knowledge, the classification system should clearly show not only the relative location of a specific resource, but also viable routes to other resources

(Jacob 2004). By describing, indexing, and classifying social media resources, a classification mechanism provides a way to enable effective sharing, search, and exploration of a large percentage of online content.

1.2 What Makes a Folksonomy a Popular Choice?

Collaborative tagging systems (CTS), also known as social tagging, have recently emerged in order to help organise user-generated content. CTS allows users either to upload their own resources and annotate them, or annotate resources on other websites, in their own language and based on their own understanding of the content. For example, Flickr (Flickr.com) is an application that uses CTS for the management and sharing of online photos. Users can easily upload photos from their mobiles, desktop, or even from email. As of September 2010, 3,000 images per minute are being uploaded to Flickr and more than 5 billion images are hosted on that website [2]. As of July 2011, it has attracted more than 90 million [3] visitors per month. Flickr is currently ranked 31^{st} by Alexa [4] among all web sites for averaging number of visitors per day [5]. Another example is CiteULike (citeulike.org), a free online bibliography manager allowing users to gather, organise, and share scholarly papers. CiteULike has become popular especially among researchers and other academic users.

The kind of informal social classification employed in CTS, where users

[2] http://blog.flickr.net/en/2010/09/19/5000000000/

[3] http://www.ebizmba.com/articles/web-2.0-websites

[4] Alexa is the leading provider of free, global web metrics. Alexa has built an unparalleled database of information about sites, including ranking, statistics, Related Links, and more.

[5] http://www.alexa.com/siteinfo/flickr.com

Figure 1.2: Source from flickr.com, images are easily uploaded and available on the web through various tools

describe and tag content with their own language or terminology, was first recognised as a 'folksonomy' by Thomas Vander Wal in July 2004 (Vander Wal 2007). When a user tags an online resource, s/he is creating an informal taxonomy. These tags are aggregated to help searchers find the information they represent. With bottom-up, user-driven, and freely chosen vocabularies, folksonomies stand in contrast to taxonomies, which use controlled terms. The relationships among the terms in a taxonomy are typically contributed by domain experts.

1.3 Is a Folksonomy Good Enough?

As the amount of resources annotated using folksonomies increases, exploration and retrieval of the tagged resources pose challenges. The major problem

with folksonomies is that the tags used to describe the content can be idiosyncratic and not understood by many users. Most tags are chosen based on individual users' own experiences and linguistic styles and preferences. Furthermore, the concepts and internal structures of folksonomies are not explicit to machines or to other software applications, even though the tags may be meaningful and coherent to the users who created them (Euzenat and Shvaiko 2007). Folksonomies tend to include all kinds of tags, ranging from standard dictionary words, to compound expressions created by users, to jargon and nonsense words (Lin et al. 2009). Due to a flat, non-hierarchical structure consisting of unsupervised vocabulary, applications that employ CTS currently offer very limited search function support in their browsing interfaces (Hotho et al. 2006). We analyse the typical problems and limitations of folksonomies and ontologies in chapter 2.

Because of the huge volume of user created data and massive demand for improved quality of search, the research community is now focusing on services based on folksonomies. Various solutions have been proposed to improve the quality of folksonomy-based search. One stream of research has attempted to refine query results using meaningful knowledge derived from the folksonomy itself. Clustering and tag clouds are widely used techniques. Clustering groups search results into several subsets and recommends related resources based on selected tags. However, clustering methods rely heavily on statistical associations or co-occurrence of tags. The effectiveness of this approach can be limited as the derived relationships are unlikely to be based on meaning. A cloud is a somewhat rough approach to organising tags. It is a graphic representation of

Figure 1.3: Explore Flickr through Tag cloud.

tags shown in sizes relative to their frequencies, making it easy for the user to see the "hot" keywords. Tag clouds normally contain very general terms, such as "travel" or "wedding" and do not indicate any semantic relationships between the tags. See Figure 1.3 for an example.

1.4 What Makes Ontology an Attractive Alternative?

Developments in semantic web technologies offer us a new approach to managing information online and overcoming the limitations of keyword-based search in CTS. The semantic web approach attempts to transform the World Wide Web into a repository containing semantic annotation of the contents of web resources that can be processed more effectively by a machine. In this vision, an ontology enables many semantic applications, including semantic search. An ontology provides controlled vocabulary for the classification of content and

a shallow representation of the information space (Guarino et al. 2002; Vallet et al. 2005). It defines a set of representational primitives with class hierarchies and relationship rules among them. This serves as a framework for a domain of knowledge (Gruber 1993).

Using an ontology as the knowledge base, semantic search can provide several improvements over classic keyword-based search. These include: (1) better recall when querying for class instances; (2) better recall by using class hierarchies and rules; (3) better precision by using query weights; (4) better precision by using structured semantic queries; (5) better precision by reducing polysemic ambiguities through the use of instance labels and the classification of concepts and documents (Castells et al. 2007).

For example, a semantic search system processes a query against the ontology and returns a set of instances and some closely related terms, based on the class hierarchies and inference rules. Then, the query is expanded by a new set of query keywords from terms in the ontology, leading to higher recall values. The results can be further ranked based on the weight of the terms by calculating the distance between the initial query keyword and the related terms in the ontology. We can thus find both highly relevant and related resources with the great precision afforded by the semantics that are encoded into those ontologies. In addition, advanced resource navigation and browsers can provide a better search experience for users as a whole.

Taking the query keyword "apple" for an example, the existing keyword-based search engines would simply return a list of all the web pages in which the term apple appears. The search results are not guaranteed to be relevant to

the intent of the query. A semantic search system is capable of using the term's contextual background to classify web pages based on different meanings of the term. If the user is interested in information on Apple Computer, only a limited number of highly relevant pages will be provided. The system can further expand the results to a more specific class, such as Mac, or to an individual model, such as MacBook Air, even if the words "Mac" and "MacBook Air" are not present in the documents. Furthermore, results belonging to 'apple' in the sense of fruit will be removed, leading to a higher precision output.

However, the performance level of the semantic search is in direct relation to the quality of the backend ontologies, expressed by things such as coverage of a specific domain, and how well those domain terms are organised into a framework by type, structure, rules, and relationships.

1.5 Research Problem

In light of the recent explosion of and interest in user-generated content and social media, the current models of folksonomy and ontology no longer suffice. They focus on a single dimension of the web – folksonomies are based on user-created, uncontrolled tags with flat structures, while ontologies are built from semantic relations among core concepts as defined by experts. Both models ignore the interdependencies between user-generated vocabulary and knowledge in a domain of interest. Consequently they are not able to provide an ontology that has significant coverage and depth in the relevant domain. Moreover, an ontology built using traditional methods may not be well-matched to the needs

of the typical online user. Not only does it ignore the non-standard words (those not found in a dictionary) that are widely used today, but it also frequently contains antiquated terms that are no longer employed in online search.

The world we live in is not a static world, but one where information changes constantly and people are always on the move. Ontologies need to be frequently updated to compile the new knowledge emerging from the daily experiences of online users. But updates done manually by experts simply cannot keep up with changes in this era of Web 2.0 (Braun et al. 2007). Here is a example of the rapidity of change in vocabulary: The compound word "website" took more than 15 years to be accepted as an alternative to "web site" in the Oxford Dictionary. By contrast, it took only two years for the social media acronyms "OMG" and "LOL" to find their way into the dictionary. Also, with classic methods, it is not possible to establish a single and unified ontology as a semantic backbone for a large number of distributed web resources. Moreover, the manual annotation of resources requires skilled professionals or ontology engineers (Wu et al. 2006b). In other words, it is still an unsolved problem to convert the huge amounts of resources annotated in CTS into formal ontological structure at an affordable cost (Vallet et al. 2005). Many significant challenges have to be overcome before we can achieve mature semantic search using CTS.

At this point, semantic web ontology development is primarily based on the manual efforts of skilled experts and professionals. However, the continued collaboration and open innovations in web technology are causing changes. Ontology construction is based more and more on shared community platforms where the utilisation of collaborative editing solutions is an important driver.

Furthermore, as we discussed above, the advances in Web 2.0 and the semantic web offer new research opportunities and challenges related to the classification of concepts and creation of architectures. These advances influence ontology development on several levels. The semantic web is rapidly providing greater ontology-based functionality in applications. This suggests that as new knowledge bases mature, ontologies must deal with and manage large-scale user participation. Other challenges are that the new ontologies carry inherently different sets of human errors, and they require rapid aggregation during the development process.

Machine computation is a powerful method that is commonly used to guide knowledge development, in particular information extraction or ontology learning from text and other resources. But in a web context, a more user-oriented view has been emerging for some time. Crowdsourcing is one of the most influential means to encourage open innovation and to solve problems. It involves outsourcing a job traditionally done by experts to non-experts, typically a large group of people, in the form of an open call (Howe 2006). It has now become a generic expression for a wide range of endeavours on the Internet, including distributed problem solving, open innovation, and market trends prediction. Implementation of mass collaboration in human computation and problem solving is cheaply and efficiently done by means of crowdsourcing services such as Mechanical Turk, among others (Eckert et al. 2010; Franklin et al. 2011; Kittur et al. 2008).

Along with a human-oriented view come questions regarding the application of a computer-oriented approach to processing the input to the ontology.

Which is the best approach to guide and develop ontologies – through the power of the machine or the knowledge of human users? Are they complementary to each other in an integrated approach where machines and humans work on different aspects of the development? If we use a folksonomy as the input for extracting ontological structures, is that any different from processing keyword text? Another aspect of a traditional approach is that decisions often are made by brainstorming within the group. In the crowdsourcing approach, making decisions this way may lead to problems, especially when there are thousands of users and some of their judgments are conflicting.

Extracting ontological structures from folksonomies can be done with the integration of computational and crowdsourcing methods. But it is important to focus on strengthening the interdependencies between folksonomies and ontologies so as to guarantee that the resulting semantic web ontology reflects the needs of users, and continues to evolve with new terms contributed by online users. At the same time, the structure should retain some of the attractive properties of a classic ontology. In particular it should be able to answer more complex queries from online users.

1.6 Research Questions and Approach

The aim of this thesis is to develop and test methods to extract an ontological structure from a folksonomy and facilitate its automatic evolution. The challenge is to build it in such a way that the resulting structure can better support semantic-based searching and browsing of online resources, even with

constantly changing usage patterns. The task corresponds to the three major issues in designing search functions: (1) How to use CTS to build an ontological structure that supports effective search and exploration, (2) How to gather the information and design a model that enables the ongoing evolution of such a structure, and (3) How to test and validate the proposed approach.

In general terms, this thesis explores the unification of the seemingly exclusive features of folksonomy and ontology. An integration of the two allows us to achieve complementarities by providing all the advantages of colloquial terms from the folksonomy and semantic relationships from the ontology. On the folksonomy side, we can exploit the semantic relations in the ontological structure to satisfy queries or navigation requests in the terms and language that are familiar to the users. On the machine-computational ontology side, we can access, translate, and integrate millions of resources from different annotated social media applications.

Our research contributes to solving both theoretical and practical problems in the integration of user-generated content through the use of Web 2.0 / semantic web technologies. Notable gaps in the literature relating to this topic have caused us to raise the following questions.

1. How should we extract shared vocabularies from large tag collections?

2. How can we find the semantic relationships for these shared vocabularies?

3. What is the best way to handle the non-standard words in folksonomies?

4. How can an ontology be evolved to reflect a fast-changing environment, including changes in both knowledge and term usage?

5. What are the advantages of a crowdsourcing method, and what are the supports and barriers to effective design and implementation of tasks that employ human intelligence in the ontology evolution process?

6. What incentives will attract millions of people to work collaboratively and contribute to the evolution and refinement of the ontology?

7. Is it possible to provide a semantically enriched search that integrates the power of the machine with the wisdom of the crowd?

The research carried out and presented in this thesis addresses the questions above. We summarise our approach below:

In this work, we propose to integrate automatic computation with human intelligence to extract ontological structures from folksonomies. Our proposed approach provides a service that processes and combines knowledge input from users, tags, social media, ontologies, and semantic search. Figure 1.4 shows the proposed framework that binds together the computational power of machine, crowd-sourced human intelligence, and semantic search services. The characteristics of our approach include the following:

1. Using the computational power of the machine to induce a preliminary structure from CTS. We employ data mining techniques, such as association rules mining, to extract knowledge from folksonomies and then combine it with the relevant terms from an existing upper-level ontology.

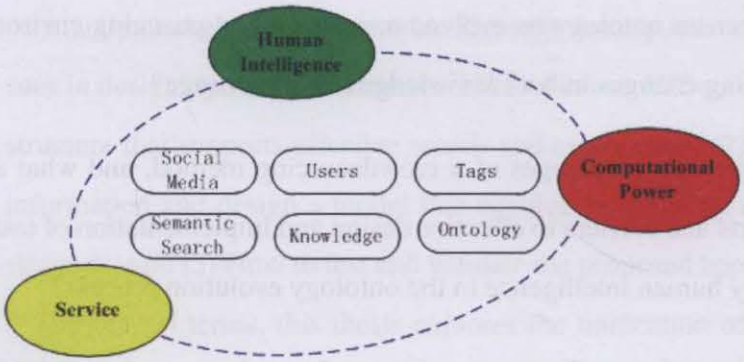


Figure 1.4: Service-oriented Ontological Structures Extraction 2.0

2. Channelling online users to evaluate and improve the structure. We further investigate the practicability of continuously updating the preliminary ontological structure from the inputs provided by online users.

3. Employing a purpose-designed platform to support the crowdsourcing flow, while providing interactive semantic search services as a motivation to the online users. We integrate the data into a search engine that can elicit knowledge from many online users for purposes of collaborative ontology evolution and refinement, as well as providing ontology-based search and exploration services.

In particular, the following methods are employed in our approach:

- Automatic computation

Association rules mining (ARM) is a data mining technique that is useful in a variety of tasks, in particular for discovering relationships among items from large datasets. The most famous algorithm for finding association rules is the

Apriori algorithm. We adopt ARM, specifically, low-support association rules mining, in the extraction process to analyse a large subset of a folksonomy. The extracted knowledge is expressed in the form of new relationships and domain vocabularies.

Previous research has attempted to semantically enrich folksonomies. However, most studies were based on a knowledge base that did not include non-standard terms (Angeletou et al. 2008a). We subdivided the folksonomy vocabulary into standard, compound, and jargon tags. A series of methods were tried to incorporate the tags. First, we fortified an existing knowledge base, namely, WordNet (Miller 1995). Standard tags in the vocabulary were mapped to WordNet to get semantic relations with the Apriori algorithm. Next, the non-standard tags, which included jargon and user-defined compound words extracted from the folksonomy, were then incorporated into the hierarchy.

With this integrated computational method, the hidden knowledge embedded in the folksonomies is transformed into formalised knowledge in the form of an ontological structure.

- Crowdsourcing

Another question is how to elicit and aggregate the intelligence of online users to build good ontologies via crowdsourcing. Since computational techniques have limitations, we attempt to find an alternative way to build ontologies using human input. This is seen as a more robust alternative to having in-house teams of experts or a chosen group of contributors solve the many and varied

problems. The basic assumption is that the crowd can bring interesting, non-trivial, and non-overlapping information, insights, or skills. These can add to the quality of the solutions when harnessed through appropriate aggregation and selection mechanisms (Davis and Lin 2011).

- Prototype: An hybrid Human-Machine System

In order to introduce a sustainable motivation level and to expand the source of labour from paid workers to a broad range of public Internet users, we further develop a framework for blending ontology evolution tasks seamlessly with public users' daily search activities. With this design, our proposed crowd-sourcing approach can get actual users involved without necessarily offering a monetary reward. Our design allows users to refine their searches on web repositories by choosing relationships between query keywords and relevant terms in the search results. With a few simple clicks, an online user helps the initial ontology to evolve, while providing better search results for that particular query.

To demonstrate the framework, we built OntoAssist, a semantic navigation tool. It enhances the native search in CTS, giving users a smart and user-friendly search engine. In particular, the disambiguation feature helps users to search more effectively. At the same time, user input to clarify term meanings is collected to help evolve the underlying ontology. On top of that, OntoAssist can be integrated with third-party commercial search engines and portals such as Google Search, Bing, or Yahoo! Search, using their APIs. As an example, the OntoAssist tool was implemented based on Yahoo! BOSS and released at www.hahia.com. It thus has the ability to provide semantic search and explore

most existing resources in CTS.

- Experimental Evaluation

We report the results of a set of experiments designed to demonstrate the ability to integrate the knowledge from folksonomies and ontologies in a way that achieves a higher level of ontological service quality than could be achieved under each structure alone. Our first experiment, which uses an automated algorithm, has produced promising initial results using two datasets from Flickr and CiteULike. Next, we evaluate ontology-based search of CTS with our SmartFolks application, using a 25,000 image dataset from MIR Flickr as our test data. By comparing our findings with the manually annotated benchmark provided by MIR, our experiment shows that the technology of semantic web can help improve the quality of user search experience and information retrieval. After that, we assess the ability of non-experts to solve two main labour-intensive tasks in ontology development – domain term selection and relationship assignment. We asked workers from Amazon Mechanical Turk (MTurk) to reproduce a variety of ground-truth ontologies. The experiments were completed in a short time, at low cost, with more than 97% accuracy. And last, the OntoAssist tool has been implemented based on Yahoo! BOSS API and released at www.hahia.com. OntoAssist is currently available online as a demonstration to discover new terms and facilitate rapid ontology evolution. Promising experimental results and analyses are reported and analysed.

1.7 Contributions

1.7.1 Ontological Structures Extraction 2.0

We describe a new integrated algorithm, called Ontological Structures Extraction 2.0 (OSE 2.0), with the following innovative properties:

1. We propose a framework and a collection of algorithms that generate domain terms and relationships using data mining of human knowledge repositories. The extraction process is empowered by association rules mining, upper ontologies, and natural language processing. It effectively organises the knowledge found in folksonomies into ontological structures. Standard tags, jargon tags, and compound tags are included in the ontology. This is important since previous work in this area has generally gained knowledge from standard tags alone, which represent only around 50% of tags created by web users.

2. We propose a crowdsourcing method as an alternative means of building the ontological structure. Our design contributes to an enhanced understanding of how to apply and extend established crowdsourcing theory and methodologies in order to improve ontology development capabilities. Our theoretical exploration of how and under what conditions crowdsourcing can be effective can also be applied to other tasks.

3. We conceptualise an open innovation platform that integrates the above-mentioned approaches to steer community concepts into the ontology. It

provides the community with advanced ontology-based search and exploration functions, as well as enabling the community to share and develop concepts that facilitate ontology evolution.

(a) The semantic search system serves as a proxy to attract users and assign a problem or the distribution of some work to a large number of independent individuals over the Internet.

(b) Using an ontological search service as a motivation enables and encourages community contribution. It benefits users by disambiguating the intent of their semantic queries. The searchers themselves accomplish this by selecting domain terms that are relevant to the query keyword and declaring the relationships between the terms. With this design, our approach attracts the participation of the crowd without necessarily offering monetary reward.

(c) We present an ontology evolution process that furthers knowledge acquisition on the strength of elucidating the semantic search intent underlying common queries. User input during the disambiguation process is collected and aggregated to find new concepts from the community, thus facilitating expansion and revision of the ontology.

(d) Our crowdsourcing approach relies on actual users instead of domain experts, and embeds ontology evolution seamlessly into the daily search activities of the general public.

In short, the essential idea of our OSE 2.0 algorithm is to extract preliminary ontological structures from folksonomies, and then provide a purpose-designed platform to keep evolving these structures while users consume the search service.

1.7.2 Crowdsourcing and Crowdservicing

This thesis investigates several mechanisms that can be applied to address the new challenges of ensuring standards for quality control while interacting with online workers. Gold standard data was used to test the participants in the tasks and to exclude cheaters. Gold standard data were questions in the HITs for which we knew the answers. We prevented workers from continuing the work if they were unable to correctly answer most or all of these questions. Measure of agreement was the second mechanism. It collected redundant inputs and assumed that a large number of Turkers agreeing to the same answer meant the answer was correct. In addition to these two solutions, we also applied other techniques to normalise the datasets, such as soliciting comments from Turkers and continuously monitoring the input results while the HITs were occurring. These mechanisms and functions were organized into a quality control workflow presented in chapter 6. This workfow can be applied to other crowdsourcing research to assure high quality data.

We developed and implemented the concept of service as motivation for crowd-based solutions to complex tasks and problems. The success of any crowdsourcing approach relies on strong and sustainable motivation to attract a

sufficient number of human agents. Monetary reward is able to attract all sorts of participants, yet it is only feasible for short term projects, such as the early stages of building an ontology. Our design represents the next stage in the evolution of crowdsourcing models, which coined as crowdservicing (Davis 2011). It highlights an actual web application service maintained by the users themselves.

1.7.3 Semantic Search Implementation Experience

We describe prototype implementations of the algorithm in two semantic search applications for CTS. Regarding implementation, we found that it is feasible to enrich a folksonomy with an extracted ontological structure and thus improve search and exploration. This appears to be a sound basis for overcoming semantic problems in CTS by making the knowledge in folksonomies explicit.

One important principle that guides the design of the extraction algorithm is to accept and incorporate non-standard tags, including jargon tags and compound-word tags. Our implementation of the algorithm on both demonstrates that non-standard tags are as important as standard tags, and should be a part of user-generated knowledge that will support intelligent access.

We describe some implementation choices that have had major impacts on system usability. In particular, we use Wikipedia-related terms to help users disambiguate their queries. This design, described in detail in chapter 7, ensures that a query can always be clarified with related terms from the larger knowledge base. Those related terms from Wikipedia will also be added to the

backend ontology with user-specified relationships.

We implement OntoAssist as a web service. This gives it the ability to integrate with existing major search engines and subsequently have access to most of the existing CTS by utilising the search engine's index. It also gives our service a large number of potential users.

Our approach is also a practice of blending crowd computation and automation into a hybrid human-machine system. It shows how integrating them achieves a level of service quality that cannot be achieved by each alone.

1.7.4 Ontology Evaluation Based on Crowdsourcing

Recruiting subjects in traditional experiments can be time-consuming and costly. We describe methods of data collection and evaluation when utilising crowdsourcing websites such as Amazon Mechanical Turk (MTurk). We present an experimental evaluation that is inexpensive, takes only a few hours, and is able to attract hundreds of users to work on the evaluation tasks.

1.8 Thesis Structure

This thesis is structured as follows.

Chapter 2 gives background information to describe the social networking developments of the World Wide Web, including Web 2.0 and the Semantic Web. The focus is on the backend technologies used in Web 2.0 (namely, folksonomies) and the semantic web (ontologies). It also compares the advantages

and limitations of these two approaches to classifying information.

Chapter 3 reviews research on folksonomies and current approaches to extracting knowledge from them. We review related studies, including both traditional methods and recently proposed collaborative or crowd- based efforts; discuss other relevant approaches to using human computation, such as "game with a purpose"; and compare their strengths and weaknesses in detail.

Chapter 4 presents theories, methodologies, and an overview of our integrated approach.

Chapter 5 describes the computational approach to building ontologies.

Chapter 6 presents a crowdsourcing alternative to help construct ontologies. We describe experiments that were conducted with Mechanical Turk (MTurk) workers, who used OntoAssist to add semantic information to keyword search results. Our findings are presented and evaluated.

Chapter 7 deals with the integration of human input gained from crowdsourcing into the construction of an ontology. We present our crowdsourcing model for evolving an ontology and detail the implementation of OntoAssist with the Yahoo! search engine.

Chapter 8 presents our conclusions, implications for researchers and users, and the limitations of our research. We comment on the direction for future research.

Chapter 2

The World Wide Web and its Social and Semantic Dimensions

2.1 Introduction

The World Wide Web ("WWW" or simply the "Web") is a global accessible information system that consists of all the public Web sites connected to the Internet worldwide. Two of the most significant, evolutionary trends in the context of the World Wide Web have been the Social Web and Semantic Web. Social web encompasses collaborative tagging systems such as delicious.com, flickr.com as well as social networking sites such as twitter.com, facebook.com. Semantic Web attempts to build a web of data that can be processed directly and indirectly by machine.

In this chapter, we provide a short history of the Web and discuss the development of its social and semantic dimensions.

2.2 History of the World Wide Web

2.2.1 Hypertext Concept

Information bases, as envisioned by Bush (1945) , Engelbart and English (1968), are fundamental to sharing information in large organization (Conklin 1988). Most people conceive of information as a collection of ideas that have been selected, organized and presented in a medium. A key element in this conception of information, from the perspective of both writers and readers, is structure.

As early as in 1945, Bush (1945)wrote a famous article in Atlantic Monthly about a photo-electrical mechanical device called a Memex, for memory extension. He discussed the problems of personal information storage and management and proposed that human can make and follow links between documents on microfiche as human brains make associations between things.

Nelson (1965) coined the word Hypertext in "A File Structure for the Complex, the Changing, and the Indeterminate". He proposed to build an up-to-date index of all the contents in the system which would accept large and growing bodies of text and commentary. The index would has an unlimited number of categories and hold commentaries and explanations connected with them. The machine-supported links (both within and between documents) are the essential feature of hypertext system which allows a nonlinear organization of text (Conklin 1988).

Engelbart set up his own Augmentation Research Center in 1963 and then developed an elaborate hypermedia—groupware system called NLS (oNLine

System). It was the first successful implementation of hypertext which was used for the the creation of digital libraries and storage and retrieval of electronic documents (Engelbart and English 1968). The demonstration of NLS in 1968 is still known as "the mother of all demos" since it presented a live video conference with staff members back in his lab 30 miles away and for the first time a computer mouse, hypertext, object addressing and dynamic file linking were used.

2.2.2 Development of the World Wide Web

In 1989, Berners-Lee (1989) proposed to create a global hypertext space where any network-accessible information could be refered to by a single "Universal Document Identifier". He wrote in 1990 a program called "WorldWideWeb" as a prototype of WWW, a point and click hypertext editor which ran on the "NeXT" machine.WorldWideWeb was a graphical point-and-click browser with mode-free editing and link creation. It merges the techniques of information retrieval and hypertext to make an easy but powerful global information system which would download and display linked images, diagrams, sounds animations and movies from anything in the large NeXTStep standard repertoire (Berners-Lee 1989; Berners-Lee et al. 1992).

The WWW has created a new information space that aims to allow information sharing within internationally dispersed teams, and the dissemination of information by support groups. As Berners-Lee (1998)said in his website:

"The dream behind the Web is of a common information space in which we

communicate by sharing information. Its universality is essential: the fact that a hypertext link can point to anything, be it personal, local or global, be it draft or highly polished. "

UDIs (now URIs), HyperText Markup Language (HTML) and HyperText Transfer Protocol (HTTP) are essential technologies for the development of the Web. The Web consists of document, links, and index. Documents are in hypertext format and contain tables, images, other presentational devices, and links to other documents. An index is a special document built for the purpose of search. HTTP is used to allow a browser to request a keyword search on the index and return a resulting document containing links to the document found. A reader can simply click on the link with a mouse to access to the desired document.

2.2.3 Growth of the World Wide Web

The WWW has grown rapidly in past decades. By 2002, many companies had their own public Web sites for publishing instant worldwide information. A number of web technology-inspired dot-com companies blossomed and became highly profitable. Traditional media such as newspaper publisher also found the Web to be a useful and profitable additional channel for content distribution. The WWW provides a rich medium to publish information, going beyond the traditional communications media of radio, television, and newspapers. It has revolutionized the way in which information is gathered, stored,

processed, shared, and used (Zhong et al. 2002). Online information has become an ingrained part of our lives.

2.3 Web 2.0 and Collaborative Tagging Systems

Recently, there has been a shift from just one-way publishing on the web to participating in a two-way "read-write" exchange. Thanks to the lower barriers to online contribution, a web user is now an active participant or publisher in the creation of user-generated content, instead of being a passive consumer of information (Breslin et al. 2009).

Web 2.0 is the outcome of changing trends in the use of World Wide Web technology that facilitated a publishing revolution in the online community .Web 2.0 is also known as the social web, due to its use for socializing and sharing information about common interests.

Today we have new opportunities for communicating and collaborating through web-based communities and services. The success of the social web is marked by rapidly increasing numbers of users and applications, including wiki-style collaborative editing, personal blogs, and online image and video-sharing sites, among many others. Web 2.0 applications provide an easy and free way to publish videos, images, music, news references, and bookmarks – all kinds of social media – online. See figure 1.2 for an example of one of the more popular social web applications, which lets people easily upload their photos any time and anywhere.

As a consequence of the large number of users and applications, the volume

of online content available has increased exponentially, giving us a wealth of useful information. This user-generated content provides real-time news and photos, and is an important everyday information source. However, it also presents the difficulty of managing the abundant data resources. Browsing and searching for something specific is not that easy. Today, the organization of digital resources is a major challenge.

2.3.1 Information Classification and Retrieval

Classification plays a vital role in information management, and helps improve the quality of searching (Qi and Davison 2009). Web page classification (also known as web page categorization) describes the process of putting a web page into a predefined category (a taxonomy). The traditional process of classifying web pages includes:

1. Determining the information architecture
2. Preparing categories and general terms
3. Building a site taxonomy
4. Annotating web pages with the terms.

The Association for Computing Machinery (ACM) Computing Classification System (CCS) [1] is a widely used standard that allows you to classify your work, usually academic papers, using a four-level tree consisting of categories and subject descriptors.

[1] http://www.acm.org/about/class/1998

The following is an example of CCS taxonomy:

Categories: H. INFORMATION SYSTEMS

H.2. DATABASE MANAGEMENT

H.2.3 Languages

Subject descriptor: Query languages

As you proceed from the root through the branches of the tree, you are browsing from general to more specific levels. For instance, $H. \rightarrow H.2$ shows a narrowing of the subject topic, from information systems to database management.

By describing, indexing, and classifying social media resources, a classification mechanism provides accurate categorization and facilitates effective sharing, search, and exploration of those resources. It enables quick content reference and navigation, and thus helps Internet users to accurately locate a complete set of resources.

2.3.2 Collaborative Tagging and Folksonomies

Recently, the collaborative tagging system (CTS) has emerged as a social web mechanism for organizing and sharing online information. CTS allows you to annotate your favorite websites using any keywords or tags relevant to the content. Social web applications that use CTS include Flickr [2], an online photo

[2]http://www.flickr.com

management and sharing application launched in February 2004 . Another example is Citeulike [3], a free online bibliography manager that allows the gathering, organizing, and sharing of scholarly papers. It is especially popular among researchers and those in academia.

The kind of informal social classification employed in CTS, where users describe and classify content using their own language or terminology, was first labeled a folksonomy by Vander Wal (2007). Folksonomies are based on tags, which are annotations that users make while creating or viewing pages on the web (Plangprasopchok and Lerman 2009). Annotation via tagging helps us to locate resources again at a later time by searching or browsing. A tag serves as a link or index to other relevant resources having the same tag. Many users may apply the same tag to a single resource, or the same tag may be used to describe several different resources. Making the tags public provides an easy way to access and share information. Tags are aggregated to help find the information they represent. Since this mechanism is the same for every user, you can find other references that have the same tag by clicking the tag, or by searching for tag keywords separately.

With bottom-up, user-driven, and freely chosen tag vocabularies, folksonomies stand in contrast to taxonomies, which use controlled terms with relationships that are typically defined by domain experts. Tagging is not a new concept, especially for librarians, who use tags to describe the content of books and group them into related categories. A folksonomy adds some new features to tagging:

[3] www.citeulike.org

- Tagging can be done by any online user, not just experts or librarians;
- Tags are usually simple descriptive terms used in everyday life;
- Tags are chosen informally and personally. There are no restrictions on defining a tag;
- Everyone can use as many tags as they like to label a resource;
- Tagging is a means for online collaboration. A resource can be tagged and annotated by many users, thus enabling users to interact with the resources provided by others (Echarte et al. 2007);
- Tagging stimulates content evolution. By making tags public online, every user can browse the resources collected by others and add new tags to them.

CTS has provided a convenient way to allow online users to collectively annotate and categorize large numbers of distributed resources from their own perspectives. However, the vast increase in the number of resources annotated using folksonomies poses challenges to exploring and retrieving those resources, due to their flat, non-hierarchical structures and their unsupervised vocabularies.

2.3.3 Typical Collaborative Tagging Systems

Collaborative tagging is also known as social tagging because collaboration occurs among users in a social environment. Accordingly, CTS is also known

as a social tagging system, and it has become one of the most popular features in Web 2.0 applications. Two types of Web 2.0 application leverage the power of folksonomies with CTS. Some, like Delicious [4] and Citeulike [5], employ CTS as their only service. Others, like Flickr and YouTube, integrate CTS as a means to organize the content created by their users.

- Flickr, Online Photo Management and Sharing Applications

Flickr is an example of CTS as an integrated function. Considered one of the best sites for managing and sharing photos online, Flickr allows you to upload pictures to its online storage and label them using tags. The tags help when organizing the photos and also when searching and sharing them online. With AJAX technology, more tags can be added to existing photos without refreshing the page.

- YouTube, the Largest Worldwide Video-sharing Community

Another example of integrated CTS is YouTube. Founded in February 2005, YouTube provides a forum that allows billions of people to upload, share, and watch videos. Like Flickr, tags are also used on YouTube to describe the uploaded content. To find a video, you can either type a keyword and search on the tags, or browse to see the most viewed videos, or click on the related topics that are presented when you search for a term.

- Delicious, a Social Bookmarking Service

[4] http://www.delicious.com/
[5] http://www.citeulike.org

Figure 2.1: Upload, share, and search photos on Flickr

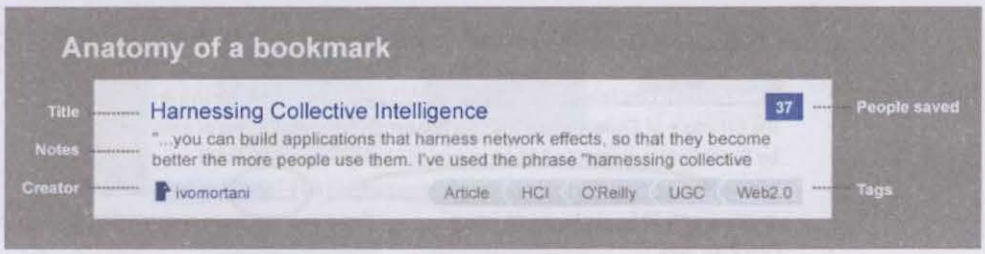


Figure 2.2: Anatomy of a bookmark in delicious.com

Delicious is a social bookmarking application and also an independent CTS. It allows you to tag, save, manage, and share all kinds of web pages in one place: music, photos, videos, and more. Unlike applications that integrate CTS with other services, such as video or photo storage, Delicious is strictly a tagging service: It bookmarks any website on the Internet with a stored URL. Many sites let you post their URLs to Delicious as a bookmark by clicking on a logo. See figure 2.2 for an example of a web page that has been tagged in Delicious.

- Citeulike

Whereas Delicious is for all kinds of web resources, Citeulike is a free tagging service for scholarly references only. For example, figure 2.3 shows a reference from Citeulike that has been annotated with these tags: collaborative, folksonomy, tagging, and structure. Once tagged, the reference is automatically indexed under these four classifications. Since this mechanism is the same for every user, you can find other similarly tagged references by clicking the tag, or by searching the tag keywords separately. In this example, 221 people assigned more than 200 tags to annotate this reference during the course of one year. If you search using any of these 200 distinct tags as keywords, you will find the

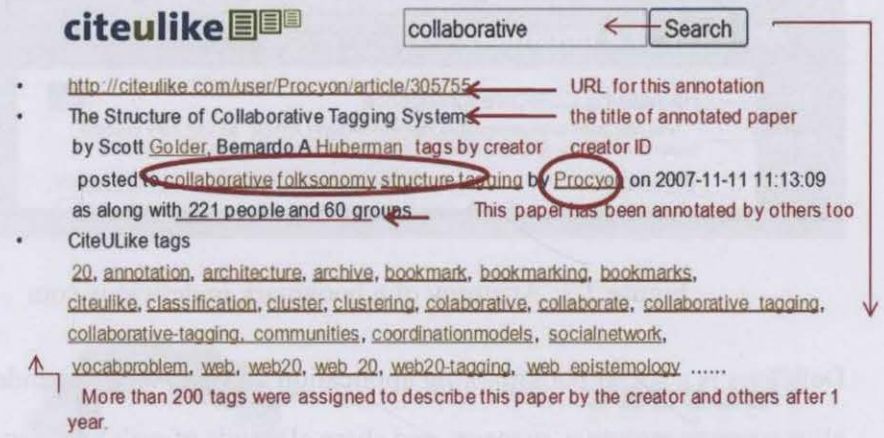


Figure 2.3: A reference to an academic paper in citeulike.org

reference.

2.3.4 Limitations of Current Search Technology in CTS

In spite of its advantages in annotating online data, the exponential increase in using CTS is posing major challenges for the exploration and retrieval of resources.

An analysis of tag data has found that 54.62 percent of tags consist of words that are not found in the dictionary. This suggests that the proportion of tags that have newly invented words or slang, misspellings, and possibly foreign words is indeed high (Suchanek et al. 2008b). According to our previous experiments, folksonomies tend to include all kinds of tags ranging from standard English words to terms created by individual users (Lin et al. 2009b).

Since search functions are based on comparing plain-text strings and then performing a match of query keywords to tag collections, various problems

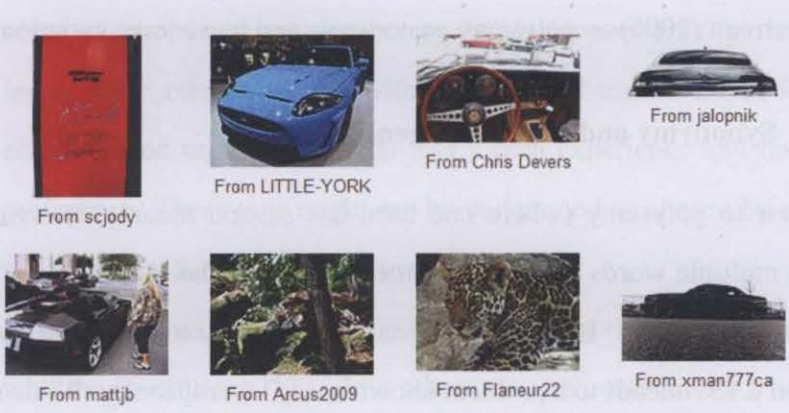


Figure 2.4: Polysemy of the tag jaguar and its search in Flickr

regarding information retrieval from CTS need to be resolved (Golder and Huberman 2005; Breslin et al. 2009; Bontcheva et al. 2006).

2.3.4.1 Polysemy and Tag Ambiguity

Golder and Huberman (2005) identified three kinds of problems in folksonomy: polysemy, synonymy, and basic level variation. Polysemy occurs when one tag has multiple, unrelated meanings. It is also known as tag ambiguity due to the lack of semantics in the text of the tag. For example, if a user does not disambiguate a query keyword like "jaguar", there is no way for a machine to understand whether the user is looking for resources about a Jaguar car, or jaguar the animal. The machine will thus locate all resources annotated with the tag jaguar and return both types of jaguar. The user then has to spend more effort reviewing the results to select the ones that have the desired content.

These kinds of problems in folksonomy have been summarized by Golder

and Huberman (2005) as polysemy, synonymy, and basic level variation.

2.3.4.2 Synonymy and Tag Heterogeneity

In contrast to polysemy (where one term has several meanings), synonymy refers to multiple words having the same or closely related meanings (Golder and Huberman 2005). If synonymous tags are used to annotate the same resources in CTS, it leads to a problem known as tag heterogeneity. While searching resources, your query keyword may not match synonymous tags that have been assigned to the resource by others. Users have to run several different queries to improve the search results. The usage of non-standard language is the main reason that this problem occurs. In order to quickly enter tag annotations, some users tend to abbreviate or combine two terms into a compound word to describe their meanings. For example, the abbreviations "USYD" and "SydneyUni" refer to the University of Sydney, and the non-standard compound "socialweb" refers to the social web.

2.3.4.3 Basic Level Variation

Searches can either bring forth relevant tags that are different terms with the same meaning (synonymy. like computer and PC), or tags that have the same term attached to resources with different meanings (polysemy, like Jaguar the car and jaguar the animal). Another problem that can lead to unsatisfactory results is basic level variation (Golder and Huberman 2005). By this we mean that people with different levels of knowledge and skill are tagging resources,

some of whom will use vocabulary that is very specific (perhaps scientific or legal) while others will tag with very general terms. In CTS, most tags are chosen based on the individual user's own experience and linguistic style or preference. These tags might not be understood or chosen by other users who are searching for that same resource.

For instance, some IT professionals may annotate a model of an Intel CPU with the specific tag "i7" while others may annotate it with a more general tag "CPU". The concept and relationship of the tags i7 and CPU are not explicit to the machine or to other systems, even though the tags may be meaningful and coherent to the users who assigned them (Euzenat and Shvaiko 2007). In this case, the machine is not able to return resources annotated with i7 to people who are looking for CPU, and vice versa.

2.3.4.4 Presentation of Search Results

Results from search engines are usually displayed as a vertically ranked list. Typically, one element in the result list contains a set of important metadata, such as a brief summary and source URL. The navigation of the results can be difficult and time-consuming. Most people do not go beyond the first page of results because it takes time to view the summaries. Thus, without proper organization or a guide to navigation, it is uncommon for users to learn what is beyond the first few results.

2.4 Semantic Web and Ontology-Based Systems

2.4.1 Ontology

Ontologies are formal structures for knowledge sharing and reuse, providing a common understanding between humans and machine applications (Fensel et al. 2005). The idea of ontologies emerged in the 20th century as a means to facilitate successful information exchange between different agents in the field of artificial intelligence (Gruber 1993). In Gruber's definition of ontology (an explicit and formal specification of a shared conceptualization), there are two key points: First, formal language must be used to describe the relationships between concepts in order to allow reasoning by computers; second, an ontology represents the shared points of view from a specific domain (Gruber et al. 1995). While taxonomies and ontologies are both hierarchical, containing terms in a specific domain and showing parent-child relationships, an ontology has more formal rules and more precise statements about the relationships of terms in a set of concepts. Reasoning make the ontology description more comprehensive.

The OWL [6] Web Ontology Language, informally OWL, has been recommended by W3C to represent the meanings of terms in an ontology and the relationships between those terms. (W3C released OWL 2 in October 2009). OWL is designed for use by machines and thus it has the ability to represent machine interpretable content on the web. OWL enables automatic referencing and the formation of content datasets. It makes the information associated with

[6] http://www.w3.org/TR/owl-features/

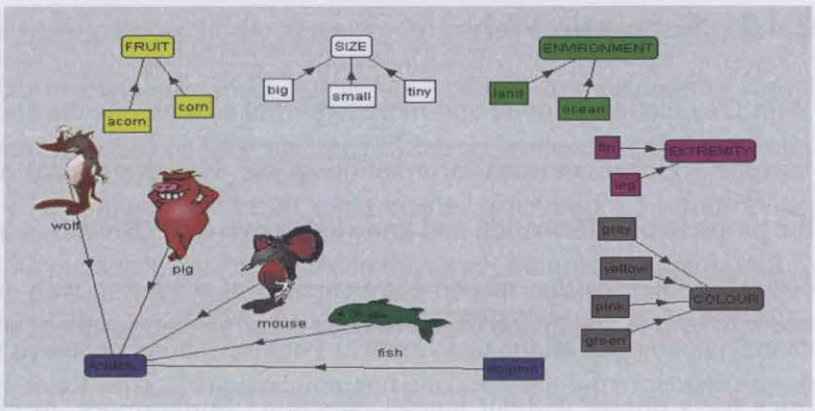


Figure 2.5: An example animal ontology retrieved from Webstructor

an ontology more amenable to machine processing and interpretation (Davies et al. 2009b).

It is expected that ontologies will bring the current web to its full potential by supporting the acquisition, maintenance, and access of semantic information. Adding meaning (semantics) to the web through ontologies will be particularly helpful in solving problems related to low search precision and poor resource navigation.

Figure 2.5 shows an example of an ontology that was built with Webstructor for the term "animal". In this ontology, animal has subclasses of wolf, pig, dolphin, etc.

Fig.2.5 shows an example ontology of animal from Webstructor [7]. In this ontology, animal has sub-classes of wolf, pig, dolphin etc.

[7] http://www.webstructor.net/worlds/animals.html

2.4.2 Semantic Web

With the creation of more and more hyperlinked web documents, the Web has become a global common information space. However, it has also amplified the problem of information and knowledge overload (Breslin et al. 2009). The central problem is that machine are capable of rendering web documents but cannot accomplish all the tasks such as booking a hotel or search for the lowest price for a printer without human direction. Furthermore, machine provides little support in human understanding, organizing the knowledge contained in the webpages because they are designed to be read by people, not machines.

Following the development of ontologies and the evolution of related web technologies, including HTML, XML, RDF, and OWL, Tim Berners-Lee coined the term Semantic Web and proposed it to be the next generation of the WWW. The semantic web is a web of data that have been assigned explicit meanings and can be directly or indirectly processed by machines (Berners-Lee et al. 2001):

"There was a second part of the dream, too, dependent on the Web being so generally used that it became a realistic mirror (or in fact the primary embodiment) of the ways in which we work and play and socialize. That was that once the state of our interactions was on line, we could then use computers to help us analyse it, make sense of what we are doing, where we individually fit in, and how we can better work together."

The main purpose of the Semantic Web is to drive the evolution of the current Web by enableing machines to "understand" and respond to complex human requests based on their meaning. Such an "understanding" requires that the relevant information sources is semantically structured, i.e resources are annotated with meta data and organized in the form of ontology. Ontologies formally describe the meanings of terms used in a web document. It provides formal semantics to all sorts of information and enabling the interoperability of humans and machines in managing information and sharing knowledge (Simperl et al. 2009; Fensel et al. 2005).

With the use of ontology, the Semantic Web allows web creators to provide metadata that is associated with web resources, and then further connected to each other with terms and relationships described in ontologies. Based on XML, the semantic web has the ability to define customized annotation schemes in an HTML document (a web page). RDF further models the resources and relations between the tags.

Semantic web technology has the potential to address many of the limitations in the current Web 2.0 applications. With a foundation of ontology, the semantic web will offer a richer representation of resources and the relationships among them, and thus will provide us with improved knowledge processing ability and more intelligent service (Gasevic et al. 2009). In this vision, we can effectively manage and access non-ontological resources by annotating them or mapping their representations to terms that exist in the ontological structure.

The use of ontology allows users to more precisely express their queries (Bontcheva et al. 2006). Today, one finds a resource on the web through searches

based on keyword matching (find the words in the text and then match them). One way to improve this is to tell the computer the meaning of the search. For instance, when we are searching on the word jaguar, there should be an annotation that jaguar is an animal and at the same time, the web pages about jaguar animals should be tagged as such.

In addition to annotation, another task is disambiguation of the search. A semantic search is more effective than today's keyword-based search because ontologies can improve the precision and accuracy of search results by looking for specific concepts and their related terms (Davies et al. 2009b; Berners-Lee et al. 2001). For example, when knowledge is structured in an ontology and a user searches for a picture of a wolf, the semantic search engine knows that s/he is looking for a certain animal. Further enhancements may be provided through extended filtering. For instance, the results could be organized into sub-categories like the color and type of wolf.

2.4.3 WordNet and other Knowledge Repositories

Existing ontologies can be reused for extending, specializing, or integrating with other ontologies. They also can serve as an upper ontology providing the base structure that is used to facilitate relationship construction. An upper ontology is an existing ontology that provides common knowledge across multiple domains.

wine, vino
=> alcohol, alcoholic beverage, intoxicant,
=> beverage, drink, drinkable, potable
=> food, nutrient
=> substance

Figure 2.6: A part of wine ontology from Wordnet

WordNet (Miller 1995) is a widely used upper ontology lexicon for the English language. Words are grouped into sets of synonyms, and WordNet provides various semantic relations between these sets. It is especially useful for displaying the hypernym relationships of both nouns and verbs.

For instance, the word wine has upper hypernyms (in the order from child to parent) such as alcohol and beverage. See figure 2.6.

The relationship hierarchy is expressed in terms of generality using interval notation. A semantic relationship between a word that is more general than others in the set is shown as

(parent-to-child or broader-to-narrower direction) $\supseteq$,

The less general (child) terms are shown as $\subseteq$,

and terms that are on the same level (equivalence) are represented with = (Giunchiglia et al. 2004).

If x is a hypernym, or parent, of y, it is shown as$x \supseteq y$. In the above example, where alcohol is more general than wine, or wine is a kind of alcohol, the representation is: alcohol $\supseteq$ wine.

Other useful existing knowledge repositories are Wikipedia, Dbpedia [8],

[8]http://dbpedia.org/

PowerSet [9], Freebase [10], ConceptNet [11], Cyc [12], Geonames [13], Yet Another Great Ontology (YAGO) [14].

DBpedia (Auer et al. 2007) is a community effort to extract structured information from Wikipedia and make this information available on the web. It currently describes more than 2.6 million things, including at least 213,000 persons, 328,000 places, 57,000 music albums, 36,000 films, and 20,000 companies.

Geonames is a geographical database containing more than 8 million geographical names that have 6.5 million unique features, located in 2.2 million populated places, and having 1.8 million alternate names. Geographical features are categorized into 9 feature classes and further subcategorized using 645 feature codes.

YAGO (Suchanek et al. 2008a) is a semantic knowledge base that recognizes more than two million entities (persons, organizations, cities). Unlike many other automatically assembled knowledge bases, YAGO has a manually confirmed accuracy of 95 percent.

2.4.4 Ontology Development and Evolution

The traditional ontology development process is normally divided into the following top-down sequence (Noy and Mcguinness 2001) :

[9] http://www.bing.com/
[10] http://www.freebase.com/
[11] http://web.media.mit.edu/~hugo/conceptnet/
[12] http://www.cyc.com/
[13] http://www.geonames.org
[14] http://www.mpi-inf.mpg.de/~suchanek/downloads/yago/

- Determining the domain and scope of the ontology
- Reusing existing ontologies
- Preparing a list of terms
- Defining classes and their hierarchy
- Defining the properties of classes
- Defining the facets of the class members
- Creating instances

Some of the critical tasks in ontology development include (Noy and Mcguinness 2001) :

- Term selection, used to enumerate the important terms in the domain of interest.
- Relationship assignment, in order to determine how pairs of terms relate and then interconnect them into a hierarchy
- Evolution, to select new terms and maintain the hierarchical structure, as the domain knowledge and usage change

2.4.5 Limitations of Current Ontology-Based Approaches

Ideally, the ontology that we build should have significant coverage and depth in the relevant domain and have pertinent resource annotations. Despite the huge

progress made in the automated process of ontology learning from text, ontology building remains a task that depends heavily on human intelligence, knowledge, and experience. The processes of construction, alignment and merging are usually handled manually and often need the involvement of domain experts and ontology engineering professionals (Noy and Musen 2000).

This can lead to several issues:

- Manual annotation of resources requires skilled professionals who are usually expensive.
- It is not easy for a few experts to establish a single and unified ontology that will serve as a semantic backbone for a large number of distributed web resources.
- Concepts may have already become obsolete by the time they are collected and incorporated into the ontology (Braun et al. 2007).
- The ontologies maintained by experts may not fit the needs of online users, since they are usually not able to participate in its evolution and have no control over the resulting ontology. Thus, the efficacy and value of the ontology-based application will be limited.
- An ontology needs to be constantly evolved to adapt to changes in domain knowledge, perspectives of the users, and explicit specifications. But currently, most ontology development is treated like a one-off task..

These issues become more severe in collaborative tagging systems. Traditional ontologies may not be well-matched with CTS because the ontologies do not

	Folksonomy	Ontology
	Freely chosen tags	Controlled Vocabularies
	bottom up by online users	Top down by domain experts
	flat structure/no relation	Hierarchical / Semantic Relation
New Terms	Fast	Slow
Users' needs	High Match	Moderate Match
Search Precision	Low	High
Cost	Low	High

Table 2.1: Folksonomy vs ontology

incorporate the non-standard vocabularies commonly employed by social network users. There is clearly a need to explore alternative approaches that are more economical and scalable.

2.5 Folksonomy vs. Ontology

The following table summarizes the properties of folksonomies and ontologies (Quintarelli 2005; Golder and Huberman 2005).

The first three rows list the major differences between folksonomies and ontologies, namely that:

- Ontologies are typically created by domain experts trying to figure out users' needs and content typologies (Quintarelli 2005)
- Folksonomies are displayed via flat sets of tags
- In contrast, ontologies have explicit formal relationships among their terms, which fall into a hierarchical structure

Because of these differences, folksonomies and ontologies have their own advantages and drawbacks:

- Folksonomies match users' real needs and language better than ontologies because they more accurately reflect the concepts of the population through user-generated content (Quintarelli 2005).

- Folksonomies have lower search precision than ontologies due to an absence of filtering for synonyms and the free-form input of non-standard terms. Because ontologies provide additional content to the terms (domain information and hierarchical structure), ontologies have a higher search precision than folksonomies.

- While it is often too expensive to build and maintain an ontology, the cost of creating a folksonomy is low because it can be developed from the contributions of online users.

2.6 Summary

In this chapter, we reviewed the history of the World Wide Web and discussed its two of the most significant, evolutionary trends, the Social Web (Web 2.0) and Semantic Web. The semantic web can only mature through the development of ontologies that provide a hierarchical organization of knowledge crafted by experts. At the same time, folksonomies are emerging with the development of Social Web, and offering themselves as new, rapidly evolving

approaches to the classification of online resources. Folksonomies and ontologies each have unique advantages and drawbacks. The drawbacks are related to several issues in the development of the semantic web and Social Web, such as low search precision in social media, and the absence of existing ontologies to support the building of semantic web applications.

Chapter 3

Review of the Research Literature

3.1 Introduction

As we discussed in the previous chapter, folksonomies and the related collaborative tagging systems (CTS) as a phenomenon has emerged from the Social Web (Web2.0), while ontologies are used as an enabling technology for the Semantic Web. There is a tendency to view these two developments as opposite and mutually exclusive to each other. A folksonomy lets public community users annotate and classify resources with freely self-chosen tags based on their own terminology and language. Those tags are then aggregated into a bottom-up organization. In contrast, the ontological approach tends to build top down structure with controlled terms that are predefined by domain experts and releated to each other through semantic links. For many areas of interest, there are still very few domain ontologies. But folksonomies are widely used in many social web applications (Van Damme et al. 2007).

With the growth of the social web and the evolution of the semantic web, there is a strong case for applying semantic web technologies to social web data (Gruber 2007). There have been a significant number of efforts to build social semantic web applications by adding semantic structures to collaborative tagging systems.

In this chapter, we review those social semantic web approaches that merge the two ideas together, either by extracting ontological structures from folksonomies, or by enriching folksonomies with existing ontologies. We first review some of the folksonomy studies, and the computational efforts in this area, including data mining, social network analysis, and ontology mapping. We also analyse research related to crowdsourcing and human intelligence methods.

3.2 The Potential Knowledge in Folksonomies

Extracting the ontological structure from a folksonomy can be a meaningful alternative to building it from full-text content or professionally chosen terms. When everyone can assign a set of freely chosen tags to the resources and these tags are obviously based on user's own knowledge or professional background, the resulting folksonomy becomes an abstraction of human thought, a semantic representation of content, and a potential knowledge base. It therefore directly reflects the vocabulary of the users and their choices in diction, terminology, and precision (Mathes 2004). Furthermore, folksonomies are particularly strong in facilitating the acquisition of new terms and are highly customizable without the need for continuous input from experts. In the real world, each person has

individual experiences and views on everything happening in their daily existence. A folksonomy provides a mechanism by which Internet users describe content on the web using their own language and terminology. This allows them to easily classify resources, and thus collect new terms from grassroots. Folksonomies are therefore potential sources of semantic information to support the evolution of an ontology (Bischoff et al. 2008). Torniai et al. (2008) proposed an approach to leverage student folksonomies to support instructors when revising and updating course domain ontologies. This approach allows for a simple and intuitive method for instructors to associate tags with concepts in their domain ontology. It provided a new source of information which can be used to ease the process of authoring and updating domain ontologies and thus promoted the wider adoption of semantic rich e-learning systems

Extracting ontological structures from folksonomies is feasible and meaningful. Even though people assign tags that are based on their own personal knowledge or professional background, the tags form a common basis of understanding that let Internet users communicate with each other (Stuckenschmidt and van Harmelen 2005). By itself, a folksonomy has the potential of being a very weak knowledge base. But when an ontological structure is extracted from a folksonomy, the result is a strong knowledge base that is adaptable in the constantly changing Internet environment. Bischoff et al. (2008) and Golder and Huberman (2006) made in-depth studies of tagging behaviours for different resources and systems, including webpages (Delicious), music (Last.fm), and images (Flickr). It was found that tags are often used to describe the different attributes of resources, such as topic, time and location, type, author/owner,

opinions/qualities, usage context, and self- reference. Since users add a new contextual dimension when doing collaborative tagging, the user-annotated tags tend to be more correlated than those keywords that are automatically extracted by machine, such as Term Extraction Web Service from Yahoo. Another experiment by Al-Khalifa and Davis (2006a) also found that folksonomies carry more semantic value than keywords extracted by machines. This research showed that users had added a new contextual dimension, which almost never happens when keywords are extracted automatically by machine, or an indexer manually assigns a keyword. Over time, a synonymous or hierarchical relationship may emerge, since these related tags are assigned by different users to the same resources.

Bischoff et al. (2008) presented an in-depth study of tagging behavior for very different kinds of resources and systems - Web pages (Del.icio.us), music (Last.fm), and images (Flickr). By analysing and classifing sample tags from these systems, the authors provided an insight of what kinds of tags are used for different resources, and tag distributions in all three tagging environments. The investigation found that web users search in the same manner that they tag. Not only can most of the tags be used for search, but users' tagging behaviors exhibit approximately the same characteristics as their searching behaviours. Thus, grassroots user tags can be used to improve search results.

Although implicit knowledge lies within the common tag collections, the gap between human language (folksonomy) and formal knowledge (ontology) is still significant. Those vocabularies that share a a specific conceptualization remain unexpressed in a folksonomy. For example, rap music/poetry that

has lyrics in street terms ("ghetto speak") has a very specific vocabulary not universally known. Moreover, the formal structures and relationships between the differing vocabularies remain hidden, with the result that practical usage of some tags is difficult in the broader ontology. Therefore, there is an obvious need for research to uncover the shared vocabularies and their associations, and the relationship between folksonomies and more found ontologies.

3.3 Computational Methods for Extraction of Ontological Structures

Ontology extraction is concerned with automatically or semi-automatically discovering knowledge from various forms of data, mostly text. In the semantic web context, it is primarily concerned with knowledge acquisition from and for web content (Buitelaar et al. 2005). Instead of manually preparing a list of terms in manual ontology development, ontology extraction starts by acquisiting of relevant terms from text and then further organizes them into a hierarchy with relationships between the terms. Cimiano (2006) described the typical ontology extraction process as follows:

- Acquisition of the relevant terminology
- Identification of synonym or semantic term variants
- Concepts extraction. Most of the research in concept extraction regards concepts as clusters of related terms. Researchers also discover concepts

from an extensional point of view such as for example all movie actors appearing on the Web.

- Defining a concept hierarchy
- Learning the relationships
- Building a relationship hierarchy

Various techniques and methodologies have been investigated for the extraction purpose. In this section, we review some of the recently studied methods and techniques and related approaches, including, but not limited to: Mining association rules, for finding associated tags and structures (Schmitz et al. 2006); Social networking techniques, for demonstrating relationships to users and studying the social nature of tagging ; Co-occurrence techniques, for finding tag pairs that have similar meanings. In particular, a subsumption model based on co-occurrence is used to find subtopic/supertopic relationships; Machine learning, clustering (Wu et al. 2006a), statistical models (Heymann and Garcia-Molina 2006), and natural language processing (NLP) techniques; Ontology matching and mapping, for finding correspondences between semantically related entities of different ontologies and for reusing existing knowledge repositories (Euzenat and Shvaiko 2007).

3.3.1 Statistical Approaches

Statistical methods are the primary approaches that have been used in earlier research to distill semantically similar or correlated terms from large corpora.

The underlying assumption is that correlated terms are used in similar contexts. In collaborative tagging systems, any number of users may have annotated a resource with tags. Thus it is also assumed that tags occurring together in same resources have similar meanings as well as contexts. In other words, given a tag T, the context of tag T can be defined as a set of tags that have a syntactic relationship to T (such as abbreviations or plural nouns) plus those tags' co-occurrence with tag T in n resources.

Most research has been undertaken using a probabilistic model, which is the mathematical foundation for statistics. Begelman et al. (2006) discussed the use of statistical techniques to identify semantically related tags and thus to enhance the user experience in collaborative tagging service. The algorithm was based on counting the number of co-occurrences of any pair of tags that were represented in a sparse matrix. The value of element is the similarity of the two tags. For example, a user tags an article about African trees with following tags: xhtml, standard, trees, biology, africa, toread, resource. Then (xhtml, standard) and (xhtml, trees) would each get one count as co- tags. After processing the whole tagspace, a cut-off point is determined to identify the significant co-tags. Pairs of tags that cooccur significantly more frequently than cut-off point are considered strongly related.

Wu et al. (2006b) used a probabilistic generative model to analyse the data and automatically derive the emergent semantics of tags, which were embedded within the co-occurrence of resources, tags, and users. The author extended the mixed statistical model for co-occurrence (Hofmann and Puzicha 1998) to a three-part probabilistic model to obtain the emergent semantics. It grouped or

Figure 3.1: All time most popular tags (Fetched on 12-Mar-2011)

clustered relevant tags together in the results, which helped users to find the appropriate tags.

Another popular statistical approach is to use a "tag cloud" or "trending terms". A tag cloud shows the most frequently used tags when annotating the social media. Trending tags are tags that people are using for search at the moment. Figure 3.1 shows the all time most popular tags used on Flickr. The size of the tags in the figure indicates their popularity.

Semantically similar tags can belong to the same class or have other associated relationships such as parent-child or "A is a kind of B". Schmitz (2006) has used a subsumption based model to further induce facet ontology from similar

tags. Heymann and Garcia-Molina (2006) have discovered an effective algorithm for converting large scale tags into a hierarchical taxonomy. They usually arranged terms hierarchically using a subsumption relation, which is calculated from the conditional co-occurrence probabilities of a pair of terms. Term x potentially subsumed term y if: $P(x|y \geq t)$ and $P(y|x \leq t)$ (t is the co-occurrence threshold).

In order to induce an ontology from Flickr, Schmitz (2006) added additional filters to the subsumption model, such as a threshold of the number of authors using a tag, to control for highly idiosyncratic vocabulary. Then candidate term-pairs are selected. From this, a graph of possible parent-child relationships is constructed, using tree-pruning and reinforcement. The experimental results show that the model can generally reflect distinct facets (Schmitz 2006). However, the results also show that the threshold is too simplistic to accurately categorize concepts into facets. The work can be improved by moving to a purely probabilistic model that combines subsumption, tree construction and pruning, and facet categorization. Community moderation via user-based approval or rejection can also help to refine the hierarchy.

3.3.2 Social Network Based Approaches

Another stream of research has employed social network analysis (SNA) for discovering the semantics in tags. A social network is a social structure made up of nodes and ties, where nodes are individuals (or organizations, groups) and ties are one or more specific types of interdependency, such as friendship,

common interest. Social networking is the grouping of individuals into specific groups. Social network analysis is grounded in important social phenomena and theoretical concepts that provide a formal, conceptual means for thing about the social world (Wasserman and Faust 1994). It also exposes the relationships between users and lets us study the social nature of tagging in CTS (Marlow et al. 2006).

By representing a folksonomy as a tripartite network of users, tags, and objects, a semantic relationship between broader and narrower tags has been unveiled through a process of graph transformation (Mika 2007). The folksonomy is defined by a set of annotations $T \subseteq A \times C \times I$ (respectively representing actors, concepts, and instances), which extends the traditional bipartite model of ontology ($C \times I$) through the incorporation of actors into the model. $H(T) = \langle V, E \rangle$ represents a hypergraph of a folksonomy where $V = A \cup C \bigcup I$, $E = \{\{a, c, i\} \mid (a, c, i) T\}$.Through the graph transformation, two associated networks are generated. One is the well-known co-occurrence network of ontology learning; the other is a semantic network based on community relations. This also enables the studying of emergent ontology from user actions in a community. In addition, two other emerging social networks, based on object and concept overlaps, have been suggested (Mika 2007).

The modularity algorithm introduced by Newman (2004)is often used to study this tripartite social network in folksonomy. Edge betweenness is defined as the number of shortest paths between pairs of nodes that run along it. It is the key value in deciding whether an edge should be removed. The notion of modularity is regarded as a measure of the goodness of a particular division of

a network. Removing edges in the network in a progressive manner to reveal the underlying community (Yeung et al. 2007). However some issues, such as observing a particular tag meaning in more than one cluster, remain to be investigated. More analysis is needed to study the effectiveness of the algorithm and how its results can be refined for the task of automatic disambiguation of tags. More promising results might be forthcoming if we used SNA tools such as Pajek and UCINET, or combined clustering algorithms to find the synonym sets of more specific terms(Monaghan and Sullivan 2006).

3.3.3 Association Rules Mining

Association rules mining was introduced by Agrawal et al. (1993). It has been mostly studied in the context of a transaction database, and deals with the "supermarket basket" problem: trying to find a subset of items that are frequently bought together by customers. This analysis helps to improve decision quality when selecting elements for a set. For instance, what to sell, how to promote items, and where to place articles on shelves. The Apriori algorithm (Agrawal and Srikant 1994) is the most famous method of finding association rules. The finding process used in the Apriori algorithm consists of two steps. First, all the frequently purchased items are identified, and then the algorithm generates the association rules.

The Apriori algorithm can be described like this: The analysis is based on a set of transaction (D) and a set of items (I), such that

D = $\{d_1, d_2, ...d_k\}$, I= $\{i_1, i_2, ...i_k\}$,

Each transaction d has some items, where d⊂I. The statement of the association rule is in the form x →y, where x ⊂I, y⊂I, and x∩y=∅. We use the capital letters X and Y to represent the set of transactions that contain x, y separately.

The support value, $s_{x,y}$, is the proportion of transactions in the data set D which contain the itemset x and y, such that

$s_{x,y}=\frac{|X\cap Y|}{|D|}$

The confidence of the rule, $c_{x,y}$, is a factor between 0 and 1, indicating a frequency of transactions in set X satisfying x also satisfying y.

$c_{x,y}=\frac{s_{x,y}}{s_x}$ or,

$c_{x,y}=\frac{|X\cap Y|}{|X|}$

While the confidence factor reflects the strength of the rule, the support value measures the statistical significance and is usually used as a minimum threshold in the analysis. Although association rule mining has been used in many domains as a technique for retrieving significant co-relations between items, little research has been conducted in folksonomy. Schmitz et al. (2006) presented a concept level notation for using association rule mining in a folksonomy and showed how association rule mining could be adopted to analyse it. Since folksonomies provide a three-dimensional dataset (user, tag, and resources), Schmitz proposed reducing the three-dimensional folksonomy to a two-dimensional format and applying association rule mining. When applying association rule mining to a folksonomy data set, association rules like $A \rightarrow B$ are found, which implies that users assigning tag A to some resources often assign tag B to them also (Schmitz et al. 2006). The association rule based approach has been extended by Schwarzkopf et al. (2007) to mine structural

features of taxonomies by pruning the less important relations between tags.

To improve the efficiency of the Apriori algorithm and overcome the problems such as too many association rules with a low support threshold in a large database, a combined use of association rules and classification methods has been proposed by Plasse et al. (2007) to find frequent co-occurrences of attributes in a basket data. In that case, minimum support has to be very low because vehicle attributes are extremely rare contrary to basket data. In addition, the number of rules increases rapidly with a low support threshold configuration. The study showed that the combined use of minimum support threshold and jaccard coefficient was more relevant for low support association rule mining and brought about an important decrease in the number of rules produced. The low support association rules mining can also be applied for different purposes, such as recommending tags, populating the super-tag relations of the folksonomies, and community detection (Stuckenschmidt and van Harmelen 2005) where attributes are also extremely rare contrary to basket data.

3.3.4 Clustering and Similarity Approaches

Clustering is an important technique in data mining for grouping a set of data objects, so that the objects within the cluster have a high similarity to each other when compared with objects in other clusters (Han and Kamber 2006). It helps us to discover data distribution and interesting patterns in the underlying data.

Begelman et al. (2006) employed clustering techniques in CTS to analyse tag similarities by grouping them and showing related tags. In order to find

the strongly related tags, they first counted the number of co-occurence (tags used for the same page) of pairs of tags, and then applied a threshold number to remove unimportant pairs. However, work is still needed to improve the similarity measurements and to overcome tag spamming and inherently ambiguous tags. To further discover the relationships within tags in clusters, several existing upper ontology resources can be used as references, including WordNet.

Approaches such as clustering and displaying related tags do not make the hierarchical relations explicit between tags. As a result, it is difficult for a user to find related resources within the cluster that have broader or narrower tags, which may better represent the user's current interests and help those who have limited knowledge of the subject. Hierarchical clustering is one of the attempts to make the hierarchical relations explicit between tags. Most of these are bottom-up methods: first they compute pair-wise tag similarities, and then the most similar tags are merged into the group. After that, pairs of groups are merged into one, until all tags are in the same group Wu et al. (2006a). On the other hand, top-down methods start from the highest level and move tags into a subclass. For example, an algorithm using graph centrality has been proposed, which aims to convert a large corpus of tags from a folksonomy into a navigable hierarchical taxonomy.

Cosine similarity between tags has been used to measure the distance from one tag to another and to organize them into a hierarchical tree by starting with a single root node representing the top of the tree, and adding other tags in decreasing order of distance Heymann and Garcia-Molina (2006).

3.3.5 Reuse of Existing Knowledge Repositories

An existing ontology can be reused for extending, specializing, or integrating with other ontologies. It can also serve as the base structure of an upper ontology that is used to facilitate relationship construction. An upper ontology is an existing ontology that provides knowledge in common use across multiple domains.

Significant research progress in the field of semantic techniques offers the prospect of extracting semantic structures and relations from folksonomies. To further discover the relationships within tags in clusters, several existing ontology resources can be used as references, including WordNet. One stream of research has taken an existing upper ontology as the base structure and used it to formulate query expansion or facilitate organizing query results (Angeletou et al. 2008b; Pan et al. 2009). WordNet has been successfully applied in many applications as a reliable upper ontology. An et al. (2007) presented an approach to automatically build a domain ontology by interweaving sub-taxonomies of WordNet with information extracted from deep web service pages. In this research, concepts and relationships from WordNet were used to bridge concept gap and tie together ontology fragments into a single ontology.

Ontology mapping and matching techniques are commonly applied to identify relationships between tags; between tags and lexical resources; and between tags and elements in an existing ontology. For example, by mapping "apple and fruit" in a food ontology, we can find the relationship that "apple" is a subclass of "fruit" (Specia and Motta 2007). WordNet has been successfully applied in

many applications as a reliable upper ontology. Laniado et al. (2007) illustrated an approach that integrated WordNet noun hierarchy into the related tags panel of Delicious. By mapping related tags to WordNet and extracting the related terms, the tags and terms were organized into a navigation tree according to semantic criteria.

Vizine-Goetz et al. (2006) presented an approach that involved encoding vocabularies according to Machine-Readable Cataloging (MARC) standards, machine matching of vocabulary terms, and categorizing candidate mappings by likelihood of valid mapping. A web-based terminology service was built based on the extracted vocabularies with associations to other schemes. Patrick et al. (2007) introduced an algorithm that used an augmented lexicon to index concept descriptors in SNOMED CT, which allowed a much faster mapping of the longest concepts in the system as opposed to the naive searching approach. It was able to encode SNOMED CT concepts, qualifiers, negations, abbreviations as well as administration entities.

To improve the quality of the extracted ontology, several researchers have proposed conducting experiments that integrate multiple techniques and resources. Specia and Motta (2007) illustrated a way to make semantics in the tag space explicit by combining shallow preprocessing strategies and statistical techniques with knowledge from existing ontologies. (Kong et al. 2005) suggested an approach that would merge the ontologies through a multi-step process of WordNet mapping, selection of concepts, similarity computation, and reconstruction.

Although the need for relevant ontological structures to support CTS systems is well understood, the upper ontology may not be well matched with tags in the folksonomy (Suchanek et al. 2008b). For example, OpenCyc[1] is widely used as an upper ontology because it describes very general concepts across all domains. However, methods heavily dependent on OpenCyc often get poor results for accuracy due to the fact that terms expanded from OpenCyc may not be frequently employed by users of a specific domain. And on the other hand, many tags gathered from collaborative tagging systems do not exist in OpenCyc (for example, 'folksonomy', 'USYD', and 'UNSW').

3.3.6 Integrated Computational Approach

The above-mentioned multiple techniques and resources can be integrated into a comprehensive approach for extracting ontology from folksonomy. Van Damme et al. (2007) proposed an approach that combines the following:

1. Statistical analysis of folksonomies, associated usage data, and their implicit social networks;
2. Online lexical resources such as dictionaries, Wordnet, and Wikipedia;
3. Ontologies and semantic web resources;
4. Ontology mapping and matching approaches;

[1]http://opencyc.org Opencyc is one of largest and most complete general knowledge bases in the world

5. Functionality that helps human actors to achieve and maintain consensus over the ontology element suggestions that result from the preceding steps.

Specia and Motta (2007) combined shallow pre-processing strategies and statistical techniques with knowledge from an existing ontology to make explicit the semantics behind the tag space. In Ontology-Based Photo Annotation (Schreiber et al. 2001), the Protégé ontology editor is integrated with a WordNet plug-in for ontology construction. FolkAnnotation (Al-Khalifa and Davis 2006b) consists of two processes: a tag extraction/normalization pipeline, and a semantic annotation pipeline. The normalization process is responsible for cleaning and pruning tags. The semantic annotation process is the backbone that generates semantic metadata using pre-defined ontologies.

Gulla and Sugumaran (2008) proposed an interactive ontology learning workbench to consider several of the extraction techniques, such as frequency-based scores, similarity measures, association rules, clustering, etc. It was also implemented as part of a project in which the extracted ontologies were used to search for movie information on the web.

The MOAT project [2] aims to provide a way for users to define tag meanings using URIs of semantic web resources (such as URIs from DBpedia, and geonames). With MOAT, users can annotate content with those URIs instead of entering free-text tags, thus leveraging content into semantic web format by

[2] http://moat-project.org/ontology

linking data together. These integrated approaches demonstrate that varied resources can be combined to help improve the quality of the extracted structures. These integrated approaches demonstrate that varied resources can be combined to help improve the quality of the extracted structures.

3.3.7 Summary of the Computational Approaches

In summary, folksonomies have their own shared vocabularies and relations which can be extracted as an ontological structure and used to improve the exploration and retrieval of digital resources. Although several computational approaches have been proposed to bring structure to folksonomies, they are not without limitations. These include the inability to decide the qualitative nature of the relationship generated by association rule mining, such as which term is more general or more narrow.

Moreover, existing work on extracting ontological structures from folksonomies has been mainly confined to standard tags that are found in a traditional dictionary. Other types of tags that cannot be found in the upper ontologies, such as compound or jargon terms, are mostly disregarded.

3.4 Human Intelligence and Crowdsourcing

Although the most sophisticated computational techniques cannot substitute for the participation of knowledge engineers and domain experts, one of the most influential alternatives to encourage open innovation and solve problems of this

kind is crowdsourcing. It involves outsourcing a job traditionally done by experts to non-experts, typically a large group of people (the "crowd") in the form of an open call (Howe 2006).

Crowdsourcing has been discussed extensively in books, papers, and online under various labels, including collective intelligence, human intelligence, mass collaboration, distributed problem solving, open innovation, crowd wisdom, and user-generated content Doan et al. (2010); Davis (2011). It harnesses the collective knowledge and intelligence of a vast number of individuals to offer solutions, and the winning ideas are often rewarded (Brabham 2008). Crowdsourcing has proved advantageous in a variety of problem-solving activities and often turns out to be more productive than traditional computer-based approaches. Human input is designed into the mechanism for solving problems that are easy for people but still difficult for computers, especially in the areas of image analysis, speech recognition, and natural language processing (Gentry et al. 2005). Central to this approach is using the aspects of knowledge gathering where humans have particular advantages – visual perception, subjective judgment, and aesthetic judgment (Dawkins and Pyle 1991b) – and applying them to information in our ever-changing world. Although there are still some challenges, ideas for building a refinement model using community-moderated support are really attractive.

A typical crowdsourcing system involves the following four activities:

- Software, usually a Web 2.0 application, that has the ability to recruit

large numbers of new users, enable contributions, and collect and interpret input solutions

- A task plan that can break down the problem into small jobs and distribute each unit of the job to independent users in the crowd via the Internet
- Motivation to attract participants, pique human interest, and fulfill their needs so that users are retained on the system
- Analytic mechanisms that can filter the noise from submissions and aggregate the useful responses

A mass collaboration system, mostly based on Web 2.0, is necessary to provide a platform to enable users to continuously contribute their ideas and, content, and to aggregate their knowledge (Brabham 2008; Niepert et al. 2009). Wikipedia, Yahoo Answers, and Amazon Mechanical Turk are among the best examples that successfully engage millions of users' participation.

Crowdsourcing has shown its power in some approaches, especially when there is a need for basic conceptual intelligence or perceptual capabilities, things that most humans take for granted (Dawkins and Pyle 1991a). Examples where crowdsourcing has been applied include Peekaboom (Von Ahn et al. 2006), a game that asks players to locate objects in images; reCAPTCHA (Von Ahn et al. 2008), a methods that requires users to read scanned words before logging into their accounts; and Ontogame (Siorpaes and Hepp 2008), a game for ontology building that asks the user to check the structure and abstraction of random wiki pages. Unlike traditional computation, where a human asks the computer

to solve some problems, in human-based computation, the computer asks humans to do certain tasks, using human intelligence or judgment to do something that a normal evolutionary algorithm cannot do (Dawkins and Pyle 1991a). In other words, human brains are treated as processors in a distributed system to address problems that computers can't yet tackle on their own (Von Ahn 2007).

Unlike traditional computation, where we can simply ask a computer to do a task by means of a software program, it is not easy to ask online users to do as the program requests. They tend to make decisions motivated by self-interest and personal needs or wants. Thus it is vital to attract humans to participate in collective computation by designing certain incentives to meet their needs (Von Ahn 2007). In the following subsections, we classify crowdsourcing systems based on users who have different motivations, and review the related research according to classification.

3.4.1 Community and Volunteers

Web 2.0 provides us not only data but large number of online users and communities. By enabling community members to actively participate in the ontology evolution process, ontology maintenance can be significantly improved, the burden of maintaining it can be shared, and the ontology can be kept up-to-date in the rapidly changing web environment.

Wikipedia is regarded as a crowdsourcing-based service because it incorporates social networking, distributed problem solving, and human computation. It has gained high visibility in the past few years. Although Wikipedia allows

authors full independence and the ability to change every article, we believe that the community corrects the mistakes of its single members. A measurement algorithm is given to justify whether an article has reached a stable state, regardless of whether it has been labeled by readers as being good (Thomas and Sheth 2007). Furthermore, the method is able to predict the current stability and maturity of an article.

Several researchers are planning to build web ontology editors based on wiki techniques. These can overcome the current shortage of ontologies being constructed and facilitate collaborative editing of an ontology (Hepp et al. 2006; Bao and Honavar 2004; Siorpaes 2007).The wiki-based ontology editor has the ability to engineer an ontology using wiki techniques such as versioning, user roles and ranks, mapping to discover similarities, support for community consensus, and ontology editing functionality. It also supports the community of domain experts with automatically generated suggestions, which the experts can discuss and vote on.

OntoWiki is a tool providing support for collaborative knowledge engineering. It does not simply integrate the spirit of existing wiki systems and semantic web knowledge through representative paradigms, but regards the knowledge base as information maps and provides functions for knowledge engineering. OntoWiki has versioning and evolution; it provides an opportunity to track, review, and selectively roll-back changes. It does full-text semantic search, and results can be filtered and sorted using semantic relations. Community support enables discussion and voting about the changes; and there is also intuitive display and editing. Social collaboration is in particular supported in OntoWiki by

features such as change tracking, commenting, rating, popularity, activity, and provenance (Hepp et al. 2006).

3.4.2 Amazon Mechanical Turk and Remunerated Users

Small payments for well-defined, simple micro-tasks are widely used in crowdsourcing applications. Online on-demand labour markets such as Amazon Mechanical Turk (MTurk) provide a cheap and efficient way to get human judgments from registered users. It has opened the door for exploration of crowdsourcing as a means for innovation (Little et al. 2010). MTurk is a web-based service that enables developers to outsource certain tasks, including data collection, information extraction, image tagging, and site filtering, to thousands of human agents all over the world. Any online user can apply and become an MTurk worker (a.k.a. Turker). Each unit of work is referred to as a Human Intelligence Task (HIT). Given the narrow scope of each HIT, the payment per HIT is relatively small. Many of them only pay US$ 0.01 but can be completed in seconds. Compared to volunteers, paid users provide rapid completion of tasks. Most of the tasks are finished in one hour.

A person who creates a task on MTurk is known as a requester. The requester defines the task, including content and time limitation, and sets a price for each HIT. Requesters may also exclude some potential Turkers based on preferred locations or threshold ratings of approved HITs. Not all the participants will receive payment. Only Turkers who complete the task and meet the requester's quality requirements can get paid; otherwise the HIT will be

rejected. Task rejections affect Turkers' approval ratings and may limit their ability to participate in other HITs. However, a requester can also give a bonus to some Turkers who do a good job.

In addition to using a Turker's task approval rating or location selection, there is one more tool that can be used to find suitable candidate Turkers: the qualifications test. A requester can ask those who are interested in the HIT to do this test first. Only the Turkers who pass the test can accept the HIT.

MTurk has been adopted for use in natural language processing tasks. It has 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 images (Little et al. 2010), learning and populating a taxonomy (Eckert et al. 2010), and assessing document relevance (Grady and Lease 2010).

Kittur et al. (2008)studied Mturk users and showed that the service is useful for tasks combining objective and subjective information gathering. However in order to harness the capability of crowdsourcing, special care must be taken when formulating tasks, especially when asking users to make subjective or qualitative judgments. Furthermore, there is no tool available in MTurk for assessing the quality of submissions. Large numbers of HITs submitted by people with different backgrounds always bring noise into the data. It is not easy for a requester to review all the submissions and decide which should be rejected and which should be accepted. To meet these needs, some companies, such as CrowdFlower, provide third-party solutions. They perform as go-betweens to MTurk and provide services to evaluate the submissions. Among them, the

gold standard provided by CrowdFlower is a very useful tool to estimate the accuracy of a Turker's submissions.

3.4.3 Games with a Purpose and the Game Player

The concept of "game with a purpose" was proposed byVon Ahn (2007) to enable humans to solve problems that computers can't yet solve, using games as incentives. These applications are designed on the premise that people around the world spend billions of hours playing computer games, and this energy can be collected. For example, Swash game 3.2is popular and welcome with kids, even those only 2 years old. They are actually helping to wash the balls in the playground, while they are playing and have fun.

In the ESP game, two players are randomly paired and asked to label the same image shown on the screen. If the input words match for the same image, the words will be collected as the tag of the image. It's easy to see that the problems machines have, such as recognizing photos, can easily be solved by humans, if large numbers of users are devoted to the game (Von Ahn et al. 2006; Von Ahn 2007; Siorpaes and Hepp 2008). These kinds of game-with-a-purpose scenarios have also been applied in ontology engineering. The first prototype of this approach, OntoGame (Siorpaes and Hepp 2008), uses Wikipedia articles as conceptual entities and presents them to players who are asked to judge and find their ontological nature and abstractions. Several methods can be employed, such as integration of lexical resources and improvement of usability and user interface. Along the same lines, utyp.net (an image labeling site)

Figure 3.2: Swash game: Kids help to wash balls when they are playing [Photograph dated 2011/03/27 by Winston Lin]

and Yahoo!Answers (a collaborative problem-solving web service) also show promising results. Ontogame has taken purpose gaming into the semantic web realm by proposing a method for ontology building. One of the game scenarios asks users to check the structure and abstraction of random wiki pages.

Google Image Labeler [3] is a feature from Google Search that allows two randomly paired online users to provide several labels for the same image. They eventually get to the point where the two labels match.

Crowdsourcing has also been successful in personal identity search (Wang et al. 2009). Known as a "Chinese-style Internet manhunt" the researchers got thousands of volunteers to collaboratively work together to extract and expose personal information about people and publish those details on the web. Without the presence of thousands of online users working together, it would not be possible to gather such precise information based on very few clues, such as a photo showing only a person's back.

Braun et al. (2007) introduced an ontology maturing process to allow the emergence of ideas from each individual and consolidate them in communities for a common terminology. This is also expected to overcome the problem of time lag between the emergence of topics and their inclusion into an ontology.

3.4.4 Computation as a By-Product of Service Use

Games can be a good incentive for some people, but not for the majority of online users. In addition to monetary rewards and games, the spirit of service and

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

other social-psychological incentives can also be incorporated into crowdsourcing tasks to promote user contributions (Antin and Cheshire 2008). The focus has shifted to offering free services such as email, downloading, or login procedures (Von Ahn et al. 2008), which are familiar to most online users. A general human computational framework that would link Internet problem solvers and problem providers for tasks such as video labeling has been proposed by (Yang et al. 2008). They suggest collecting "common sense" contributions from online users to solve problems like image identification. Several detailed technical challenges are addressed, such as preventing a malicious party from attacking others, removing answers from bots, and distilling human answers to produce high-quality solution responses. Free email or online storage service was suggested as the motivation for Internet users to offer correct answers.

For instance, Facebook 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 attracted thousands of volunteers and completed the French translation task within a few days (Kirkpatrick 2008; Gallaugher 2010).

Von Ahn et al. (2008) ingeniously used human efforts during login verification procedures. CAPTCHA is a widespread security measure used on the web that tries to ensure a human is logging in by asking for input that requires

deciphering scanned words presented as an image. This is a task that computers (auto-login bots) cannot easily perform (yet). The CAPTCHA text transcription test has achieved a word accuracy exceeding 99%, almost at the level of professional human transcriptionists. Although people are more accurate than computers at transcribing scanned text (much better than optical character recognition (OCR) programs by about 20%), they are too expensive and only a few extremely important documents are manually transcribed.

From CAPTCHA Von Ahn developed reCAPTCHA, in order to improve the OCR digitization process of books and printed materials. Using word images supplied by more than 40,000 subscribing websites, reCAPTCHA asks humans to identify the image text at login. But the system needs not only to collect the recognised text but also to verify the user's answer to distinguish human from computer. To do this, two words are presented at login, one of which is known by the computer and the other is not. If the user can successfully type the known one, the computer assumes the other word is also correct and accepts it. To account for human errors, no more than three human guesses are allowed for submitting a correct answer.

Braun et al. (2007) proposed an image-based navigation system that would manage a domain-specific ontology and allow it to mature as a by-product of the daily work of users. In this system, instead of tagging a new image with additional tags, users pulled one image over or under another via drag and drop. The tags that annotated the upper image were then classified as the more general terms. Through this drag-and-drop operation, users' collective knowledge was harnessed to obtain better organization of image libraries while simultaneously

expediting their image labeling work.

Using a purpose-designed system, we can embed the task of building and maintaining ontologies into users' everyday work processes and create the conditions for the ontology to continuously evolve without the help of knowledge engineers (Braun et al. 2007). Limpens et al. (2009) constructed a semantically enriched navigation system using bookmarks. It provides a functionality that enables users to reject or accept broader or narrower tags. These inputs were recorded for further ontology maintenance.

3.5 Semantic Search

With the progress of research in ontology and the semantic web, more and more applications have been developed to utilize ontology for organizing and retrieving information. The typical application is a semantic search engine. Search is the most well-known method to retrieve information. Enabled by semantic web standards and technologies, semantic search offers a more effective search capability than that offered by today's keyword-based search engines (Davies et al. 2009b). Mangold (2007) defined semantic search as an information retrieval process that exploits domain knowledge which can be formalised by means of an ontology.

Matching a query to relevant documents or determining similarity among documents requires the investigation of not only the term, but also the concept that the query represents, which necessitates domain knowledge and reasoning

ability. Parikh and Sundaresan (2008) from eBay Research Labs have experimented and inferred semantic relationships among queries from online search transaction data, specifically product buying activity in e-commerce. Further, the extraction of relationships has been used to improve search relevance and make related query recommendations. A textual similarity method has been used to make connection graphs between similar terms.

In a survey of semantic search engine approaches, Mangold (2007) has found that there are two possible architectures: stand-alone search engines and meta-search engines. The stand-alone search engine crawls through documents, stores their meta-data in an index, and evaluates query requests based on the index. The meta-search engine distributes queries to an index maintained by other search engines and then combines the results afterwards.

An example of a stand-alone semantic search system designed by Hwang et al. (2006) consists of the following four phases. First, it crawls web pages and processes the pages in an HTML parser. It then classifies resources through phasing the ontology, grasping the main concepts, and extracts the domain concepts using WordNet. Following that, a Jaccard similarity formula is applied to get a consistency value. As the last step, the system identifies the representative concepts of what the user wants to find, and shows results that match with the index ontology.

Another framework by Monaghan and Sullivan (2006) illustrates how to make photo annotations for future photo recall by using web services and ontologies. The results of these semantic searches demonstrate a promising future of the integration of ontology and web services.

Noesis is a meta-semantic search engine created by Movva et al. (2007) that helps atmospheric scientists and researchers perform more focused and productive retrieval of the data they need. It simultaneously searches multiple third-party web services like Yahoo and Google for the indexed resources. Generally, search engines lack a semantic understanding of the resources. The semantic search capabilities in Noesis are enabled by the integrated domain ontology and ultimately allow users to refine their search queries using these domain ontologies. Semantic search gives better precision to their results.

For instance, Noesis provides the user with three sets of additional terms that can be used to append or rephrase the search query. These sets could fall into categories such as specializations/generalizations, synonyms, or related terms. Ontology is organized in tree-like taxonomies, where the child nodes and parent nodes represent the specialization and generalization separately and provide a possibility for either a more detailed or a broader search. Including synonyms and related terms also provides better search coverage by appending these terms to the query. Although Noesis is a semantic search engine focused on atmospheric science, it can be configured in other domains, if other domain ontologies are available. At the back end of Noesis is Pellet [4], an OWL DL reasoner, which is pre-loaded with the ontologies and can translate a query into multiple queries covering both narrow and general concepts, and then return search results back to the web service.

[4]http://clarkparsia.com/pellet/

3.6 Summary

After considering the pros and cons of applying ontologies and folksonomies in searching and browsing for resources, we believe that significant benefit can be gained by integrating these two approaches. Using a folksonomy as the resource for extracting conceptual knowledge, we can create an ontology that reflects the terminology of the users and accesses a large number of associated resources. This integrated approach will preserve the strengths of both folksonomy and ontology. Terms that users are familiar with can be linked to structured resources for better searching and browsing.

Although several approaches have been proposed to bring structure to folksonomies, they do not come without limitations. These include the inability to decide the relations generated by association rule mining (such as which term is more general and which is more narrow) and the significance of tags that cannot be found in the upper ontologies. We briefly list some of the limitations below:

1. Machine learning and statistics are commonly used ways to find the relationships between tags. They have limited ability in computing the child-parent relations between instances or concepts. For example, a co-occurrence technique can reflect certain relations between tags in a cluster, but it does not necessarily indicate that there is a parent-child relation between them. Thus, it can hardly categorize concepts into a hierarchical structure.

2. Although candidate concepts and relationships may be generated via learning toolsets, human labour is still needed to verify the suggestions and complete the ontology.

3. Semantic problems in a folksonomy can be partially solved by reusing existing upper-level ontologies to integrate structures. However, methods that rely on existing ontologies frequently are inaccurate, because most of the tags derived from collaborative tagging systems do not exist in WordNet. Newly emerging terminology or non-standard terms have been left out, including widely used words, such as jargon or compound terms

4. The challenge of updating the ontology incrementally has so far not been dealt with properly. Most ontologies that have been built using an automatic or semi-automatic approach do not reflect our fast-changing environment, including knowledge and usage changes. The fact that knowledge changes quickly and users are also increasing makes the need for updated ontologies more pressing. In particular, most of the existing constructions neglect non-standard words, which are used in folksonomies to quickly express users' ideas.

5. There is no efficient tool to evaluate the final ontology or to compare the different extraction techniques. Thus, domain professionals or ontology experts are always needed to check the results.

6. Crowdsourcing has shown its advantage in ontology evolution. But it is hard to find a practical product in this field. Web 2.0 has attracted

hundreds of millions of online users, and most of them spend a lot of time on the web. We need to find a way to use their daily input, and should consider well what are reasonable incentives to attract them. Several attempts have shown that crowdsourcing human computation is a promising method to bring non-experts together to tackle some difficult problems – ontology refinement and evolution – which normally need participation from domain experts.

Chapter 4

Theory, Research Methodology, and Approach

4.1 Introduction

This chapter presents the theoretical foundation for this study, and introduces the research methods employed in this study, i.e, prototyping and experimental research methods. We also discuss our integrated framework and explain the relationship of our approach and the existing framework of human-machine integration theory.

As we discussed in chapter 3, the central problem of extracting valid ontological structures from CTS is that the extracting approach often relies on machine intelligence alone. Our integrated framework, "Ontological Structures Extraction 2.0" (OSE 2.0), innovatively combines the computational power of the machine with semantic search services that gather corresponding knowledge

from online users or the crowd.

This forms a basis for chapters 5 through 7, which explain the details of the framework, its implementation, and the experiments.

4.2 Theoretical Background

4.2.1 Pure Computational Model

A computational model is a mathematical model in computer science that requires extensive computational resources to describe how a system functions. AI might be seen as a useful existence proof for the computational model. Artificial Intelligence (AI) is "the science and engineering of making intelligent machines" (McCarthy and Hayes 1969). With computational intelligence, computers could be trained to think like humans do, to learn from human experiences, and to recognize patterns in large amounts of complex data. Pure computational model has achieved great success in the past half century, becoming a key technology that is used in many areas, including medical diagnosis, agriculture, data mining, semantic web, and machine learning. Today's novel applications of computational model range from semantic searches that understand query intent, to banking systems that detect attempted credit card fraud (Waltz 1997).

Progress in pure computational model has been due to several factors, such as continuous interest and efforts from research and industry, and a greater emphasis on solving specific subproblems. But the primary factor that has accelerated computational development has been the increasing computational power

of computers (McCorduck and Ebrary 2004). More and more, we are relying on computers to solve problems.

For example, as we discussed in chapter 2, information retrieval is the area of study in computer science that deals with searching for things on the Internet. The approach of the Semantic Web uses computational techniques to enable sharing and reusing knowledge on the World Wide Web, while providing data interoperability across applications. It also utilizes machine learning techniques to help develop algorithms that allow computers to evolve behaviours based on empirical data, and to automatically acquire domain-specific knowledge.

4.2.2 Limitations of the Computational Approach

While computational modeling of the computational approach has made much progress in the past four decades and has become a vital part of our life, its capabilities are still limited. There are unresolved problems with classical computational model. A computer can not replace people in many areas, especially when it comes to knowledge gathering, where humans have particular advantages in visual perception, subjective judgment, and aesthetic judgment (Dawkins and Pyle 1991a).

The current data explosion on the web, with all its diversity, has also made it increasingly difficult to provide people with information specific to their needs (Kordon 2010). Issues related to commonsense knowledge and reasoning also must be addressed. Computational model is a technology for learning from human experience. Many applications, including natural language processing,

require vast amounts of information that represent human knowledge of the real world. It is still difficult and time consuming to build a repository that has so much information.

4.2.3 Logic of Integration of Computational and Human Intelligence

Human-computer integration is the theory that computational power is enhanced by outsourcing certain steps to human beings. It is a strategy for solving complex problems based on the idea of effective collaboration between humans and computers. Surprisingly, it often turns out to be more robust and productive than traditional methods (Kosorukoff and Goldberg 2002). Two key elements are emphasized in integration. The first element, which is used in the semantic web, is that machine-based technologies such as machine learning and natural language processing are a foundation for discovering new patterns, relationships, and structures. The second element is that the integration of machine computational power and human intelligence is indispensable (Kordon 2010; Malone et al. 2009). For example, Kosorukoff (2001) proposed a multi-agent approach to analysis and engineering genetic algorithm that combined the intelligence of humans and the computational power of genetic algorithm within one framework. It offered a low-cost and convenient solution that allowed large and distributed groups of individuals to creatively solve the common and individual problems. In chapter 3, we have also discussed other projects such as Peekaboom (Von Ahn et al. 2006), reCAPTCHA (Von Ahn et al. 2008), and

Table 4.1: Division of human and computer in the integrated computation

Selection agent / Innovation agent	Computer	Human
Computer	Genetic Algorithm	Interactive genetic algorithm
Human	Computerized Tests	Human-based genetic algorithm

Ontogame (Siorpaes and Hepp 2008).

An evolutionary model was proposed by Kosorukoff (2001) in his work to describe the division of human and machine labor in the integrated framework (see Table 4.1). It shows a carefully designed mechanism that relies on humans in some role.

In the Web 2.0 era, there are still some challenging research problems that need to be solved before we can realize the full potential of human-computer intelligence integration. (Howe 2006) suggested that this act of mass subcontracting is sufficiently different from traditional small-scale outsourcing to merit a new name: crowdsourcing. The term crowdsourcing puts forth the idea that the World Wide Web can facilitate the aggregation and selection of useful information and knowledge which is contributed by a potentially large number of people (the 'crowd') connected to the Internet. It builds on the principles of Web 2.0 (the participative or read-write web), which enables any interested person to contribute ideas, content, or even services over the Internet. Wikipedia (www.wikipedia.com) and OpenStreetMap (www.openstreetmap.org) are good examples of this form of distributed information gathering and organisation in action.

4.3 Research Methodology

Having presented the integration concept, we continue with a sketch of the research methods employed in this thesis as partial demonstration and validation of the idea. Informed by the literature on human computation and computational model, as well as literature on various aspects of integration theory, we have developed an underlying framework for our research.

The approach we adopt is to develop a prototype integrated application in the area of ontology extraction and evolution. The task is eminently amenable to our research approach. Currently, many ontologies are developed by teams of experts (i.e., human intelligence) but the task of maintaining and evolving them over time has proven to be difficult (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. 2007). There are good reasons to believe that the integrated approach offers greater potential.

Below is a discussion of the research methods.

4.3.1 Prototyping and Experimentation

In the field of design science research, artificial objects or phenomena are studied and designs are made to meet certain goals (Simon 1996). In our studies, we have designed an innovative framework, i.e., "Ontological Structures Extraction 2.0" (OSE 2.0), as an artefact and analysed its use to improve our understanding of how ontological structures are extracted from folksonomies. The artefacts in our design research include - but are not limited to – the creation of algorithms

Table 4.2: Research design of the studies

	Study 1
Perspectives	computational model as intelligence input
Research Objectives	Investigating computational intelligence for ontology extraction
Research Methods	Prototyping and experimental approach
Prototyping	*SmartFolks*
Related Chapter	Chapter 5

Table 4.3: Research design of the studies (continue)

	Study 2
Perspectives	Human as intelligence input
Research Objectives	Investigating human intelligence for ontology evolution
Research Methods	Experimental approach
Related Chapter	Chapter 6

(for example, association rule mining), human-computer interfaces, semantic search system methodologies, the implementation of prototype systems, and a variety of other approaches and techniques.

Table 4.2 , 4.3 and 4.4describe the different focuses of the complementary research studies developed in the approach.

	Study 3
Perspectives	Integrated intelligence input
Research Objectives	Investigating complementary value of the previous approaches
Research Methods	Prototyping and experimental approach
Prototyping	*OntoAssist*
Related Chapter	Chapter 7

Table 4.4: Research design of the studies (continue 2)

In study 1, we use the computational power of the machine to induce a preliminary structure from CTS. We employ data mining techniques, such as association rules mining, to extract knowledge from folksonomies and then combine it with the relevant terms from an existing upper-level ontology. We have implemented ***SmartFolks,*** a web system to illustrate the semantic searching and browsing capability for resources annotated by means of a folksonomy.

In study 2, we then further investigate the practicability of continuously updating the preliminary ontological structure from the inputs provided by online users. Amazon Mechanical Turk (MTurk) was utilized as a crowdsourcing platform in this study.

In study 3, we integrate computational method introduced in study 1 with crowdsourcing method introduced in study 2 and design a new sustainable framework - OSE 2.0. It expands the source of labour from paid workers to a broad range of public Internet users and blends ontology evolution tasks seamlessly with public users' daily search activities.

To demonstrate OSE 2.0 framework we built ***OntoAssist***, a semantic navigation tool. It enhances the native search in CTS, giving users a smart and user-friendly search engine. In particular, the disambiguation feature helps users to search more effectively. At the same time, user input to clarify term meanings is collected to help evolve the underlying ontology. On top of that, OntoAssist can be integrated with third-party commercial search engines and portals such as Google Search, Bing, or Yahoo! Search, using their APIs. As an example, the OntoAssist tool was implemented based on Yahoo! BOSS and released at www.hahia.com. It thus has the ability to provide semantic search and explore

most existing resources in CTS.

To understand the nature of a folksonomy and the effect of a design that integrates human and machine to extract an ontological structure, we must observe phenomena and test algorithm. We have designed several experiments to test our theories, and made a detailed plan for data collection and analysis. Our models were implemented with the Java and PHP programming languages. We then ran these programs and recorded the experimental results, while carefully controlling the model parameters.

The results were analysed and compared to the existing ontology learning models, to see if they supported the following hypotheses:

1. There are hidden sematic relationships among tags in a folksonomy. Data mining techniques can extract these relationships, and find the shared vocabulary and semantics.

2. Human intelligence can be introduced to improve ontology evolvement that is based on the power of the machine.

3. The resulting structure can better support semantics-based searching and browsing of online resources, even with constantly changing usage patterns.

In this thesis, we report the results of experiments that integrate the knowledge from folksonomies and ontologies in a way that achieves a higher level of ontological service quality than could be achieved by each structure alone. Our first experiment, which uses an automated algorithm, has produced promising

initial results using two datasets from Flickr and CiteULike.

Next, we evaluate ontology-based search of CTS with our SmartFolks application, using a 25,000 image dataset from MIR Flickr as our test data. By comparing our findings with the manually annotated benchmark provided by MIR, we show that the technology of semantic web can help users improve the quality of their search and information retrieval experiences.

OntoAssist is currently available online as a demonstration to discover new terms and facilitate rapid ontology evolution. Based on this demo site, we have obtained promising experimental results.

See section 4.4 for an overview of the integrated approach and OntoAssist prototype. The detailed descriptions of each study are reported in chapter 5, chapter 6, and chapter 7, respectively.

4.4 Integrated Approach Overview

In this section, we present an overview of our proposed framework, termed "Ontological Structures Extraction 2.0". Our goal is to develop and test methods to extract ontological structures from folksonomies and facilitate their automatic evolution. By extracting an ontological structure from the tags collected in a folksonomy, we can add explicit semantics to Web 2.0 applications, and use the knowledge of search engine users to help build semantic web structures. Specifically, our framework does an initial automated extraction by exploiting the power of low support association rules mining supplemented by an upper ontology such as WordNet. Also, it integrates the knowledge of search engine

OSE 2.0	SmartFolks	OntoAssist
User Interface	y	y + Related Terms and Relationships Generator
Query Processing	y	y
Knowledge Base	y	y + Wikipedia Repository
Ontology Evolution	n	y
Folksonomy	y	y
Social Media	y	y+ social media provided by search engine giants

Table 4.5: Overview of OSE 2.0 framework and its components and implementations

users to help evolve the extracted ontology with the employment of crowdsourcing.

OSE 2.0 allows users to do more than just passively use the ontology created by experts. By providing a specially designed interface, users can interact and collaborate with each other in CTS and exercise some control over the development process for an ontology, and thus get improved semantic search service based on the evolving ontology.

Table 4.5 shows that OntoAssist is an implementation of OSE 2.0 framework which has six layers. Compared with SmartFolks, an implementation of computational model, OntoAssist has additional ability for ontology evolution and provides Wikipedia as an complementary knowledge base. See Figure 4.1 for an overview of OntoAssist and its main components.

The creative and problem solving power comes from both the machine and humans, whereby the machine generates the related tags and humans assign semantic relationships or propose new domain terms. Although humans can excel at rapid conceptual assertions, such as assigning semantic relationships, they have difficulty finding highly related terms that are located within millions

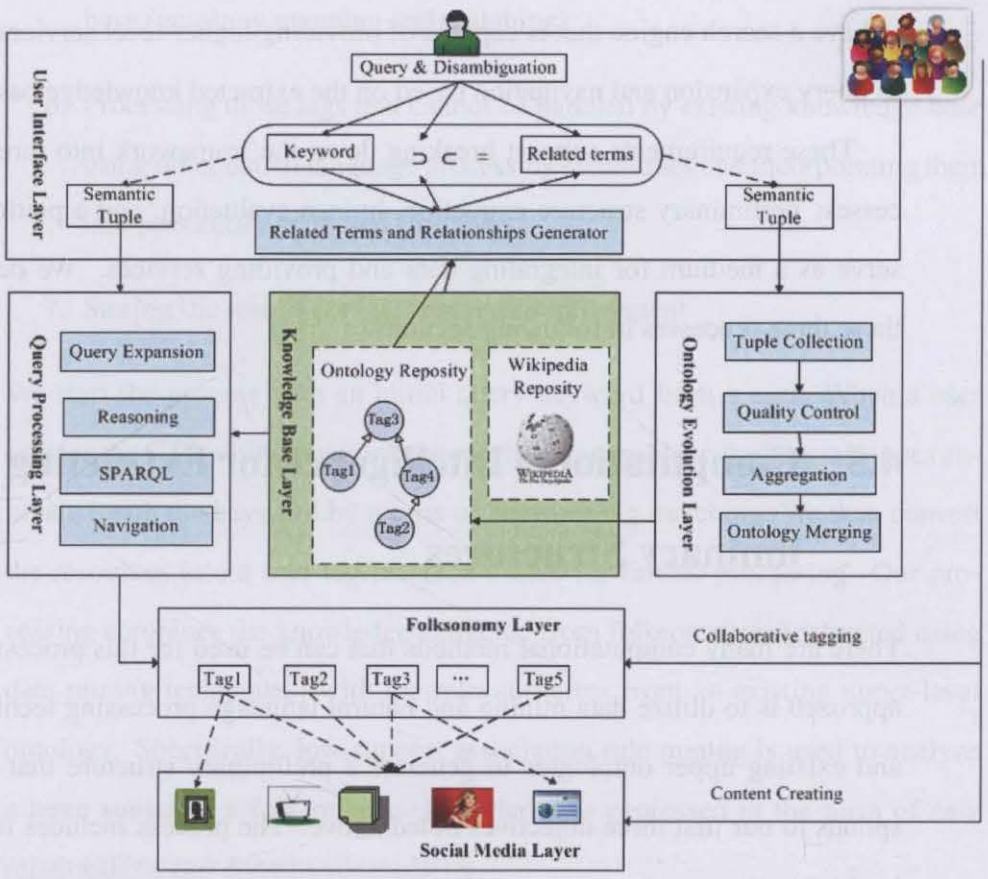


Figure 4.1: Overview of OntoAssist as an implementation of OSE 2.0

of tags. In order for humans to make decisions under these conditions, they need the machine to prepare a preliminary structure of related terms. Furthermore, a well-defined web interface is necessary to present the partial structure to the users and coordinate activities between human and machine. It is also necessary to have a search engine that is capable of providing higher-level services, such as query expansion and navigation based on the extracted knowledge base.

These requirements suggest breaking down the framework into three processes: preliminary structure extraction, human evaluation, and a platform to serve as a medium for integrating data and providing services. We describe these three processes in following sections.

4.5 Computational Intelligence for Extracting Preliminary Structures

There are many computational methods that can be used for this process. Our approach is to utilize data mining and natural language processing techniques and existing upper ontologies to generate a preliminary structure that corresponds to our first three objectives noted above. The process includes the following:

1. Retrieving appropriate resources annotated with seed keywords from CTS (information retrieval)

2. Converting the resources into user-tag-resources models

3. Analyzing the non-directional associations between tags (data mining)

4. Extracting tags that have an association with other tags

5. Organizing tags into ontological structures using the existing knowledge base (ontology mapping and matching)

6. Processing those tags that cannot be handled by existing knowledge base using other natual language processing techniques and incorporating them into previously extracted structures

7. Storing the results for later use and improvement

We start the process with an initial query keyword from a user. When a user queries the CTS with a keyword, the machine is capable to find the resources annotated with the keyword by means of keyword/tag matching. We then convert the resources into a user-tag-resource model for further processing. Our processing combines the knowledge extracted from folksonomies (extracted using data mining techniques) with the relevant terms from an existing upper-level ontology. Specifically, low-support association rule mining is used to analyze a large subset of a folksonomy. Knowledge is expressed in the form of new relationships and domain vocabularies.

We further divide the tag word-formation into three elements: standard tag, compound tag, and jargon tag. Standard tags in the vocabulary are mapped to WordNet in order to obtain semantic relationships. Jargon tags and user-defined compounds are then incorporated into the hierarchy based on domain

knowledge that has been extracted from the folksonomy. Thus, the hidden semantic knowledge embedded in the folksonomy is merged into a formalized ontological structure. At the end of the process, the extracted structure is stored and used as a knowledge base for query expansion or disambiguation purposes.

See chapter 5 for more details about the computer based knowledge extraction process.

4.6 Human Intelligence for Evaluating and Improving the Ontological Structure

Computational processes can accomplish many things beyond tag matching, even candidate concepts and relationships can be generated by computational processes. But, very often, human labor is still needed to verify the suggestions and construct the ontology, as well as to assist with the incremental evolution of the ontology. As stated in objective above, we employ a crowdsourcing model to facilitate distributed problem solving of this kind. We first explore the capability of web users for ontology building, both in the tasks of selecting domain terms and assigning relationships to them. See Figure 4.2 for an overview of the iterative and parallel crowdsourcing processes.

4.6.1 Designing the Task

Each human intelligence task (HIT) is designed to solicit the participants' knowledge of a specific term. An ontology basically describes terms, and the types of

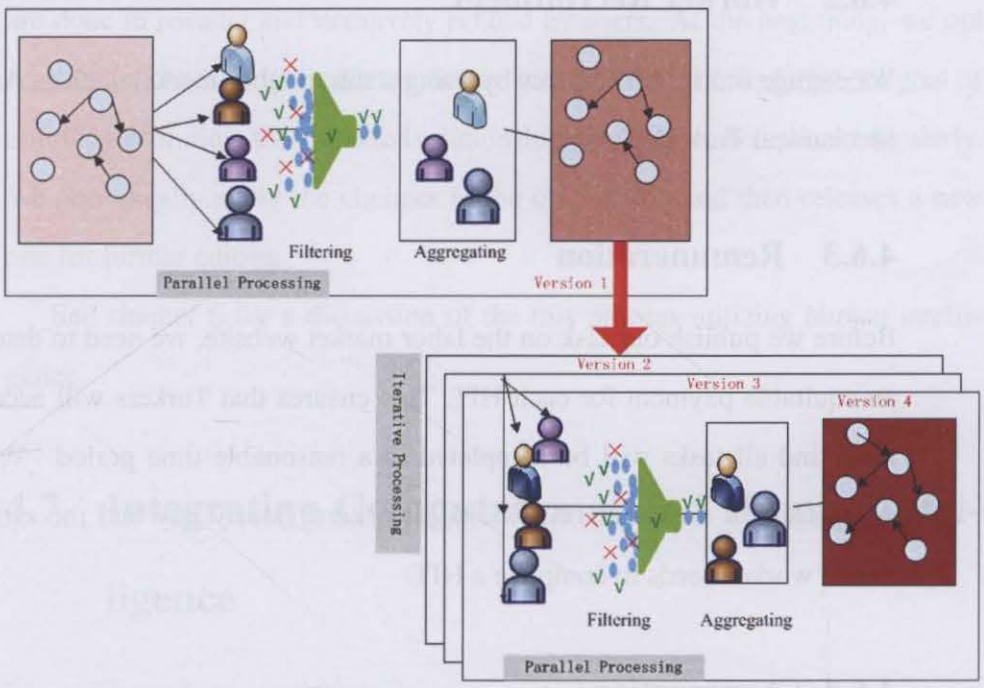


Figure 4.2: Iterative and parallel crowdsourcing processing

relationships between each pair of terms. Thus, an ontology can be expressed as a list of tuples in the form of (term x, relationship r, related term y). For example, "orange, is a kind of, fruit". We designate each of the tuples as a HIT. Thus, we ask the user evaluate each tuple and rate them with value from 1 to 5.

4.6.2 Worker Recruitment

We engage online participants by using a micro-labor market such as Amazon Mechanical Turk (MTurk).

4.6.3 Remuneration

Before we publish our task on the labor market website, we need to determine an equitable payment for each HIT. This ensures that Turkers will accept the tasks and all tasks will be completed in a reasonable time period. We base compensation on factors that include workers' hourly pay and the estimated time a worker needs to complete a HIT.

4.6.4 Aggregation

The decision task aggregates multiple responses. We assume that one expression is correct if there is agreement among the majority of users. Furthermore, we do not treat all user inputs equally. We set up a number of golden standards to distinguish between trusted and untrusted users. Inputs made by trusted users have a more heavily weighted impact on the assessment of the collected responses.

4.6.5 Parallel Processing

Using human computation as part of an iterative (repetitive) process improves quality of responses, while employing parallel processing usually yields a greater variety of responses and yields the best results in brainstorming tasks and domain transcription Little et al. (2010). Our results are obtained from tasks that are done in parallel and iteratively refined by users. At the beginning, we opt for parallel processing in order to ensure response variety. To fulfill our goal of ontology evolution, the extracted relationship data need to be updated regularly. We periodically apply the changes to the old version and then releases a new one for further editing.

See chapter 6 for a discussion of the this process utilizing human intelligence.

4.7 Integrating Computational and Human Intelligence

The success of any crowdsourcing approach depends on providing strong and sustainable motivation to attract a sufficient number of human agents. Monetary award is able to attract a large number and variety of participants. But it is only feasible for short-term projects, such as the early stages of building the ontology. For integration tasks, we present a method that offers sustainable motivation to attract a wide range of Internet users.

We piggyback the integration onto a search engine that is capable of searching multiple CTS on popular social networking sites such as flickr.com, delicious.com, and youtube.com. This immediately gives us a large number of potential online participants. To assure enough user input, the integration module provides simple and intuitive semantic navigation of the query results. This helps the user to locate the desirable terms by filtering out tens of thousands unrelated entries. Moreover, the underlying ontology continues to evolve based on user inputs. Over time, users can see how the services provided by the integrated platform improve. This helps to retain existing users and to attract new ones.

Fig 4.3 shows the data flow of the integration procedure. It also illustrates the users' activities, services, methods and techniques embedded within the lifecycle of the maturing ontology.

4.7.1 Ontology Extraction

Initially the web resources consist of various content types including photo, video, and web pages. These are conceptualized or grouped by community users into collections of tags in CTS. At this point, we begin the work of extracting the candidate ontological structure from these collections. New terms are added and relationships are refined during the semantic search engine phase. Finally, the ontological structure is further refined by community users and converted to OWL format by the system. The extracted ontology can then be used for both semantic searches and other applications.

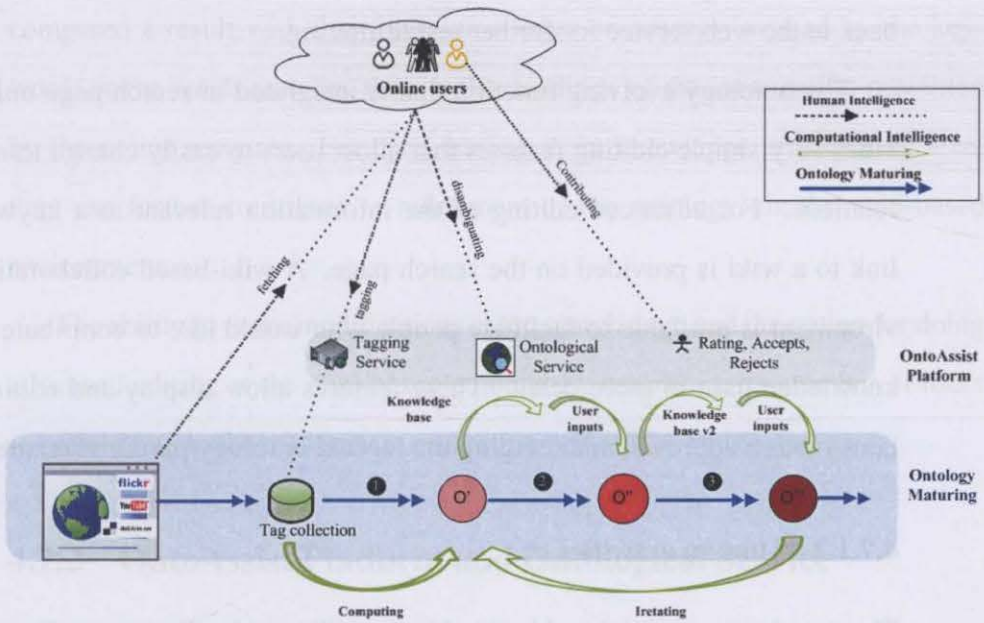


Figure 4.3: Life cycle, processes, activities and view of the methodology

4.7.1.1 Computational activities

The extraction stage exploits the power of low support association rule mining supplemented by an upper ontology such as WordNet.The aggregated individual knowledge of folksonomies is converted into a draft ontological structure.

The search engine is a mashup that uses the Yahoo BOSS (Build your Own Search Service) open search platform, a key term extractor, and an ontology extracted from a previous step. It not only provides a semantic search function using the latest ontology, but it also allows users to refine the search result by modifying the related ontology. Pellet, an OWL DL reasoner, is pre-loaded with the ontologies and translates a query into multiple subqueries about concepts that are either narrower or broader in scope. The translated query then returns

back to the web service for further result filtering.

The ontology evolving function that is integrated in search page only provides very simple eiditing features that allow users to easily change terms and relations. For advanced editing of the information relevant to a keyword, a link to a wiki is provided on the search page. A wiki-based collaborative environment is available to facilitate people who would like to contribute to the knowledge base in more detail. These features allow display and editing, reconstruction approval, and merging the faceted ontology/partial structure.

4.7.1.2 Human activities

The term human as it is used here refers to millions of online users. One principle of our research design is that we are not employing participants to work for us, but rather are providing a web service for them. We collect their intelligence by analyzing users' procedures, logs, or output, and use that information for the extraction process.

Human intelligence is gathered throughout the process of evolving the ontology. Users first browse the web pages or online resources interest them, then tag based on their understanding of the content. The tag collections are human wisdom. After keywords are entered and searched in the semantic search engine, a user can review the generated output, and decide either to modify the terms and relations of the ontology, or directly click on one of the results. The users' action (selection or modification of search results) reflects that they have

compared a result with their own knowledge or concept invoked by the keyword, or the result matches their understandings of the information presented by the online resource. For users having more knowledge in a specific area, they can further contribute to refinement of the ontology via the wiki-based environment.

The ability to make modifications at the search page and the option for doing advanced editing at the wiki-based community enable a large number of users to collaborate in the evolvement and refinement of the ontology.

4.7.2 OntoAssist Platform and Ontological Service

OntoAssist platform is also a medium that provides the following:

- Tagging service: Accesses most of the well-known social media repositories and their tag collections
- Ontological Service: Offers semantic search assist to help users clarify their queries and improve search precision and recall. The user-interface for disambiguation also serves as a means to collect user' knowledge.
- Improved crowdsourcing model: The ontological service acts as reward and incentive to motivate users and encourage their continued participation.

To help users refine their searches, it suggests search terms and semantic relationships. First, the service elicits inputs by listing terms semantically related to the query keywords and offering possible semantic relationships such as 'is-a'

or subsumption after the user conducts a normal search. With this type of help, a user can make the semantic concept more precise by simply selecting a related term provided by the ontology and assigning this relationship between the query keyword and the related term. The semantic search engine will subsequently return better results with a reasoning technology based on the disambiguated query.

For example, by classifying 'apple' as 'is-a' kind of 'computer', the system relates the query results to more specific class such as 'Mac' or an individual model such as 'MacBook Air' while it removes results belonging to 'fruit'. We then collect and aggregate these terms and relationships from different search sessions. Every user-assigned relationship is recorded even if it is in disagreement with the existing knowledge base. The long-term records are eventually split into several clusters to reflect knowledge from different domains. We assume that a user specified semantic relationship is correct, as long as it passes the test of the rule of majority or some other aggregation method that we employ. After that, we introduce a mechanism to periodically merge changes with older versions of the ontology and release an improved version. In short, we demonstrate how to capture the search intent of a user to help with the evolution of the ontology while also improving search results.

4.8 Summary

This chapter began with an introduction to the fundamental theory, i.e. the pure computational model and human-computer integration. It then explained the

research methodologies employed in this research, i.e. the experimental and prototyping methods. After that, we discussed the critical points and models of integration approach for these methods of extraction.

The detailed implementation of prototypes, and a series of experimental evaluation will be introduced in the next three chapters respectively.

Chapter 5

Computational Approach to Extract Ontological Structures

5.1 Introduction

In the previous chapter, we described a framework for extracting ontological structures from folksonomies, based on the integration of computational and crowdsourcing methods. This chapter presents details of the traditional automated computations that discover knowledge in folksonomies and add lightweight structures to the ontology. The two main components of the extraction process that are described here are association rules mining, which is used to represent the knowledge hidden in folksonomies, and an upper ontology (WordNet), which finds relationships among tags.

We propose an architecture for semantic search in CTS. An application called SmartFolks has been implemented to illustrate and explore our ideas.

Promising initial results using two datasets from Flickr and CiteULike are reported.

5.2 Overview

In folksonomies, natural language has been used to annotate and recall resources. Because the human language inputs are not controlled, the vocabularies used in folksonomies fall into the following types:

1. Standard tags, which can be found in traditional dictionaries, e.g., "genomics"

2. Compound tags, which include a non-standard expression, but one of the terms can be found in a dictionary, e.g., "evolutionary-genomics"

3. Jargon tags: popular, non-standard expressions that are used to quickly express users' ideas, e.g., "scientometrics", "folksonomy", "CSCW"

4. Other nonsense or misspelled tags

We propose an integrated approach to address the challenge of extracting ontological structures from folksonomies. Our questions are as follows:

1. How should we extract shared vocabularies from large and dynamic datasets? Users create a large number of new terms every day, but not all of them are useful to others.

2. How can the semantic relations from these shared vocabularies be found?

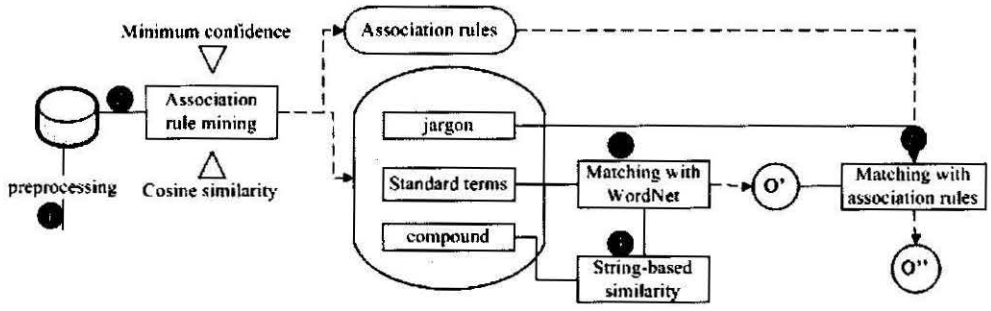


Figure 5.1: The extraction process

3. How should the non-standard tags in folksonomies be handled? For instance, terms like 'folksonomy', 'ESWC', and 'JWS' cannot be found in a traditional dictionary or existing knowledge base.

4. How can the resulting ontological structure help to improve a search for annotated resources?

Note that tags in CTS will be called terms when they become part of an ontology.

In this section, we present our integrated bottom-up and top-down architecture that aims to extract ontological structures from folksonomies, based on the above-mentioned four types of vocabulary. A visual representation of the entire extraction architecture is presented in Fig.5.1.

The system proceeds as follows:

1. **Preprocessing.** In the data preprocessing phase, resources with only one tag or tagged by languages other than English are excluded. However, we should be very careful in this step not to delete jargon and compound

tags. Thus methods like traditional dictionary filtering are not appropriate in this phase.

2. **Association Rules.** Based on association rules mining algorithm (Schmitz et al. 2006; Agrawal et al. 1993; Agrawal and Srikant 1994; Plasse et al. 2007; Liu et al. 2003), we developed a low support association rules mining algorithm to generate association rules representing the relations between correlated tags. In brief, there are three subtasks:

    - Discovering shared vocabularies or essential tags, where a tag should have a certain relationship with other tags. This is the basis for the ontological structure.
    - Extracting the association rules between jargon and standard tags. Association rules are treated as ontology matchers to incorporate jargon into the ontological structure.
    - Retrieving associated terms and excluding non-relevant ones with WordNet.

3. **Standard Tags.** WordNet is implemented as an upper ontology to provide a semantic relationship, which is called a hypernym. After WordNet has connected the semantic relations of standard tags, they are organised into a hierarchical structure.

4. **Compound Tags.** A series of similarity filters are employed to interpret the compound tags before matching them with WordNet.

5. **Jargon Tags.** Jargon tags are incorporated into the previously built ontological structure by matching tags using association rules and a similarity coefficient.

In the following subsections, we discuss each of the steps in detail.

5.3 Mining Association Rules among Tags

The association rules mining (Schmitz et al. 2006; Agrawal et al. 1993; Agrawal and Srikant 1994; Plasse et al. 2007; Liu et al. 2003) is adopted to our datasets to discover possible pair-wise associations between tags. An Apriori association rules mining algorithm has been proposed to solve the "supermarket basket" problem and to discover interesting relationships between items. For example, if 90% of the supermarket transactions that include butter and bread also include milk, the relationship shown as $\{butter, bread\} \rightarrow milk$ with a confidence value of 0.9 (Agrawal and Srikant 1994; Agrawal et al. 1993). Such analysis is based on past transaction data consisting of a set of transactions $D = (d_1, d_2, ...d_k)$ and a set of items, $I = (i_1, i_2, ...i_k)$. In our approach, given a dataset from CTS where every resource is annotated with a set of tags by several online users, the resources set corresponds to D transactions and the tags set corresponds to I items.

The aim of association rules mining in CTS is to generate associations between tags in the form $t_a \rightarrow t_c$ between tags $t_a \bigcap t_c$. The tags have support and

confidence ratings above certain thresholds, called minimum support and minimum confidence. Support of a rule is simply computed as the percent of the resources containing the tag pair. Confidence is computed as the ratio of the number of resources containing both tags $t_a \bigcap t_c$ and the number of resources containing only one tag t_a . While the confidence threshold reflects the strength of the rule, the support threshold measures the coverage.

As a folksonomy is collectively built by various users, the tags in folksonomies usually follow a Zipf distribution.

- a few general tags that occur very frequent
- a medium number of tags with middle-of-the-road scores
- a huge number of tags that rarely occur (the right tail in the diagram)

Traditional association rules mining algorithms normally set relatively high support and confidence thresholds to find common and strong rules. However, this is not the case for folksonomies. Setting a relatively high support threshold is likely to miss important associations among tags in the long tail of the Zipf distribution. Hence we adopt a very low support threshold that includes tags that do not occur very frequently in our analysis. Lower support may inadvertently bring a lot of noise into the rule set. To offset this effect, we introduce cosine similarity (Cattuto et al. 2008; Markines et al. 2009) to filter out possible noise.

However, the single minimum support and confidence in traditional association rule mining has its limitation (Liu et al. 2003) and is not appropriate for

our task:

- If we specify a higher minimum support threshold, only a few popular or general tags will be generated. In other words, the frequently appearing tags generated by this threshold can only reflect the top levels of hierarchy, and do not give us more specific or lower classes. Taking science tags as an example, terms like Ajax or folksonomy are used in only a small number of papers, hence the support Ajax -> web will be very low and will be pruned if we set the support threshold too high. However, the confidence value of the rule can be high.

- To find high confidence but less common tags, we have to reduce the minimum support, which will highly increase the number and complexity of rules, most of which are of little help to our construction of hierarchy. Furthermore, a low support threshold increases the difficulty of finding the proper words to associate with new words.

To apply this measure, we first convert datasets from folksonomies into a metric space V. Given a pair of tags (x, y), tag x is expressed as a vector in this space, where each dimension corresponds to a resource and value indicating whether or not the tag appears in the resource (Salton and McGill 1986). This tag-resource model can be converted into a 0/1 matrix because whether a tag appears in a resource should be 0 (does not appear) or 1 (does appear).

Equation (1) shows a 0/1 matrix for tag(x, y), where each column represents a resource and each row represents a tag, x or y. If a specific tag appears in the

resource, the intersection (row, column) = 1. If not, the value is 0. The traditional cosine similarity between (x, y) can be measured as Eq. (2). Considering the occurrence value is only 1 and 0 in folksonomies, then Eq. (1) can be simplified as Eq. (3), where the capital letters X and Y correspond to the set of resources having tags x or y.

$$\begin{pmatrix} r_1 & r_2 & r_3 & r_4 & \dots & \dots & \dots & \dots & r_n \\ x & 1 & 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ y & 0 & 1 & 1 & 1 & \dots & 0 & 0 & 0 \end{pmatrix} \tag{5.1}$$

$$cos(x,y) = \frac{\sum\limits_{i\in|v|} \times \overrightarrow{x}_i \times \overrightarrow{y}_i}{\sqrt{\sum\limits_{i\in[v]} \overrightarrow{x}_i^2} \times \sqrt{\sum\limits_{i\in[v]} \overrightarrow{y}_i^2}} \tag{5.2}$$

$$cos(x,y) = \frac{|X \cap Y|}{\sqrt{|x| \times |y|}} \tag{5.3}$$

Compared to the support measurement, a cosine similarity measurement not only provides a correlation value between two tags, but it also enables us to prune the rule set because it does not include the resources that contain neither x nor y. Cosine similarity also helps to exclude "high confidence" but poorly correlated rules.

Considering the above-mentioned specialty in our approach, Apriori, the earliest and a highly efficient algorithm to mine association rules, does not fit our purposes well. We modify it and develop a simplified version of the Apriori algorithm, LApriori. Using LApriori, we only calculate the relationships between tag pairs, and both the antecedent and consequent words can only have

one tag. An additional cosine similarity threshold is set to offset the noise caused by low support and to compare the relevance between tags (see Algorithm.LApriori).

Algorithm 1: LApriori to discover association rules in folksonomies

```
Data: resources and tags
Result: association rules
L1 = frequent 1-item sets;
foreach resource r do
    foreach pair of tags {t_a,t_c} in r do
        if t_a ∈ L1 and t_c ∈ L1 then
            increase support of {t_a,t_c} by 1 ;
        end
    end
end
foreach frequent 2-item set {x,y} do
    cos(x,y)= Support(x,y)/sqrt(support(x)×support(y));
    if cos(x,y) ≥ min_sim then
        return x → y
    end
end
```

5.4 Building Basic Structures Using WordNet

We use WordNet as the upper ontology and compute each semantic relation between tags in terms of hypernym relations from WordNet. A term that is more generic or more abstract than a given term is considered to be a hypernym. For example, in Figure 5.2, the term wine has the following upper hypernyms: alcohol, beverage, drink, red, etc.

Algorithm 2: Folk2Onto to find more general term for each essential tag

Data: essential tag
Result: more general term

E1= essential tags ;
foreach *tag* t_k *in E1* **do**
    U_k = the more general term for t_k , set U_k = null;
    S_k = get all tags related to t_k from association rules;
    W_k = get all hypernyms for t_k from WordNet;
    candidate hypernyms set $\{h_1 \ldots h_n \ldots\} = S_k \cap W_k$;
    foreach h_n *in candidate hypernyms* **do**
        if U_k *is null* **then**
            $U_k = h_n$;
        end
        else if U_k *is not null and* h_n *is a hypernym of* U_k **then**
            continue;
        end
        else if U_k *is not null and* U_k *is a hypernym of* h_n **then**
            $U_k = h_n$;
        end
    end
end

Sense 1:	Sense 2:
wine, vino	wine
=> alcohol, alcoholic beverage, intoxicant,	=>red
=> beverage, drink, drinkable, potable	=>color
=> food, nutrient	

Figure 5.2: A sample ontological structure for "wine"

Possible semantic relations between them are described as more general ($\supseteq$) , less general ($\subseteq$) , or equivalence (=) (Giunchiglia et al. 2004). x $\supseteq$ y, if x is a hypernym of y. For example, alcohol is a hypernym of wine, and we can say that alcohol is more general than wine, or wine is-a kind of alcohol, alcohol $\supseteq$ wine.

In folksonomies, we added another two definitions: essential tags and candidate hypernyms. Essential tags are all distinct tags existing in association rules filtered by predefined thresholds. Candidate hypernyms are hypernyms that exist in related tags only. For example, if beverage and food are two hypernyms for wine and also related to wine through association rules, then beverage and food are candidate hypernyms for wine. On the other hand, although alcohol and red are also hypernyms for wine, we do not consider them to be candidate hypernyms because they have no relationship with wine in the generated association rules. We only use hypernyms that exist both in WordNet and association rules, because hypernym terms not related to certain tags in folksonomies do not reflect the subjective knowledge well.

Sense selection is important in the employment of WordNet. Terms in

WordNet usually have several senses of meaning which may relate to different types of domain knowledge. For example, in Fig.5.2, wine has two senses, alcohol and red. Wine is both in the food domain and the color domain. In order to select the correct hypernym in a corresponding domain, we first find all the candidate hypernyms and then determine which one will be selected by matching the hypernyms to the root terms of a given domain. For example, alcohol and red are two candidate hypernyms for the food domain ontology that we are building. We check all the hypernyms from near to far, and then find out that alcohol has one hypernym, food, which is in our domain terms, while the term red does not have hypernym terms among its known domain terms, such as food. Then we pick up alcohol as wine's hypernym.

Based on the above-mentioned considerations, we designed the Folk2Onto algorithm to find more general terms for each essential tag (see Algorithm.Folk2Onto).

For example, given a set of tags food, beverage, wine, milk, the following semantic relations (see Eq(4)) or ontological structures were generated as shown in Fig.5.3.

$$\begin{pmatrix} beverage & \supseteq & wine \\ beverage & \supseteq & milk \\ food & \supseteq & beverage \end{pmatrix} \tag{5.4}$$

Beside hypernyms, WordNet also provides semantic relations such as meronyms, synonyms, and antonyms which can potentially be helpful in our approach.

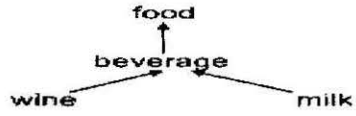


Figure 5.3: A sample ontological structure for "wine"

5.5 Adding Non-standard Terms to the Light-weight Ontology

5.5.1 Compound Tags: Token-based Similarity Matching

Compound tags are non-standard terms and thus cannot be processed by WordNet without transformation. Here we adopt a series of filters provided by Jawbone[1] to analyse the compound tags. If they match certain defined criteria, the compound tags will be reserved and represented by base terms for more general parent finding. In detail, the following term filters are applied to check whether the compound tag has a particular relationship to another term existing in WordNet:

- EndWithFilter operates by splitting the compound tags into independent tokens of standard terms. The last word in the compound is used to represent the whole compound, For example, collaborative-tagging is represented by tagging.

- StartsWithFilter operates in a similar way as EndWithFilter except that the first token is used to represent the whole word. We apply this filter after the EndWithFilter because the first part of a compound is usually a

[1] http://mfwallace.googlepages.com/jawbone.html

definitive term while the last part is usually a subject which reflects the main meaning of the compound tag.

Note that we do not replace or transform the compounds into standard terms, but only use them as interpreters for semantic relation discovery.

5.5.2 Jargon Tags: Combining of Association Rules and Similarity Ranking

In this step, jargon tags are incorporated into the previously built ontological structure. This is done with a matcher using graph centrality in a similarity graph of tags (Heymann and Garcia-Molina 2006). Although jargon tags are also non-standard and cannot be recognized by WordNet, the association rules show their relations with other common tags. Considering each jargon word and its related standard tags as a separate subset in vector space, the tag similarity graph for each subset is a subgraph where each tag is represented by a vertex and the cosine similarity measures the distance between them.

The incorporation process considers each jargon tag as the central node of a subgraph. Then it adds each related standard tag in the subgraph. Based on the matcher between this jargon and its related standard tag, the jargon tag is incorporated into the ontological structure. If there is more than one standard tag associated with the jargon tag, the tag with the highest cosine similarity index will have priority. Association rules involving jargon usually have the jargon as the antecedent. Thus, the jargon tag will be considered a child of its consequence in the rule. This incorporation repeats until all jargon tags have

been connected with their related standard tags in the structure.

For example, a jargon tag, folksonomy, is associated with four standard tags – tagging, plurality, social, and ontology. Ranking by cosine similarity, the rule "folksonomy → tagging"" was selected. Based on this match, folksonomy was incorporated into the ontological structure as a child of tagging.

5.6 Experimental Results

5.6.1 Datasets

The experiments for extracting ontological structures were based on two separate CTS collections taken from CiteULike.org and Flickr.com. The collection from CiteULike was crawled using several keywords; for example, science, philosophy, research. We got 30,769 rows of data, where each row contains a research paper citation with a set of tags from online users. Another dataset from Flickr was assembled using the Flickr API, consisting of a set of methods for users to call up photos, photosets, and other uniquely identifiable objects. We crawled the data using a narrow keyword – fruit – and collected 18,555 rows of data. Preprocessing operations were performed to clean up the datasets. For the Flickr data, we only kept one record for each user because many users batch upload multiple photos with the same tags. These repetitive tags would have caused a biased support count in the association rules mining step. Other cleanup methods were applied to remove the tags labelled notag, a system generated label for an empty tag. We also removed objects with only one tag. Table

Table 5.1: Statistics of collections used in the ontological structures experiment

5.1 shows the descriptive statistics of the collections after preprocessing.

	Collection	
	Citeulike	Flickr
Resources	30,769	18,555
After cleaning	25,937	6,462
Distinct tags	26,709	16,832
Users	4,068	6,462
Seed keywords	science,philosophy,research	fruit

5.6.2 Association Rules

Three parameters were necessary to determine our approach: minimum support (*minsup*), confidence (*minconf*), and cosine similarity (*mincos*). We counted the number of essential tags with different minsup thresholds and observed that most of the essential tags did not occur frequently (see Fig.5.4). Moreover, the investigation of the initial association rule set revealed some interesting patterns of cosine similarity. The value of similarity between pairs of synonyms or subclasses that fell under the same upper class tended to be high, sometimes close to 1. On the other hand, the similarity value between a subclass tag and its parent or upper class tag tended to be low. For instance, food is the parent of beverage in WordNet, and the cosine similarity between food and beverage is low because food is a general term that is associated with many other tags in

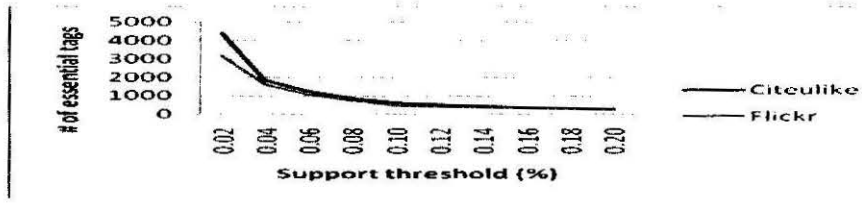


Figure 5.4: Distribution of essential tags

the dataset.

In order to find a proper *mincos* threshold, we tested different values from 0.1 to 1 and evaluated the ontology extracted. The analysis shows that selecting a relatively low value for *mincos* (0.2) tended to preserve more relations between upper and subclass tags and the lateral relations among subclass tags. Thus, we set *mincos* to 0.2. The *minsup* was set to a very low value, 0.02%, to include low-occurrence tags and reflect their relationships (see Figure 5.x above). [fig:tagdist] The *minconf* confidence value was set to 0.8, which is relatively high.

We observed that a total of 152,372 rules were generated from CiteULike at 0.02% minsup. These rules were significantly reduced to 24,025 by using a cosine similarity that was set to 0.2 with 0.8 confidence thresholds. Approximately 4,000 essential tags were found after filtering through minsup, mincos, and minconf. These results also demonstrate the necessity of a very low support threshold. In both these experiments, a support value of 0.02% retains relations between approximately 4,000 essential tags. But if we increase the support threshold to 0.18%, it only keeps relations between 300 essential tags, a low support in traditional associational rules mining.

Table 5.2: Rules with 0.02% support, 80% confidence

Rules	Support	Confidence	Cosine	Yes/No
folksonomy -> tags	1.59%	0.82	0.722	Yes
macroeconomics -> economics	0.09%	0.96	0.2671	Yes
cyber-ethnography -> ethnography	0.06%	1.00	0.2872	Yes
asc->collaboration	0.03%	1.00	0.172	No
final ->social	0.04%	0.90	0.1679	No
seeking -> information	0.03%	0.85	0.1605	No

Table.5.2 shows the effect of the three thresholds. It contains six randomly selected low support rules generated at support threshold 0.02% and confidence threshold 0.8. The low support value helps to preserve rare occurrences of pairs while cosine similarity acts as a guard to exclude rules consisting of tag pairs not highly related. For example, the relationship between macroeconomics and economics was revealed under a low support threshold. On the other hand, although the confidence for the rule final -> social is higher than 0.8, it was excluded because its cosine similarity fell below mincos. If we set minsup higher than 0.18% or mincos higher than 0.3, both the second and third rules will not be revealed or included in the final ontological structure.

5.6.3 Resulting Ontologies

In this section, we present and evaluate the resulting ontological structure. We measure how well the extracted ontology reflects domain knowledge and how much the results can be used to influence and improve the results of certain tasks, including multi-dimensional views, and cataloguing and indexing.

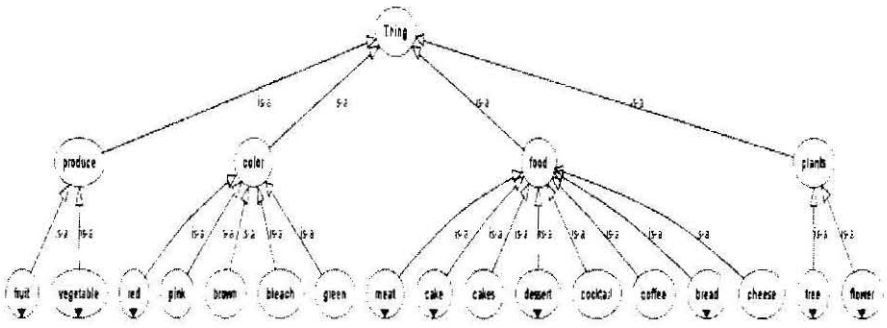


Figure 5.5: A fragment output of "fruit" ontological structure, extracted from the Flickr dataset

5.6.3.1 Result from Flickr dataset

The results from a search with the keyword *fruit* were successfully organised into several dimensions in our approach (see Figure 5.5 and Figure 5.6). In these concept dimensions, the terms that had the most subclasses were: produce, plant, food, and color.

We evaluated the extracted ontology against a "gold ontology" shown in Fig.5.7 [2].

The precision (93%) and recall (63%) were estimated by manually identifying relevant terms from the comparison ontology. Since knowledge in the food domain remains relatively stable, our precision was quite good. However, the recall was not high, because certain terms in the golden ontology are rarely used anymore. For example, although terms like fleshy are missing in our ontology, this term actually rarely appears in tags by online users. We checked photos annotated with the tag fleshy on the Flickr website, and found that only a few

[2]http://www.sei.cmu.edu/isis/guide/technologies/owl-s.htm

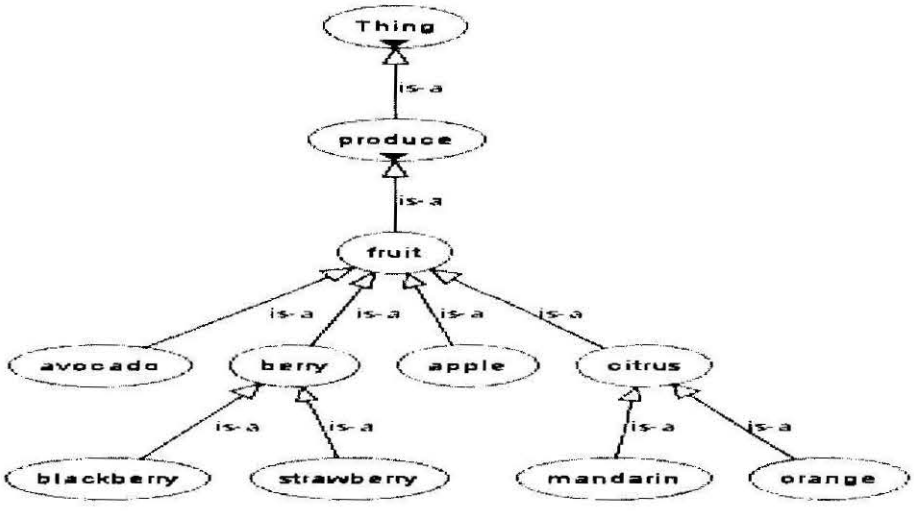


Figure 5.6: Partial subclass output of "fruit" ontological structure

images had this tag. We further evaluated the quality of the extracted ontology and observed that our results have a structure and show relationships between terms that are similar to the golden ontology shown above. See Fig.5.7. The results also show that our method produces more specific terms and additional levels than the golden ontology. For example, in our ontology the term citrus includes the subclasses orange and mandarin. However, our results do not provide enough information about the properties of each fruit, such as flavour, or the fact that strawberries are seedless. The reason is that we currently only consider the hypernym relation from WordNet.

After that, we compared our results with clusters of query results from

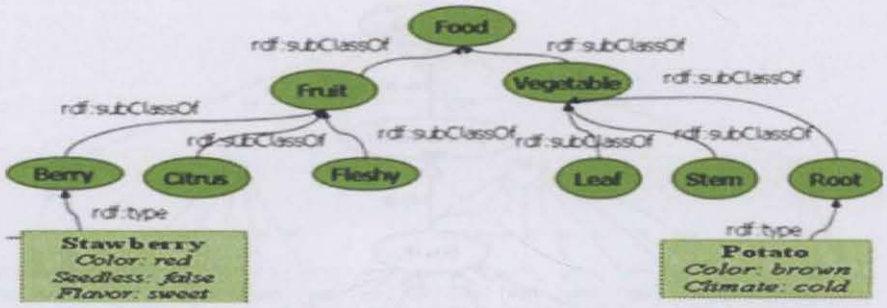


Figure 5.7: An ontology of food

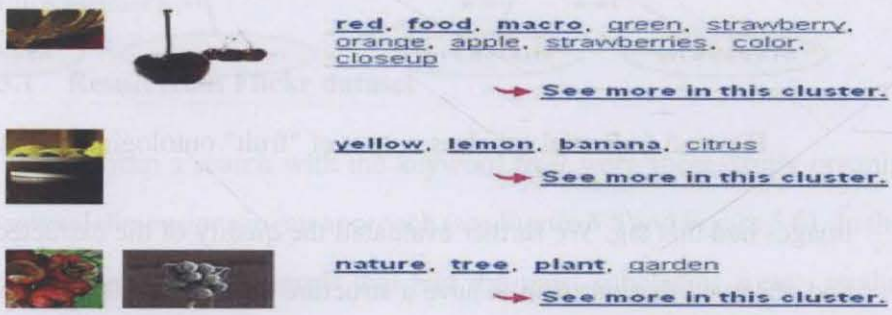


Figure 5.8: Fruit clusters from Flickr

Flickr.com [3] , using the keyword fruit (see Fig.5.8). There are three main clusters in the screenshot. The first is "red, food, etc." The second is "yellow, banana, etc." And the third is "nature, tree, plant, etc." Although we can see that the third cluster has terms that are mainly about nature and should be separate from the other clusters, there is no significant difference between the first and second clusters, since they are mostly the names of fruits. Furthermore, the Flickr tags – like food, or yellow, or red – are not distinguished correctly, and are mixed in the two clusters.

[3] http://www.flickr.com/photos/tags/fruit/clusters/

In contrast, our terms related to fruit are clearly classified into four dimensions, as shown in Fig.5.5. Furthermore, our structure provides detailed subclasses in each dimension. For example, the term berry is placed under the produce dimension, and could be further navigated into blackberry or strawberry, as the figure below shows. In short, the extracted ontological structure reflects the fruit domain knowledge well and organises the related resources into several navigable dimensions.

5.6.3.2 Results from Citeulike

Figure 5.9 illustrates a fragment of the results from CiteULike in the science domain. The related terms are organised into a five-level ontological structure, which gives users an overview of science knowledge. In order to test the possibility of using this structure for cataloguing and indexing the annotated resources, we performed basic indexing based on tag-matching. For example, anthropology and biology are organised under the science class. Then, biology is further divided into genetics and neurobiology. We evaluated the catalogues manually and made sure that the number beside each term showed the numbers of papers contained in the corresponding catalogue.

We also checked that compound and jargon terms, such as evolutionary-genomics, evolutionary-proteomics, and sociobiology, were appropriately incorporated at the correct hierarchical level (as shown in Figure 5.9). In total, 1,540 terms were incorporated into the ontological structure. Among those terms, 35.65% of them were standard terms from WordNet and more than

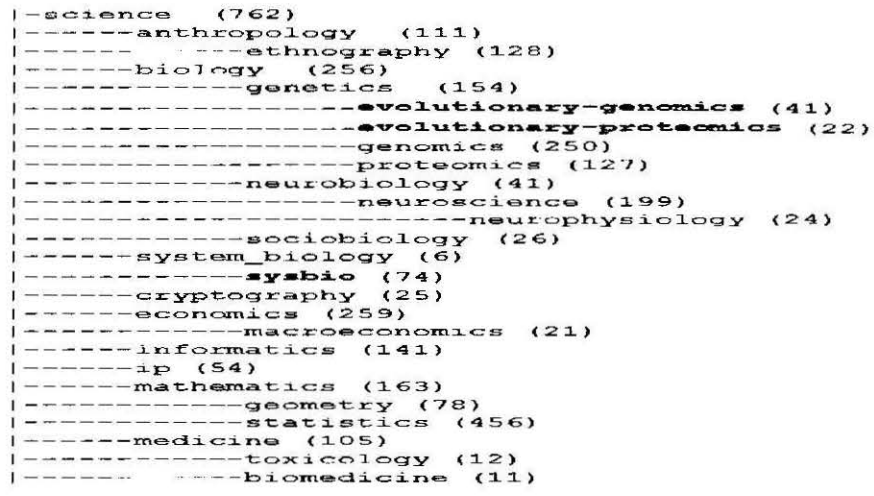


Figure 5.9: A fragment of ontological structure in the science domain

64% were non-standard terms from user tags. Among the non-standard terms, 36.17% were compound words and 28.18% were jargon terms.

5.7 Application I: Semantic Search and Exploration for Images

5.7.1 Architecture for Semantic Search in CTS

In this section, we introduce a general conceptual model of a semantic search system (see Figure5.10) and some potential application scenarios of the resulting ontologically structured folksonomy.

A typical folksonomy based system usually consists of the following three layers that provide resources to users through tag matching.

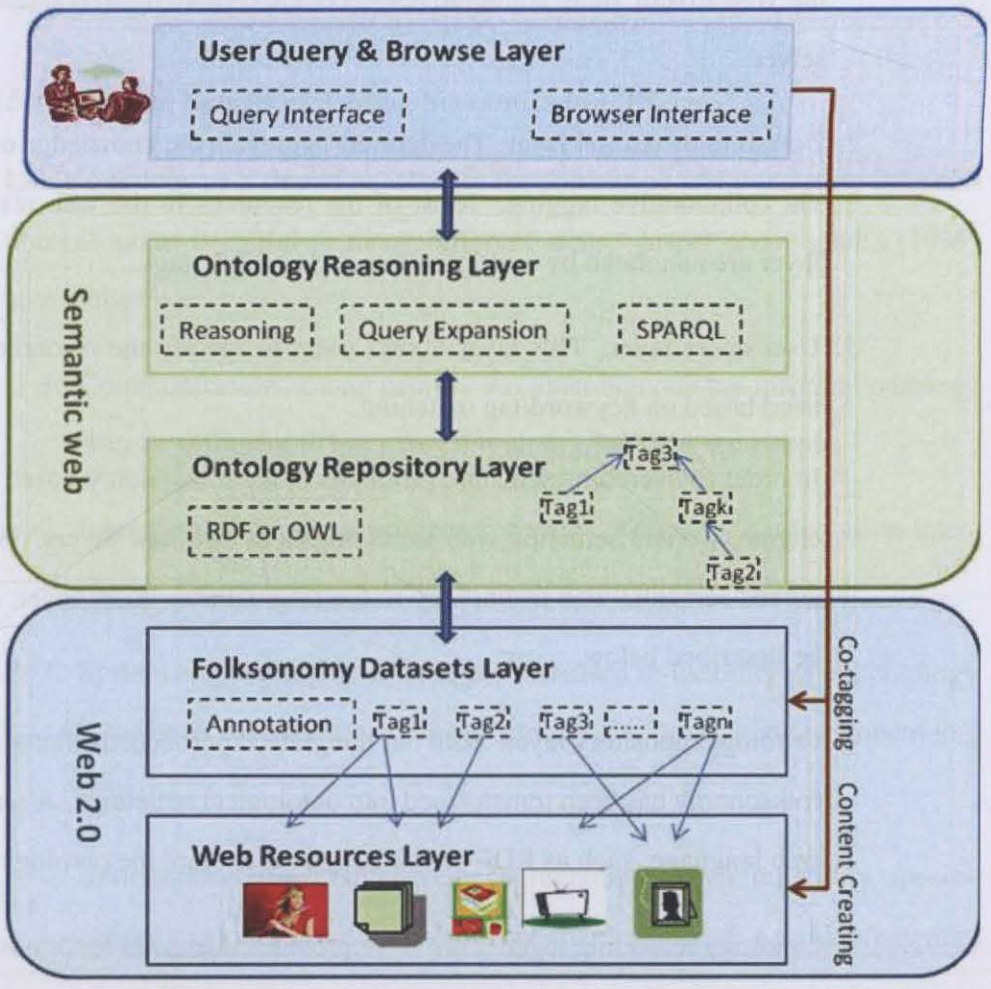


Figure 5.10: Semantic search system model

1. Web resources layer. This layer represents different kinds of online resources such as music files, documents, ebooks, movies, or images. In the Web 2.0 era, more and more resources are created by web users themselves.

2. Folksonomy dataset layer. The datasets aggregate the knowledge of users via collaborative tagging. Most of the resources in the web resources layer are annotated by multiple users and multiple tags.

3. User query layer. This layer allows users to specify the resources they need based on keyword/tag matching.

   In order to overcome semantic problems in the folksonomy-based search engine, we add semantic web technologies in two new layers. We also embed semantic web technology in the original user query layer, as will be described below.

4. Ontology repository layer. With our integrated approach, the flat-structured folksonomy has been transformed into ontological structures. A semantic web language, such as RDF or OWL, is used to store the ontology.

5. Ontology reasoning layer. This layer provides advanced functions based on the ontology. Query expansion and reasoning are processed based on the semantic meaning of the nominated keyword/tag. The SPARQL language is used to query the ontology.

In addition to placing semantic web applications in the repository and reasoning

layers, we embed semantic web technology in the user query layer. Query suggestion is provided based on the stored ontology. We also include hierarchical structured browsing as a complement to the query function.

5.7.1.1 The Benefit of Adding Semantics to CTS

Using ontology as a shared vocabulary for comparing and translating information resources is useful in many different areas. Jasper and Uschold (1999) named four:

1. Communication among people. An unambiguous but informal ontology may be sufficient to improve communication between people.

2. Interoperability among computer systems. Ontology is used as an interchange format between different modeling methods, softwares.

3. System engineering benefits as per increased re-usability where ontology is the basis for a formal coding or process and is a shared component in a system.

4. Information retrieval improvements regarding search, reliability, specification, and knowledge acquisition and maintenance. E.g an ontology may be used as meta-data serving as an index into a repository of information.

Below we discuss several areas where ontological structure can benefit from CTS under the conceptual architecture proposed above:

1. **Multiple dimensions view**: Certain ontologies can be used to organise research results into different dimensions, such as topic, date, or location. In each dimension, relevant resources are organised in a hierarchical structure.

2. **Cataloguing and indexing**: Ontological structure provides an expressive way for accessing and browsing large resources. While a query needs a prespecified keyword for information retrieval, organised catalogues index the keywords and let a user quickly understand an outline and directly browse for further information.

3. **Query expansion and sources integration**: Basing our search on ontology, we can match the query keywords and the potential results at a semantics level, by providing related results on the basis of the shared vocabulary. In CTS, resources are represented by a set of tags and will be returned to a user only if the query keyword matches one of the tags. Using the preprocessed query over concept name, we can maximise precision and recall with respect to the semantics of the resource definition, assuming all relevant information resources have been correctly assigned to ontological classes (Stuckenschmidt and van Harmelen 2005). For example, keywords in a query can be replaced by their approximations in the ontological structure, and related instances will be returned.

4. **Tagging suggestion with dynamic ontologies**: Tagging suggestion is useful because it helps you create a subset of tags. Suggesting relevant

ontological classes to the user will not only improve the tagging experience, but increase classification quality. Moreover, a dynamic ontology can be used to address the awkward and inflexible annotation interfaces that are based on closed, hierarchical vocabularies (Schmitz 2006). Ontology can thus be used to integrate heterogeneous databases, enabling interoperability among disparate systems and specifying interfaces to independent, knowledge-based services (Gruber 2007). For instance, by representing web resources with a conceptual meaning and placing them in hierarchical structures, the machine application, such as a search engine, can find the primary resource and related resources by semantic understanding of them.

5.7.2 SmartFolks, the Implemented Application

We have implemented a web system to illustrate the semantic searching and browsing capability for resources annotated by means of a folksonomy. Our prototype system was based on a five-layer conceptual model of semantic search, and our test dataset came from the MIR Flickr photos assembled by (Huiskes and Lew 2008). This image collection consists of 25,000 images downloaded from the Flickr website via its public API. It represents a real community both in tags and content. Furthermore, image metadata is also provided in the collection, which consists of information such as the type of camera, date-time, and exposure settings used when taking the corresponding picture. Jena (http://jena.sourceforge.net/) is an open-source Java framework that provides

a programmatic environment for building semantic web applications. It also includes a rules-based inference engine. A Jena framework was used in the ontology repository layer and reasoning layer to provide ontology storage and other operations such as reasoning or query expansion.

Since location and time are two key dimensions that are used to annotate the images, we applied the methodology described in section 5.4 to the dataset and got two additional ontologies about location and time. After that, we integrated these ontologies, converted the results into RDF format, and employed it as a backend knowledge base in this system. We used a Java server page and the J2SDK development kit for this demonstration application. The plain text content of the MIR Flickr datasets were transformed and imported into MySQL. The SmartFolks demo site is available at http://smartFolks.thetag.org. Figure 5.11 is a screenshot of the SmartFolks website.

The system highlights three areas where ontological structure can benefit the CTS. These are described below.

5.7.2.1 Categorising Through Navigational Browsing

Web browsing is an important aspect of information-seeking behaviour that complements searching (Davies et al. 2009b). Ontological structures provide an expressive way to catalogue and index large numbers of digital resources. While a query needs a prespecified keyword list for information retrieval, the ontological structures give users a quick understanding of the subjective knowledge, allowing them to directly browse for further information.

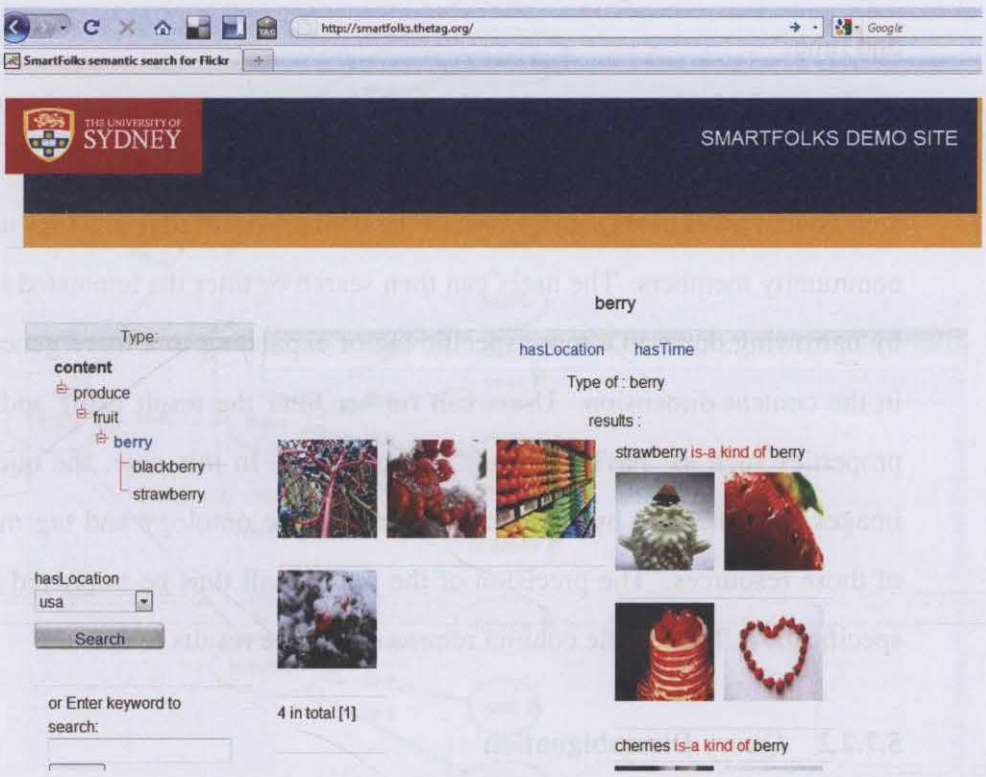


Figure 5.11: Snapshot of the smartFolks web page

In order to provide ontology-based image browsing and navigation capability, we have integrated the jOWL[4] browser plugin into our web system. This is a jQuery-based Javascript visualisation tool for navigating and viewing an ontology in OWL or RDF format. In Figure 5.11, the left hand column displays tags that have been organised into three selected dimensions: content, location, and time.

Figure 5.12 illustrates a visualisation of an ontology fragment for content dimension. The visualisation of the whole ontology in the left column of the web system gives users a quick idea of the domain knowledge and tags used by community members. The users can then search or filter the annotated images by narrowing down to a more specific tag or expanding to a more general tag in the content dimension. Users can further filter the result using additional properties such as 'hasTime' and 'hasLocation'. In this case, the query for images is formulated by selecting the class in the ontology and tag matches of those resources. The precision of the search will thus be improved by the specification. The middle column returns the image results to users.

5.7.2.2 Query Disambiguation

We can further match the query keyword to a specific word sense by providing users with domain contexts derived from the ontology structure, and asking them to select the most appropriate one. The ambiguities of a user's query will thus be reduced by applying more contexts to the keyword-tag association

[4]http://jowl.ontologyonline.org/ jOWL is a jQuery based javascript plugin for navigating and visualising ontology in OWL or RDF format.

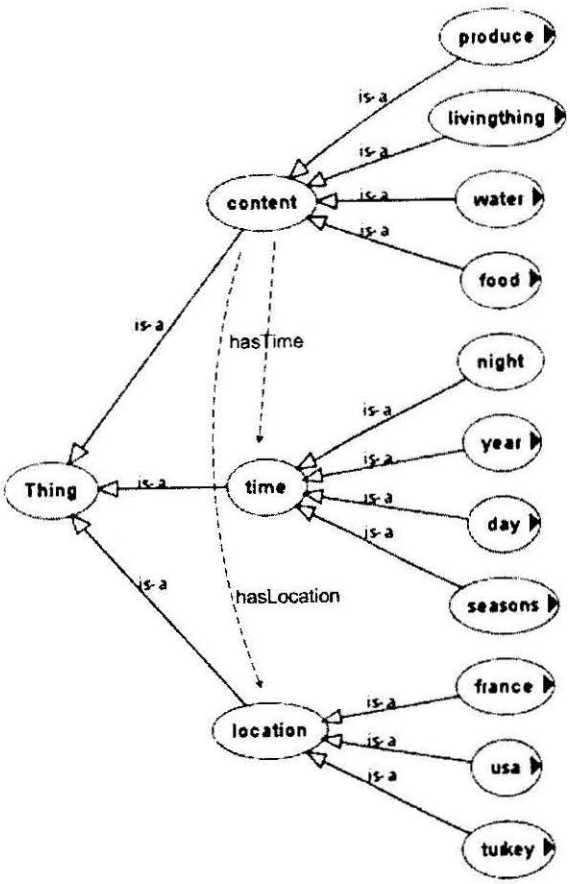


Figure 5.12: Visualization of ontology fragment showing three main dimension

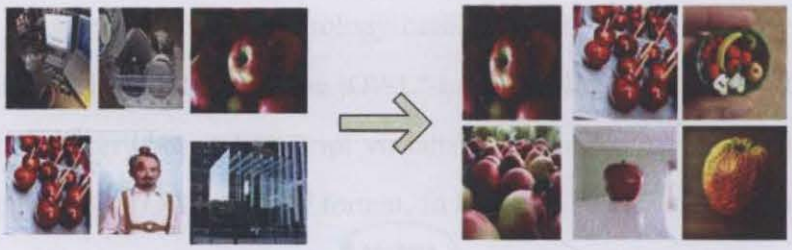


Figure 5.13: Query disambiguation example

(Pan et al. 2009; Specia and Motta 2007). If a user is looking for resources with a polysemous tag such as apple, we will present all related word senses for further selection. One is upper-case 'apple' with 'company' and another is 'fruit'. Figure 5.13 illustrates some image examples of apple. After specifying the correct sense – in this case, fruit, the system filters the result by adding the related fruit tag to the apple query. In this way, the images showing products of Apple Company are removed. As a result, the precision of the apple query in the MIR Flickr image dataset markedly increases from 13% to 100%.

5.7.2.3 Query Expansion

Query expansion is a common method of improving recall in information retrieval. In ontology-based expansion, one term serves as an input that expands the query to a set of terms (broader or narrower) based on hierarchical structures within the domain. In CTS, a set of tags is attached to a collection of resources, and search is typically done via the matching of keywords to tags. Using the preprocessed query over a concept name, we can enhance precision and recall

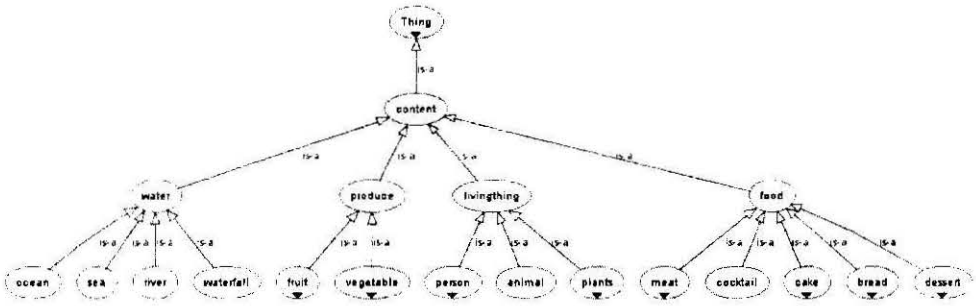


Figure 5.14: A visualization of an ontology fragment showing content dimension

the semantics of the term if all relevant information resources have been correctly assigned to the pertinent ontological classes (Stuckenschmidt and van Harmelen 2005).

Our method separates the search result into two parts. The first part displays the search results from the initial computation. The second part shows the result of query expansion, which is presented on the right side of the webpage. After the user specifies a keyword or selects a term in the navigation, the system will also present images annotated with related class tags. The expanded query displays the top k nearest subclasses from the ontology. Next, we collect the top n images annotated with each of the tags. Thus, k*n additional images are generated from the whole collection. In our demonstration, we set k=3, and n=10. To evaluate the proposed query expansion, we took 20 tags as our test queries, using precision and recall as our measures of success. All relevance assessments of the selected 20 queries have been provided by annotations in (Huiskes and Lew 2008). Figure 5.15 illustrates that when a user is looking for water, the system will not only return images that have been tagged as water,

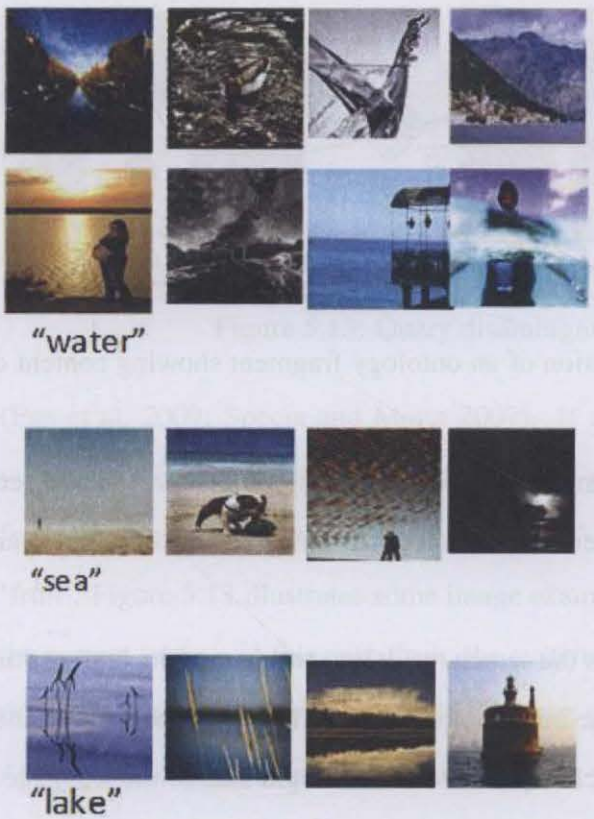


Figure 5.15: Water and its relevant picture

but also retrieve pictures annotated with the top subclasses based on ontology described in Figure 5.14, such as sea and lake. We compared our method with a simple tag-based search.

The results in Table 5.3 show that average recalls are greatly improved while maintaining almost the same level of precision.

Table 5.3: Analysis of results for query expansion

	precision	recall
normal tag-based search	92.8%	6.51%
with expansion	90.8%	18.81%

5.8 Summary

In this chapter, we have proposed an integrated computational approach to extracting ontological structures from collaborative tagging systems. By analysing four kinds of word formations found in folksonomies (standard tags, jargon tags, compound tags, and nonsense tags), our approach has produced promising initial results using datasets taken from Flickr and CiteULike.

Though WordNet as an upper ontology resource contains a wide range of common words, it does not cover special domain vocabulary and cannot reflect recent changes in usage. In CTS, many of the tags are in the form of jargon and compound terms. Mapping terms with the WordNet ontology is obviously not enough to find the relationships among tags that contain non-standard terms. Thus, additional consideration was given to incorporating these informal or special terms found in tags into ontological structures. To do this, we matched tags by using association rule mining and token-based similarity. Unlike data clustering techniques, association rule mining is an unsupervised method that finds interesting associations between datasets. We applied the association rules to find semantically related tags that became the basis for further ontology building. Furthermore, we simplified the a priori algorithm to find two-item set rules

and introduced a new cosine coefficient, which significantly improved the efficiency in low support mining.

We also implemented a semantic search prototype based on the resulting structure. This shows that the technology of semantic web can help users to improve their experience with information search and retrieval.

Chapter 6

Ontology Development and Evolution Using Crowdsourcing

This chapter describes our method to construct an ontology by integrating judgments from a large number of human evaluators working through crowdsourcing platform such as Amazon Mechanical Turk. As stated in chapters 2 and 3, ontologies are generally developed by small groups of experts, but this may not be the best approach. Costs can be prohibitive and assembling suitable experts can be time-consuming. Besides, even experts have difficulty keeping up with the advances in knowledge in the open, dynamic World Wide Web environment. Through crowdsourcing we can aggregate knowledge from the crowd to help determine relevant terms and their relationships for the ontology.

6.1 The Basic Workflow and Terminology of MTurk

Amazon Mechanical Turk (MTurk) is a micro-task marketplace for work that requires human knowledge. It gives us access to an on-demand, scalable workforce with the flexibility to increase or decrease the number of workers quickly, and pay only when satisfied with the results (Sorokin and Forsyth 2008).

In this section, we adapt the basic framework in MTurk and the specific terminologies used there [1].

Amazon Mechanical Turk, referred to as MTurk. MTurk is a public, online task service. Requesters post jobs that are short-term, freelance tasks that can be done over the Internet. Workers accept and complete these Human Intelligence Tasks, called HITs. Upon the requester's approval, the worker is paid via Amazon's payment service. These HITs pay small amounts – most pay less than one U.S. dollar. Requesters must have an address and bank account in the United States; however, people living in other countries can still access MTurk and become requesters via third-party proxies. Workers can be from countries outside the USA, but there are payment restrictions. For this reason, the majority of MTurk workers are from the United States and India.

Requester. The entity creating and posting a job on MTurk is the requester. As noted above, requesters must have an address and bank account in the United States. The requester puts the job, called a HIT, in the Mechanical Turk format, which involves writing a description of the task and supplying any URL links to the task elements. The requester chooses a maximum time allowed to complete

[1] Adapted from MTurk online documents https://requester.mturk.com/help/faq

the HIT, the payment amount (called a reward), and, optionally, a minimum approval rating or successful completion of a test as qualification to work on the HIT. MTurk collects a commission that is 10% of the payment amount per HIT.

Worker. also known as Turker. Those who accept and work on MTurk HITs are called workers or Turkers. Because of Amazon's payment restrictions to foreign countries, most Turkers are from the United States and India. Turkers can select from thousands of HITs, most paying less than one U.S. dollar, with completion time limits usually less than a few hours. Workers accumulate an approval rating for the HITs they complete and submit. This is the percentage of the Turker's approved HITs. Requesters can specify a minimum approval qualification for Turkers to work on their HITs.

Human Intelligence Task, or **HIT**. This is an individual micro-task that is done by Turkers. A HIT can be as simple as labelling an image with one keyword. HITS pay small amounts of money, ranging from one cent to several dollars. Most HITs pay less than one U.S. dollar.

Maximum HITs per worker. This is the maximum number of HITs from one requester that an individual Turker is allowed to submit. For example, one requester may create several HITs of the same type, such as labelling images. Perhaps there are five HITs, each with the same description but having different images to label. The requester can limit the number of times the Turker can label the images (and thus the number of images labelled by a single Turker) by limiting the number of HITs that particular worker can accept.

Minimum Workers per HIT. This is the minimum number of individual

workers required to work on a particular HIT. For example, if a requester's HIT is a fixed set of images to be labelled by several different workers, the minimum number of Turkers would be specified with this flag. More accurate results usually require more workers per HIT.

Judgment. The HIT content or response submitted by a Turker.

HITs per assignment. This is the number of HITs from the same requester that a Turker is allowed to work on concurrently.

Figure 6.1 shows the ease of work distribution using MTurk.

The basic workflow is as follow:

1. Planning stage. A requester defines the goal of the job and breaks it down into practical steps. Suppose you want to organise one thousand images with labels, and you want each image to have up to three keywords associated with it.

2. Task design stage. A requester designs the task interface where questions and instructions are given. It is important to make the task instructions clear and concise in order to get accurate answers from workers. The reward (payment amount) per HIT and maximum workers per HIT are also specified at this stage.

3. Publication on the market. The tasks will be publicly listed on the MTurk website (https://www.mturk.com) after the requester releases the HIT. Turkers can find it by searching or browsing the available HITs.

4. Processing. Workers may select and view the details of the available HITs

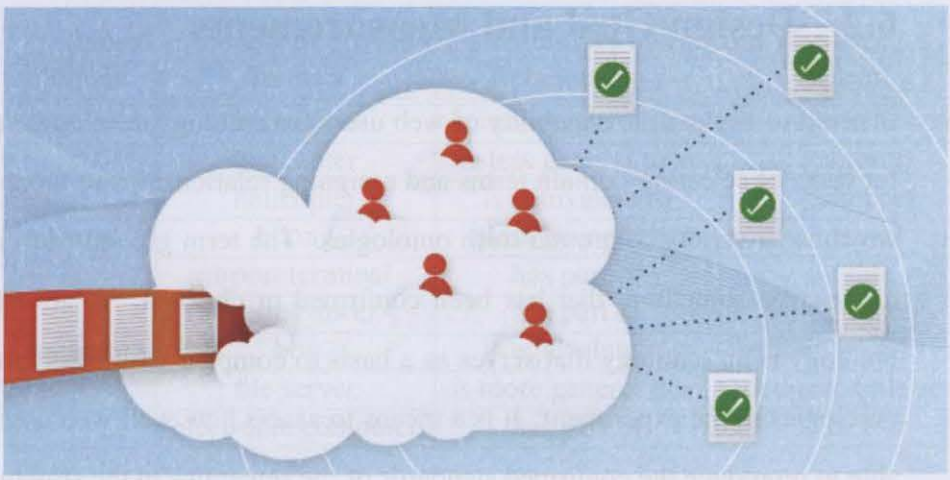


Figure 6.1: Work distribution made easy with Amazon Mechanical Turk

and accept one if it is of interest to them. A number of factors may affect worker interest, including the amount of reward offered, the clarity of the instructions, and if the HIT matches their experience and knowledge. After a Turker accepts a HIT, it must be submitted before its duration time limit expires.

5. Review and payment. A requester reviews the submitted judgments and approves or rejects them. Only approved HITs are paid. The requester can pay an optional bonus for excellent submissions. MTurk charges the requester 10% of the HIT reward as a commission fee.

6.2 Design Goal and Measurements

In order to explore the capability of web users for ontology development, both for tasks of selecting domain terms and assigning relationships to those terms, we chose a variety of ground-truth ontologies. The term ground-truth is used to describe something that has been confirmed or checked. A ground-truth ontology is an ontology that serves as a basis to compare with other extracted ontologies in the experiment. It is a means to assess how well web users were able to reproduce the confirmed standards of the semantics in the structure.

Essentially, as we discussed in chapter 2, an ontology describes terms and the types of relationships between pairs of terms. Thus, an ontology can be expressed as a list of tuples in the form of term x, relationship r, and related term y. For example: orange, is a kind of, fruit. We designated each of the tuples in our dataset as a HIT. Thus, the experimental setup was: Given a term x, can the user correctly find the related term y from several optional terms, and select a proper relationship r from a list of possible relationships with the term x?

6.2.1 Source Data

We drew on WordNet as the source of the ground-truth ontologies. WordNet is a widely accepted upper ontology which describes very general concepts across all domains. We extracted terms relating to three domains: vehicles, computers, and travel by using these terms as keywords and querying on WordNet 2.1. To make the relationships easy to understand, we changed the original relationship

Table 6.1: Sample of ground-truth ontologies taken from WordNet 2.1

domain	Terms x	Relationship r	Term y
Vehicle	helicopter	is more general than	cargo helicopter
	helicopter	is less general than	aircraft
	helicopter	is equivalent to	chopper
Travel	hiking	is less general than	travel
	airport terminal	has part of	gate
	control tower	is a part of	airport
computer	bbs	is equivalent to	bulletin board system
	file server	is more general than	dedicated file server
	electronic computer	has part of	busbar

("is a kind of...") to the comparisons "is less general than", and "is more general than". We selected 180 tuples as datasets, 60 for each domain. Table 6.1 shows a selection of these tuples.

6.2.2 HIT Description

Each HIT was designed to solicit a human agent's knowledge of a specific term. The procedure consisted of the following three steps:

First, a term x was selected from the dataset and presented to Turkers.

Second, two terms – a relevant term y from the same tuple, together with another term from a different domain – were presented as possibly related to x. Turkers were requested to review these two terms and select as term y the one most closely associated with the term x from step one.

In the last step, a Turker specified a type of relationship between the given term x and the selected term y. The types of relationships consisted of "is more general than", "is less general than", "is equivalent to","is a part of " and "has

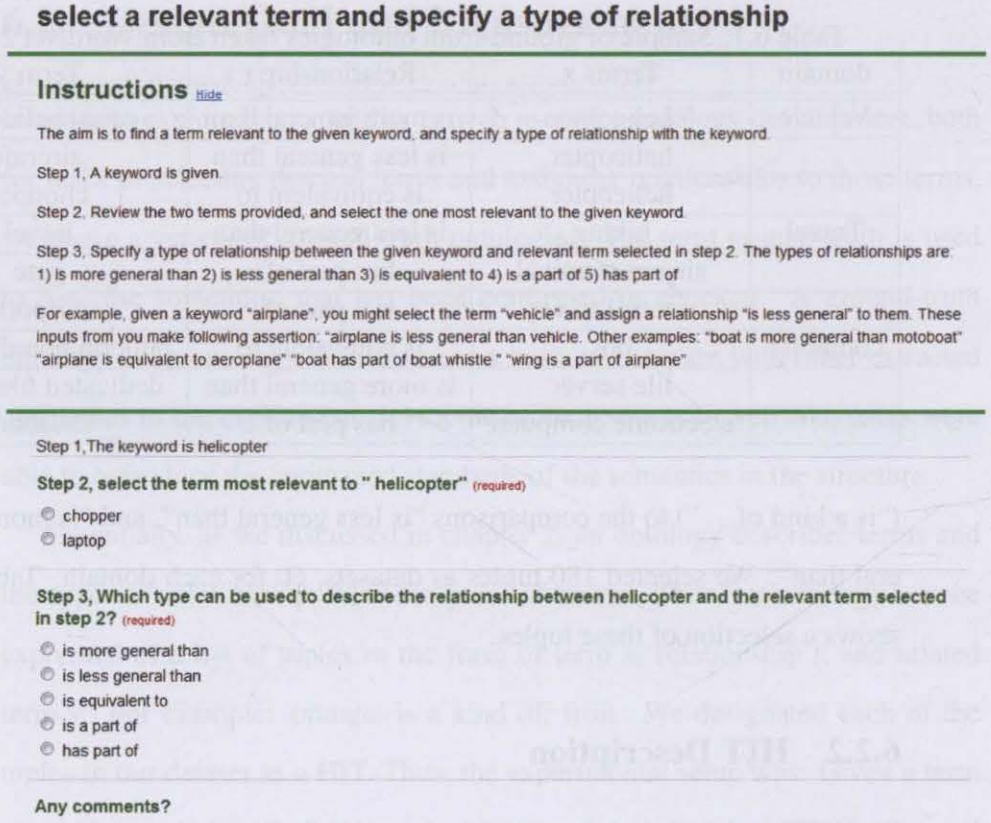


Figure 6.2: A screenshot of a HIT submitted to MTurk via Crowdflower

part of". For example, given the keyword airplane, a Turker might select a related term, vehicle, and assign the relationship "is less general than", thus making the following assertion: Airplane is less general than vehicle. Other examples included: boat is more general than motorboat; airplane is equivalent to aeroplane; boat has part of boat whistle; wing is a part of airplane. This example was shown to Turkers before they started the HIT. Figure 6.2 is an example of an actual HIT used in our experiment.

6.2.3 Worker Recruitment

Our HITs were published on MTurk using CrowdFlower (http://www.crowdflower.com), a third party proxy company that provides access to MTurk for requesters outside the United States. We set up several Turker requirements to restrict access to our HITs. Since the work required disambiguation and conceptualisation of English terms, we only allowed Turkers from countries in which the usage of English is widespread among the general population. We assumed that Turkers from countries where English is commonly spoken would be more familiar with these concepts than people from countries in which English is rarely used.

6.2.4 Remuneration and Cost

Before we published our task, we needed to arrive at a reasonable payment amount for each HIT. We wanted to ensure that Turkers would choose our HITs and the total tasks would be completed in reasonable time windows. The amount was based on an hourly rate and the estimated time needed to complete a HIT. We ran a calibration test using a tool provided by Crowdflower, which suggested that we give $8.78 (including the commission fees paid to Amazon and Crowdflower) for the travel domain task (see 6.4). This price was based on $2 per hour (a common hourly pay in MTurk), around $0.01 per HIT.

We increased the payment to $0.02 per HIT for the second experiment (the vehicle domain task) and kept the $0.01 rate for the third experiment (the computer domain task).

6.3 Quality Control

Responses from the crowd often come with noise. There are several mechanisms that can be applied to avoid or limit distortions in the data, including gold standard test and measure of agreement. In addition to these two solutions, we also applied other techniques to normalise the datasets, such as soliciting comments from Turkers and continuously monitoring the input results while the HITs were occurring.

We organised the above-mentioned mechanisms and functions according to a best practices guide for requesters [2] provided by MTurk . In the subsections that follow, key components of our experimental task design will be presented.

6.3.1 Overview of the quality control workflow

Figure 6.3 describes the quality control workflow that a requester can use for each task. It allows us to monitor the work and thus improve the efficiency and accuracy of HIT performance.

There are two parties in the workflow process: requesters and workers. A requester can select Turkers for a particular task according to certain imposed qualifications. By default, a HIT is open to all registered MTurk users who become workers.

At the beginning, there may be several workers looking at the published task descriptions and intending to accept the HIT. The Turker is allowed to accept and start the HIT only after the requirements are met. The qualifications

[2]http://mturkpublic.s3.amazonaws.com/docs/MTURK_BP.pdf

test submissions are evaluated according to this process: Turker judgments on gold standard questions are automatically evaluated and a qualification score is generated based on the number of correctly answered questions. If the Turker fails most of the gold standard questions and gets a low score, their HIT will be rejected. Workers can appeal the rejection decision. The requester can either reverse the decision, or keep the rejection. All judgments from qualified workers will be added into the collection. This process repeats until the task receives enough judgments.

The detailed quality control process is incorporated into the task procedure as follows:

1. A worker finds and reviews a task listed in MTurk.

    (a) Good task instruction at the HIT interface ensures workers understand the questions and the directions about what is or is not acceptable.

2. The worker can accept the task if the basic qualifications are met.

    (a) Only Turkers who meet the basic requirements can accept the task. For instance, their location must be in the list of countries that allows them to participate.

    (b) Multiple workers are allowed and required to work on the same HIT. The maximum number of workers is set by the requester.

3. The Turker performs the task and submits a judgment.

(a) If the HIT is one of the gold standard questions, then

i. We compare the judgment with the reference answer. If the judgment matches the reference, then this worker's qualification score increases. Otherwise, the score decreases. For example, if a Turker correctly answers the first gold standard question, the score is 100%. If the Turker misses the second gold standard question, the score will decrease to 50%.

ii. We reject all submissions having a score lower than 50%, unless the Turker is working on their first gold standard question and the answer is wrong. We mark those who don't meet the the gold standard qualifications as untrusted workers and at the same time their judgments become untrusted.

iii. We stop the job if two-thirds of the total judgments are untrusted. An alert email is then sent to the requester for manual intervention.

(b) If the HIT is not a gold standard questions, then

i. We add the judgment to the collection of submissions.

4. Communication during task processing

(This step is optional.)

(a) Workers can make suggestions and comments on the tasks.

(b) Workers can ask for a review of rejected judgments.

After reading a rejection appeal from a Turker, the requester can forgive the submission (reverse the rejection). The qualification score will be be automatically adjusted. Or the requester can let the rejection decision stand. A reply message is sent to the Turker.

(c) A requester can modify the gold standard questions if worker comments or appeals indicate it is necessary

5. Results aggregation

   (a) If a judgment is the same as others, which means all workers agree on the judgment, this input is combined with similar inputs for this HIT.

   (b) If a judgment is not the same as other judgments, which means disagreement exists, this input is held for further analysis.

   (c) After all judgments are submitted, the judgment with a majority agreement is selected as the combined result.

6. The previous steps are repeated until the minimum required number of trusted workers completes the task, or the HIT expires.

7. Turkers whose final qualification score is higher than 50% are considered trusted workers and their submissions are approved.

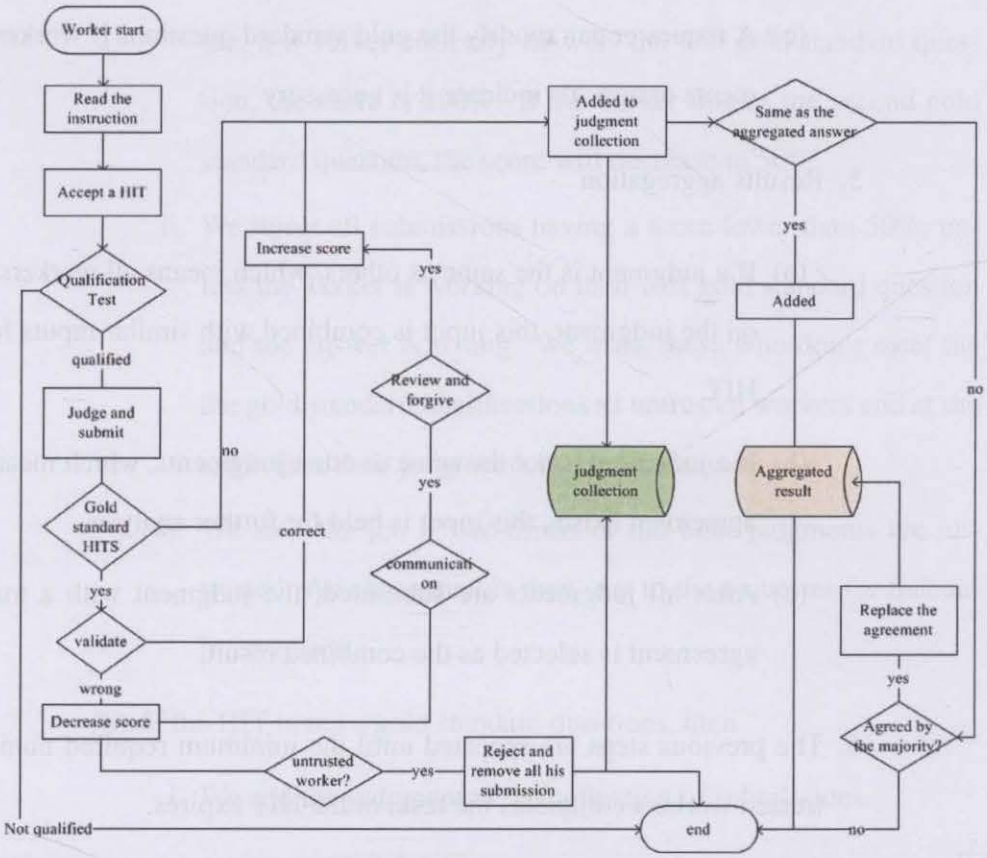


Figure 6.3: Quality control process

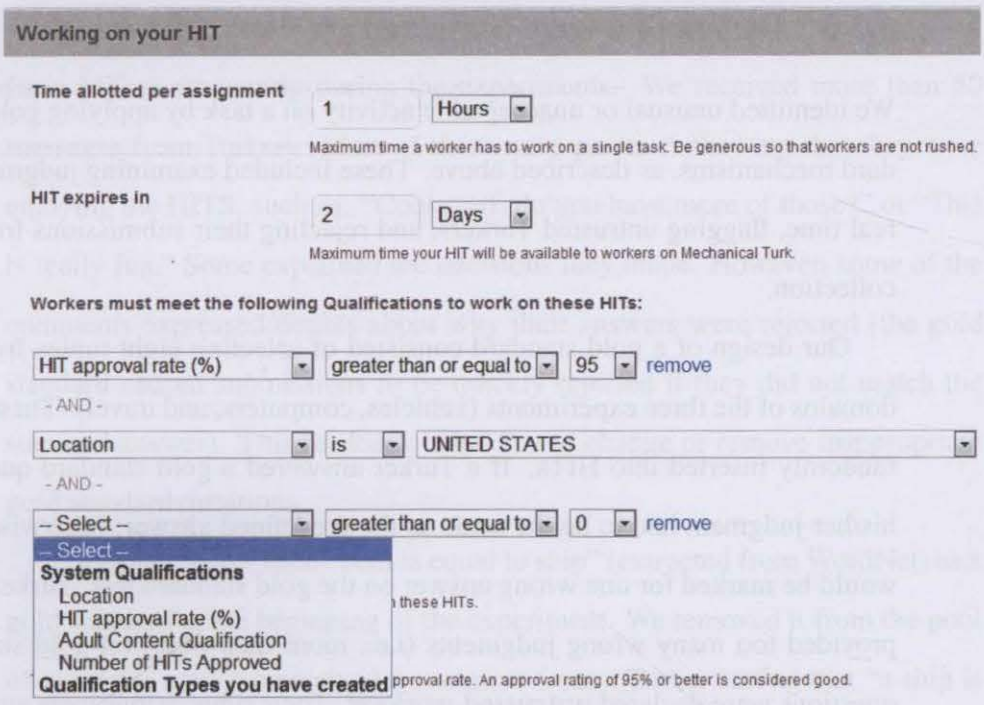


Figure 6.4: Qualifications requirement setting in MTurk

6.3.2 Qualifications

In addition to location setting, MTurk also provides another qualification, a HIT approval rate. It is the ratio of a Turker's accepted HITs compared to the total number of HITs submitted since they registered with MTurk. We set the HIT approval rate to be greater than or equal to 95%, which is recommended by MTurk as a good performance record. See Figure 6.4 for a screenshot of the qualifications settings.

6.3.3 Design of a Gold Standard

We identified unusual or unacceptable activity on a task by applying gold standard mechanisms, as described above. These included examining judgments in real time, flagging untrusted Turkers, and rejecting their submissions from the collection.

Our design of a gold standard consisted of selecting eight tuples from the domains of the three experiments (vehicles, computers, and travel). These were randomly inserted into HITs. If a Turker answered a gold standard question, his/her judgment had to be the same as the predefined answer; otherwise, s/he would be marked for one wrong answer on the gold standard test. Turkers who provided too many wrong judgments (i.e., more than 66%) on gold standard questions were declared **untrusted workers**. Their submissions were automatically rejected and were not included in the final dataset. The other turkers were treated as **trusted workers** and their submission are accepted for further analysis.

6.3.4 Communication with Turkers

We did not simply publish the tasks and wait for them to be completed. Communication with Turkers helped us to realise problems in the design of the HIT and gave us a chance to do some adjustments. Turkers can communicate with the requester by using a comment box inside the HIT, or by sending email to the requester (via an interface function provided by MTurk). They were able to ask questions about an individual HIT or comment on the tasks in general.

In order to increase the quality of our evaluation, we checked the email from MTurk frequently during the experiments. We received more than 50 messages from Turkers. Most of them were positive, showing that they were enjoying the HITS, such as: "Cool stuff, do you have more of those?" or "This is really fun." Some explained the decisions they made. However, some of the comments expressed doubts about why their answers were rejected (the gold standard caused submissions to be quickly rejected if they did not match the standard answer). This feedback helped us to change or remove inappropriate gold standard questions.

For example, we used "boat is equal to ship" (extracted from WordNet) as a gold standard at the beginning of the experiment. We removed it from the pool of questions after we received a comment from a Turker stating that "a ship is not the really the same as a boat. Both ships and boats are vessels for travelling on water, but a boat is more general than a ship". The live statistics also showed that this question had a high error rate, which meant that something about it was confusing to many workers. Therefore, we decided it was not suitable as a gold standard question.

We responded to all general queries in a timely manner so that the workers would better understand the general task or questions asked in the task. We think that this improved the quality of the answers on our other HITs.

After receiving several comments about the design, we also became aware that the task wording needed improvement. For example, one Turker said, "I have to say that the natural order of reading the questions is 'keyword' then

'match', and my default sentence, from years of test taking, is to use the 'answer/match' at the beginning of the sentence and compare it (more/less/equal) to the keyword." However, we were not able to change the interface or the design of the task after the HITs were published.

6.3.5 Intervention during Task Submission

In addition to adjusting our tasks based on Turker feedback, we also manually banned bad or untrusted Turkers and rejected their submissions. During the processing of a HIT, we attempted to discover untrusted users (other than those found by the gold standards) or those who were cheating on their submissions. For example, submissions from Turkers who tended to have low agreements were investigated even though they might have had a high trust rating. They were flagged and banned from participating if cheating or poor behaviour was found. Submissions could be rejected manually, which freed financial resources and helped get new Turkers to participate in the task.

6.3.6 Combining Inputs Based on Agreements

The above-mentioned techniques helped to remove poor results from the data. However, variances in HIT responses from multiple Turkers were still present. Agreement is an important parameter for combining the judgments/submissions from trusted Turkers and working out a common concept from them.

The maximum number of Turkers who were allowed to work on a particular HIT was set to eight Turkers for each HIT.

Agreement describes the percentage of Turkers who have the same response to a HIT. It is a proxy for HIT accuracy, which means that high agreement usually signifies higher accuracy. Low agreement indicates that the HIT may be too difficult for the participating Turkers.

The formula below offer one approach to establishing the common judgment of a specific input, together with the level of agreement when an disagreement occurs.

common judgment = max(agreement of claim 1, agreement of claim2,.. agreement of claim n in the group)

For example, perhaps we received two different judgments from a set of eight Turkers who worked on the same HIT. Six of them claimed that "A is a kind of B" (claim 1), while only two of them claimed that "A is equivalent to B" (claim 2).

The agreement of claim 1 is 75% (six out of eight) and claim 2 is 25% (two out of eight). Based on the rule of majority, the claim with greater agreement –"A is a kind of B" (claim 1) – would be kept as the result for that group of submissions. The agreement percentage does not have to be more than 50% to form a majority. As an example, let's say eight workers did the same HIT and their judgments were four different claims. The number of agreements that were the same for each of the claims was [1,2,2,3]. In this case, the claim that three workers agreed on is 38% (3 out of 8). Whenever a tie vote occurs, another round of judgments will be introduced to resolve it.

Table 6.2: Statistics of the HITs and Turkers

Domain	HIT	Judgments		Unique Trusted Turkers
		Trusted	Untrusted	
travel	60	525	1135	17
vehicle	60	550	685	18
computer	60	520	450	17
Total:	180	1595	2270	52

6.4 Results

To explore the abilities of Turkers to classify different domains, we conducted Experiment 1, Experiment 2, Experiment 3 asynchronously, starting with the travel domain, followed by vehicle, and then computer, respectively.

Table 6.2 shows summary statistics of the data related to the three experiments. More than 250 Turkers (including 52 trusted Turkers) accepted the HITs. The judgments were, in the first instance, classified as trusted or untrusted. For example, the vehicle domain consisted of 60 HITs and received a total of 1,235 judgments, of which only 550 judgments were deemed trusted.

The raw data comprised a list of judgments. A judgment is the result of a Turker's work on a specific HIT, which can be extracted and expressed as an extended tuple that includes the Turker's identity (workerID), a term x, a relationship r, and a related term y. For instance, the first line of Figure 6.5 shows a judgment made by a Turker with Worker ID 3606 who added the following data: "computer, is more general than, analog computer". (Other identifiers, such as response create-time and location, were also recorded.) Here is another judgment about the same x and y terms: "277021, computer, is equivalent to,

_unit_id	_created_at	_started_at	_trust	workerid	country	_city	terma	relationship	termb
31312346	8/20/2010 13:39:52	8/20/2010 13:39:27	0.923076923	3606	IND	Madras	computer	is more general than	analog computer
31312346	8/20/2010 13:44:45	8/20/2010 13:44:04	0.80952381	277021	IND	Hyderabad	computer	is more general than	analog computer
31312346	8/20/2010 13:46:18	8/20/2010 13:45:35	0.916666667	609323	ZAF	Pretoria	computer	is more general than	analog computer
31312346	8/20/2010 13:46:54	8/20/2010 13:45:53	0.916666667	236790	AUS	Kenwick	computer	is more general than	analog computer
31312346	8/20/2010 13:56:52	8/20/2010 13:56:18	0.909090909	242537	MLT	Marsa	computer	is more general than	analog computer
31312346	8/20/2010 13:57:07	8/20/2010 13:39:15	0.80952381	277021	IND	Hyderabad	computer	is equivalent to	analog computer
31312346	8/20/2010 14:16:55	8/20/2010 14:16:33	0.875	133190	GBR	Derby	computer	is more general than	analog computer
31312346	8/20/2010 14:45:42	8/20/2010 14:40:36	0.909090909	591472	USA	Fenton	computer	is more general than	analog computer

Figure 6.5: Submissions from 8 Turkers on a same HIT

analog computer".

6.4.1 Overall Quality

An analysis of the trusted judgments demonstrates that there was a high level of agreement in the HITs: up to 97% on the selection of related terms task, more than 48% on the determination of relationships task (see Table 6.3), and higher than 40% agreement on both of them. The agreement percentage means that for each HIT, more than three Turkers gave the same response to the same HIT.

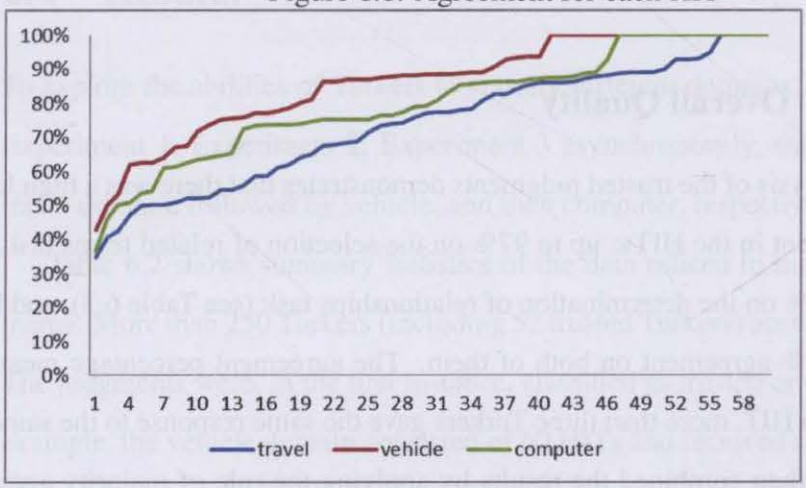
We then combined the results by applying the rule of majority agreement described in section 6.3.6. For example, "computer, is more general than, analog computer" is the result of the combination of the eight judgments in Figure 6.5. With this combination, we finally assembled an aggregated view for each of the HITs. By comparing all these aggregated results with the original datasets, we found that Turkers reproduced the ontology tuples with more than 90% accuracy.

The results also indicate that higher agreement leads to higher accuracy. For example, if we only accept judgments that have greater than 70% agreement, the accuracy of the resulting ontology can reach up to 98%.

Table 6.3: Agreement and Accuracy

domain	agreement		accuracy based on the number of judgments in agreement
	related term	relationship	
vehicle	93.26%	59.21%	95%
travel	94.78%	48.75%	95%
computer	97.12%	58.27%	90%

Figure 6.6: Agreement for each HIT



In short, these experiments show that it is feasible to use MTurk to employ web users for timely and cost-effective development of a large-scale ontology.

6.4.2 Work Distribution

In our experiments, the quality control mechanisms described in section 6.3 successfully identified a total of 2,270 untrusted judgments, equivalent to 58.8% of 3,865 judgments. See Figure 6.7.

For each experiment, there were many untrusted judgments in each domain.

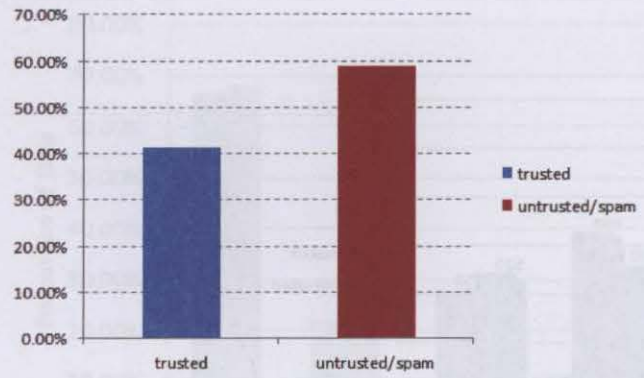


Figure 6.7: In total, 58.8% of the work are produced by workers with low qualification score and marked as untrusted work. Only 41.2% are trusted.

As shown in Figure 6.8, the untrusted judgments climbed to two-thirds of the total in the travel domain. We manually checked the untrusted judgments and did not find any good submission inside. This result is disturbing and it should serve as a warning to researchers who use crowdsourcing for their experiments. It also shows the importance of using quality control mechanisms to identify and remove spurious judgments.

To further investigate the behaviour of workers making judgments, we ran a live visualisation on the judgments made by Turkers over time. See Figure 6.9. Those untrusted judgments were removed from the final results. We notice that some Turkers quit the HIT after they made their first wrong decision.

A further analysis was conducted on HITs from trusted Turkers (with a trust value between 0.5 and 1). Data for the vehicle domain shows that the majority (89% cumulative) of Turkers displayed a high trust value ($\geqslant 0.8$). These Turkers with higher trust values also submitted more HITs than the Turkers with lower

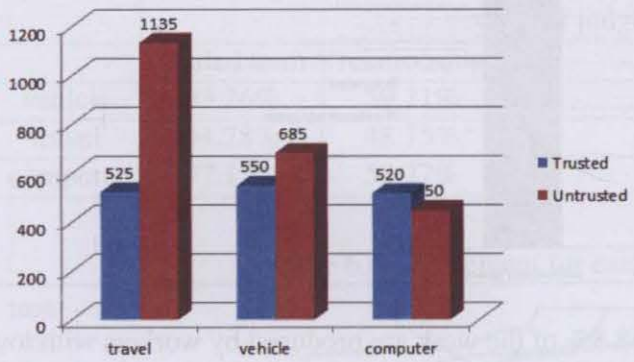
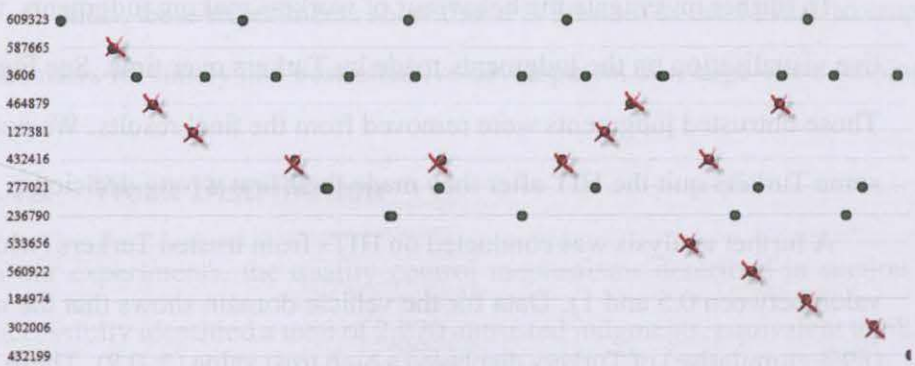


Figure 6.8: A drill down analysis at the untrusted and trusted judgments collected for different domains

Figure 6.9: Live stats: Each row indicates judgments made by a Turker over time. Untrusted judgments have an orange cross and trusted judgments have a green dot.



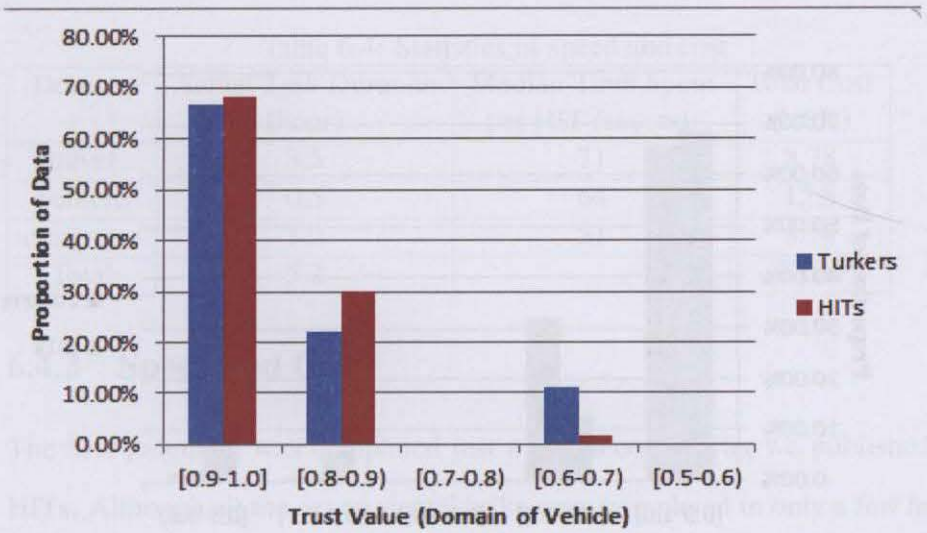


Figure 6.10: Quality and Workload of the Turkers (Domain of Vehicle)

trust values (< 0.8). A cumulative value of 98% of the HITs were submitted by Turkers with a trust value greater than 0.8. Figure 6.10 shows the distribution of trust values of the Turkers who submitted them.

We also observed this phenomenon in the other two experiments. In both the computer and travel domains, more than 97% of the HITs were submitted by Turkers who had trust values above 0.8. See 6.11 and 6.12.

From the work distribution analysis, we hypothesised that while a large portion of Turkers provided untrusted work, they could be identified with certain quality control mechanisms and then removed from the final results. In any case, all three experiments retained a large amount of data that came from Turkers with very high trust values.

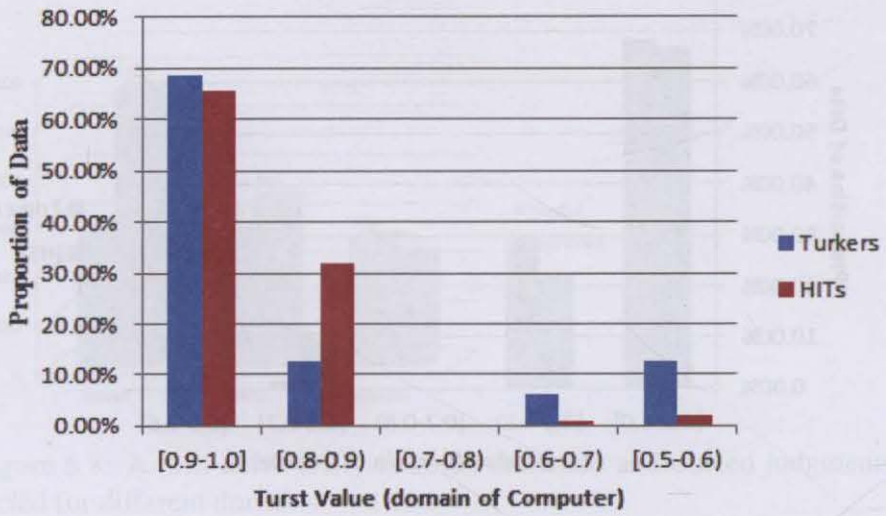


Figure 6.11: Quality and Workload of the Turkers (Domain of Computer)

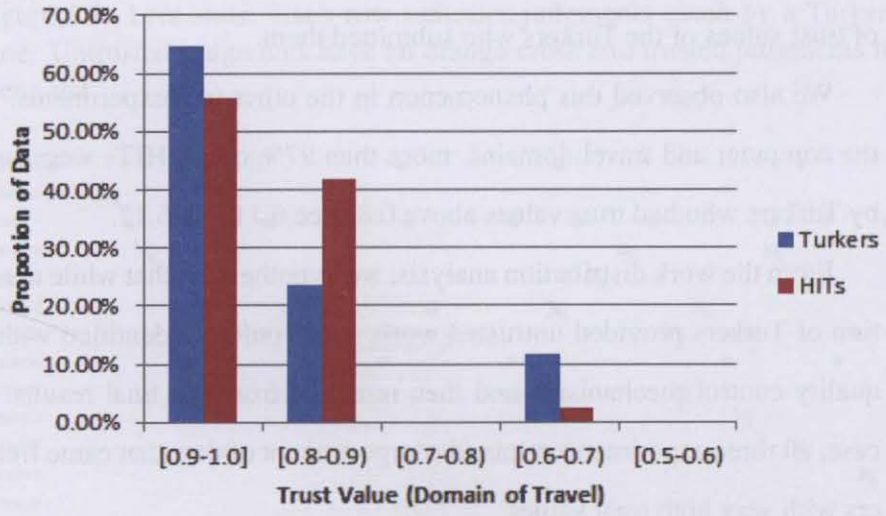


Figure 6.12: Quality and Workload of the Turkers (Domain of Travel)

Table 6.4: Statistics of speed and cost

Domain	Actual Task Duration (hour)	Median Time Spent per HIT (second)	Total Cost (USD)
travel	3.5	71	8.78
vehicle	0.5	64	15.8
computer	1.2	51	8.78
Total	5.2		33.36

6.4.3 Speed and Cost

The first judgment was completed just a few seconds after we published our HITs. Although all the experimental tasks were completed in only a few hours, time durations were different for the different domains (see Table 6.3). The first experiment on travel was finished in 3.5 hours, while the second experiment on vehicles was completed in only a half hour. This could be the effect of a higher payment because we had almost doubled the offer in the second experiment. However, price may not be the only factor that affects speed. For the computer domain experiment, the task completion time was shortened to 1.2 hours after we cut the pay from $15.80 down to $8.78, but it was still completed sooner than the travel domain experiment, which paid the same amount.

Lower-wage jobs may take longer to complete because fewer people are interested in those jobs. However, the analysis shows that Turkers do not spend a shorter amount of time on a HIT because it pays less. The scatter chart in Figure 6.13 depicts the overlap of average time spent on the HITs. See Figure 6.14 for a more detailed description about time spent on each HIT. In the vehicle domain experiment, the median response time was 64 seconds, which is less than the 71 seconds spent on the travel domain, and greater than the time spent

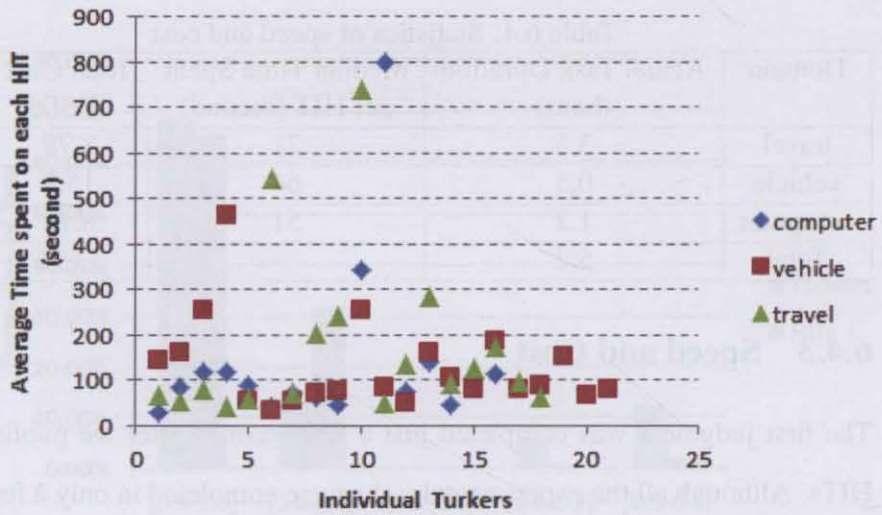


Figure 6.13: Average time spent on each hit by individual Turkers

on the computer domain (51 seconds). We also observed a minimum time of 18 seconds in both the vehicle and computer domains, and 21 seconds in the travel domain. Even so, there were still many judgments that took several minutes.

6.5 Discussion

6.5.1 Our Experience

We have reported the results of a set of experiments designed to explore the feasibility of using crowdsourcing for ontology development, including domain term selection and relationship assignment. We assessed the ability of non-experts to solve these two problem tasks by asking workers from Amazon Mechanical Turk (MTurk), using Crowdflower as a management service,

Time spent for each HIT (computer)		*travel*		*vehicle*	
Mean	127.6942	Mean	114.4933	Mean	132.0127
Standard Error	10.03116	Standard Error	5.25044	Standard Error	8.066771
Median	51	Median	71	Median	64
Mode	42	Mode	59	Mode	42
Standard Deviation	228.7457	Standard Deviation	120.3027	Standard Deviation	189.1825
Sample Variance	52324.59	Sample Variance	14472.74	Sample Variance	35790.03
Kurtosis	10.57978	Kurtosis	5.930251	Kurtosis	10.23021
Skewness	3.396565	Skewness	2.503264	Skewness	3.220095
Range	1100	Range	577	Range	975
Minimum	18	Minimum	21	Minimum	18
Maximum	1118	Maximum	598	Maximum	993
Sum	66401	Sum	60109	Sum	72607
Count	520	Count	525	Count	550

Figure 6.14: Statistics for time spent on each HIT (Unit: second)

to reproduce a variety of ground-truth ontologies. These ontologies, covering the domains vehicle, travel, and computer, are all subsets of WordNet. The experiments were completed in a short time at low cost.

The results indicate that the crowd achieved greater than 93% agreement in the recognition of related terms and greater than 48% agreement about the type of relationship between each pair of terms. By comparing the ground-truth ontologies with the results agreed to by a majority of workers, we also found that these workers were able to reproduce the WordNet ontologies with an accuracy level of more than 90%.

In short, our results suggest that markets such as Mechanical Turk are good sources of on-demand labour for ontology building. In future experiments, we plan to use crowdsourcing to compensate for the deficiencies in the WordNet ontology.

6.5.2 Experts vs. Crowds

In our experiments, we assumed that the WordNet ontology was absolutely correct since it was created by experts. However, there is some evidence that this may not always be the case. For example, in WordNet the word "certificate" is a synonym of "certification". But one Turker argued that certificate has a slightly different semantic, where a certificate is the outcome of certification. "I can see where 'equivalent' is an appropriate answer, but my reasoning was that a certificate was part of the certification process...you get a certificate once you're certified...making a certificate a part of certification." Another example is the relationship between "rent" and "lease". These terms are also synonyms in WordNet. However, one Turker contested that "Rent and lease are two types of arrangements. Not sure how they are related". In certain circumstances, it is true that renting and leasing are different.

6.6 Summary

In this chapter, we have brought in the human component, and explored the possibility of using non-experts to help build ontologies. Our experiments attempted to aggregate the knowledge of web users in general, using Amazon Mechanical Turk (Mturk) as a crowdsource. The experiments were completed in a short time, at low cost, with more than 90% accuracy.

Chapter 7

Improving Semantic Search by Integrating Crowdsourcing into Ontological Service

7.1 Introduction

We have proposed a semantic search architecture based on ontological structures extracted from folksonomies in order to overcome problems in collaborative tagging systems (see chapter 5). Our solution provides intelligent access to social media with the abilities of query disambiguation, query expansion, content categorization, and navigational browsing. However, problems related to accuracy and ongoing ontology evolution still persist due to rapid changes in community knowledge and the limited power of machines. This can result in

poor performance for ontology-based semantic search and browse functions.

In the experiments reported in chapter 6, crowdsourcing was shown to be a promising alternative for ontology development. By distributing related-term selection and relationship assignment tasks that are traditionally done by experts to workers from platform such as Amazon Mechanical Turk, we obtained very high accuracy in the aggregated results with short completion times and low cost. Our findings suggest that online users can be good sources of on-demand labour for ontology development.

In this chapter, we present the OntoAssist system architecture, an integrated search and navigation solution that not only exploits the power of the machine to automatically extract an ontology, but also uses human input via crowdsourcing to integrate the knowledge gained from online search. It introduces a sustainable motivation level and expands the labour source from a limited number of paid workers to vast numbers of public Internet users. Ontology evolution tasks are blended seamlessly with a public user's daily search activities. With this design, we can motivate online participation from Internet users. Our design lets people refine their search query results with a few simple clicks, and specify relationships between a query keyword and relevant terms. The initial ontology can thus evolve and provide better search results to all internet users.

OntoAssist can be integrated with the APIs of most of the existing search engines, such as Google (http://www.google.com), Microsoft Bing (http://www.bing.com), or Yahoo! (http://search.yahoo.com). For our research, we implemented the OntoAssist tool through the Yahoo! BOSS API. It is available online as a demonstration at www.hahia.com.

7.2 Conceptual Model and Design Considerations

Augmentation as a core principle of complex system design has been championed by Engelbart (among others) for over fifty years. This is in contrast with the "automation" theme that had been dominant in information technology-related fields for a long time. Augmentation concept entails the synthesis of technologies and systems for manipulating information and the exercising of human intellect to improve individual and group processes and knowledge work (Engelbart and English 1968). Internet-related developments have already created the conditions for this vision to be realized and crowdsourcing research is taking it further by making it possible to effectively aggregate and combine the inputs of a large number of human intellects. The logic and value of such aggregation has attracted much attention (Sunstein 2006).

For the past several years, approaches in the area of ontology and semantic search have been oriented towards exploiting social media to conduct efficient, productive exploration and retrieval of online information. These exercises, including our efforts described in chapters 5 and 6, have tested a number of techniques for developing ontologies that constitute the back-end of semantic search knowledge bases which include data mining, social networking analysis, statistical analysis, and the recent application of crowdsourcing. However, the most important question that has not yet been answered well is the integration of human and machine systems to achieve a high level of accuracy in knowledge acquisition, with minimal response time and reduced maintenance costs.

For purposes of human and machine integration, the basic idea is to let

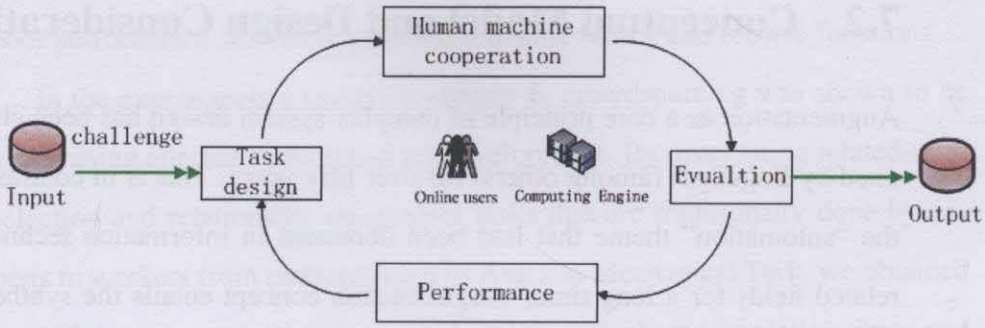


Figure 7.1: Conceptual model

machines do what they can do well, such as generating a preliminary ontological structure, and then let humans improve it. Figure 7.1 shows a high level overview of our proposed conceptual model.

This problem solving approach combines the computational power of machine with human computation and creativity. We employ machines to identify potential domain terms and relationships among them, and then use the intelligence of human to evaluate and improve the preliminary results. Crowdsourcing provides a collaborative human computation platform that allows tasks to be easily distributed among thousands of participants. This enables problems that are easy for humans but difficult for machines to be solved.

For example, humans can excel at rapid conceptual assertions, such as assigning semantic relationships, but they have difficulty finding highly related terms that are located within millions of tags. In order for humans to make decisions under these conditions, they need the machine to prepare a preliminary structure of related terms. Furthermore, a well-defined web interface is necessary to present the partial structure to the users and coordinate activities

between human and machine.

In this section, we highlight several issues to consider when designing a hybrid system that can fulfill this integration.

1. **Micro-labour market vs community**

   Online micro-labour markets, including Amazon Mechanical Turk, have been providing large numbers of workers for crowdsourcing tasks. As we discussed in chapter 6, MTurk has proven to be a good alternative for many things, including ontology development. However, there are still some negative aspects to using MTurk. Workers complain that they have difficulty finding good HITs to complete, while requesters often receive poor submissions from workers who may not really understand the questions.

   In general, people from special interest communities may have better knowledge about the topics in their field of interest than people outside it, since they share similar resources, preferences, and needs. Therefore, when dealing with specialized domains, such as healthcare or chemistry, interest communities may be better sources for crowdsourcing tasks than Amazon Mechanical Turk.

2. **Monetary reward vs free service**

   The success of any crowdsourcing approach relies on strong and sustainable motivation to attract a sufficient number of human agents. Monetary reward can attract all sorts of participants. But offering payment is only

practical for short term projects, such as the early stages of building an ontology. Providing a semantic search engine that can be scaled up to serve the Internet public at large requires ongoing change and development of the backend ontology. The cost of maintaining long-term worker participation is still untenable.

But there are other incentives that attract people to come and work on the tasks. For example, some people are willing to work in exchange for free services, such as downloading software. Others may volunteer to work on an open system because they can contribute to something bigger than what they can do by themselves. Furthermore, good services can be a bit addictive and attract users to visit them regularly. Even for many of the workers on MTurk, money is not the only reason that motivates them to sit for 10 or 20 minutes completing HITs that pay only one or two cents. They may repeatedly engage in these online tasks because they are challenging and fun.

3. **Performance and Variability**

People and machines display much variability in the speed and quality of their work (Franklin et al. 2011). In our previous experiments we encountered malicious behaviour and received spamming submissions. Uneven quality was observed both among the submissions from different individual workers as well as the input from the same workers who did different experiments.

4. **Job breakdown and task design**

   It is important to break down the whole job and define which are the tasks for humans and which are for the machine. The tasks that we ask humans to complete are those that are difficult for a computer to do. In addition, the human-machine interface must be attractive and easy to use, since people are not willing to spend much time on query interaction.

7.3 System Architecture

Figure 4.1 represents a more detailed architecture of OntoAssist as described below. OntoAssist is a semantic search tool that creates a synergistic partnership between human and machine. It consists of computational methods for extracting preliminary ontological structures from folksonomies, followed by human enrichment and improvement to proivide a better ontological service. As an independent semantic search system, it can do repetitive computing tasks to extract ontological structures from keyword input, and then refine the search results based on the ontology. In the case of ontology evolution, it can acquire and analyse users' input data to improve the structure.

OntoAssist comprises four core modules: a user search interface, an ontology extraction module, an ontology evolution module, and a semantic search module.

1. User Search Interface

   The user search interface consists of the following: A keyword interface,

where users can input search terms; a related terms generator that gathers related terms automatically from the ontology repository and Wikipedia; and a relationship selector for determining possible relationships based on the ontology. A traditional query interface allows users to type in keywords and search for online resources. This is extended with a disambiguation component that lets users choose related terms and assign semantic relationships to them. Users can make explicit the semantics of their query by simply selecting one of the related terms and assigning a relationship between the query keyword and the term.

2. Ontology extraction module

   Using the input query, this module automatically finds a cluster of related tags using association rules mining techniques. The tags are then organized into a partial ontological structure (see chapter 5 for a more detailed description of the algorithms). These partial ontologies are incorporated into the ontology repository.

3. Ontology Evolution Module

   The ontology evolution module records all the relationships and terms selected by users. Users' query logs are recorded as well. All of these term-to-term relationships remain in an unreleased status until they are validated by the system.

4. Semantic navigation module

   The aim of this module is to provide an ontological service so that we

can improve search precision and recall, and provide the user with better navigation based on domain knowledge. By assigning a relationship between the query keyword and one of the related terms, a user is able to express his/her query intent in a format that the machine understands. Thus, the precision can be improved by performing an advanced search that matches additional related terms. It also removes pages that have unwanted terms from other domains. 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, the refined results can be pushed to the user automatically without the need to refresh the web page. Semantic navigation is a plus to improve search results by letting users quickly explore a concept in the relevant domain they have chosen.

In the following sections, we focus on three key features of OntoAssist: a method to attract sustained input from the crowd; ontology evolution based on crowdsourcing; and complementary domain knowledge support. Implemention and experimental results are also presented.

7.4 Ontological Service Using Crowdsourcing

7.4.1 Sustainable crowdsourcing motivation

To attract a wide range of Internet users, we piggybacked OntoAssist onto a general purpose search engine. This immediately gave us a large number of candidate participants. To ensure enough traffic, OntoAssist was designed to

provide simple and intuitive semantic navigation over query results. OntoAssist helps a user to locate the desirable result efficiently by filtering out tens of thousands of unrelated entries. Moreover, OntoAssist continues evolving its underlying ontology with the help of user input. Users can see the improvement of the search service over time. This helps to retain existing users and to attract new ones.

The search interface is a web service that integrates the intelligence of the machine directly into human query processing by suggesting related terms and relationships. The ontological service is fueled by visitors to the website who search with keywords, disambiguate their search intent by specifying relationships with the term in question, and receive improved query results.

7.4.2 Crowdsourcing 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. In order to expand the base ontology, OntoAssist aggregates many user inputs from the semantic navigation interface. The ontology evolution model of OntoAssist consists of the following:

1. Eliciting users' knowledge via semantic navigation

   The design of the semantic navigation component in OntoAssist is based on the general search assist tools provided by most search engines. Queries submitted to search engines usually consist of very short keyword phrases. Offering assistance in the search, such as suggesting related terms, is

useful for determining the query intent. 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 results. Popular search assist applications include Yahoo! search assist, Google related search suggestion, Bing related search, and so on. The search assist functions are concerned with 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 the general and semantic associations of terms and their relationships for ontology evolution purposes. The semantic navigation component allows users to express their search intent as a tuple (keyword, relation, related term). For instance, if the original query keyword is python, a user can refine the search via the tuple: (python, is a kind of, programming language).

2. User input aggregation

   We then gather together these terms and relationships from different query sessions. We assume that one expression is correct if a majority of users agree on it. Furthermore, we do not treat all user inputs equally. Analysing query logs helps us to separate users into trusted or untrusted groups for purposes of knowledge collection. We provide an option for users to register and log in for the use of personalized services and to record their behaviours. This makes it easy to distinguish registered users that

are trusted versus the untrusted. We analyse the query log to determine the trustworthiness of anonymous users. Inputs made by trusted users strongly influence the assessment of the collections.

3. Version control and automatic update

   (Noy et al. 2006) presented a framework for collaborative ontology development that was designed for domain experts. We adapted the framework to use in an Internet environment where large numbers of non-experts are able to contribute. The adapted framework has the following features. It is asynchronous: Every user checks out a part of a concept related to his/her own query, edits it, and submits it back to the system. The system is monitored: All changes are recorded, as well as 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 periodically applies the changes to the old version and then releases a new one for further editing. Change conflicts are resolved during the aggregation, using majority rule techniques adjusted by user impact.

7.4.3 Domain Knowledge Support

The semantic representation of user search intent is expressed as a list of terms and a set of possible relationships. The construction of semantic representation follows two simple guidelines: it should be understandable to the user 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

comprehensively and should be able to reflect new and emergent terms.

We attempt to leverage both the extracted ontology and related terms generated during a seach by using Wikipedia's category feature. Clearly, there is always a gap between the number of terms representing user search intent and the amount of existing domain terms. Wikipedia, one of the best and biggest online knowledge databases, can help us infer a user's query intent when certain keywords are not available or are not correctly interpreted in the existing ontology. The article and category links provided in Wikipedia show a kind of semantic connection to each node. Initially, we map the query into the extracted ontology and get related terms and relationships. We also map the query into the Wikipedia link graph to obtain additional relevant terms. Thus, a comprehensive set of candidate conceptual terms and relationships can be developed.

In short, we show how the search intent of an online user can be captured to help evolve the ontology while helping to refine search results. For example, we analyse the query log and find out that several user searches for python agree on these inputs: "Python, is a kind of, programming language", "CPython, is a kind of, python", and "Jython, is a kind of, python". We then incorporate them into an initial computer ontology. This enables the system to expand the query for python to CPython and Jython. It also removes search results that are not a "programming language", such as snake or animal.

7.5 Integrated Application

This section describes the semantic navigation support tool upon which the service is based. OntoAssist is integrated with the Yahoo! search engine at the website www.hahia.com. We show how this system can be used to assist users in performing disambiguation of search intent while contributing to ontology evolution.

The hahia.com website is built on a browser/server model. The user interface is developed with PHP and AJAX and runs on the Apache 2.2 web server. The backend of our platform is a web service that generates related terms from the extracted ontology and Wikipedia. It is developed using Java language and JAWS API, and returns related terms in the XML format. The platform is also a web search engine based on Yahoo! Search BOSS framework, which utilizes the entire Yahoo! Search index, ranking, and relevance algorithm. MySQL is used as a database to store all user and other log information. The entire backend runs on a CentOS 5.3 server.

7.6 Prototype Demonstration

The alpha version of the OntoAssist platform is available at http://www.hahia.com. Figure 7.2 is a screenshot of the hahia.com index page. The main user interface is at the top, including a search box and disambiguation assistance box. There are two separate columns under the disambiguation box. The left hand column lists terms from the ontology base grouped into different domains. When a user

selects one of the related terms and one of the relationships, the system refines the search result and returns a new result in the right column, while at the same time removing from the left column the unselected terms.

The following is a typical example of the semantic search process. In this scenario, a user is doing a search on Flickr.com with the keyword jaguar.

1. A user inputs the keyword jaguar and submits it to the system.

2. Related terms (including cat, car, and band) are generated and displayed.

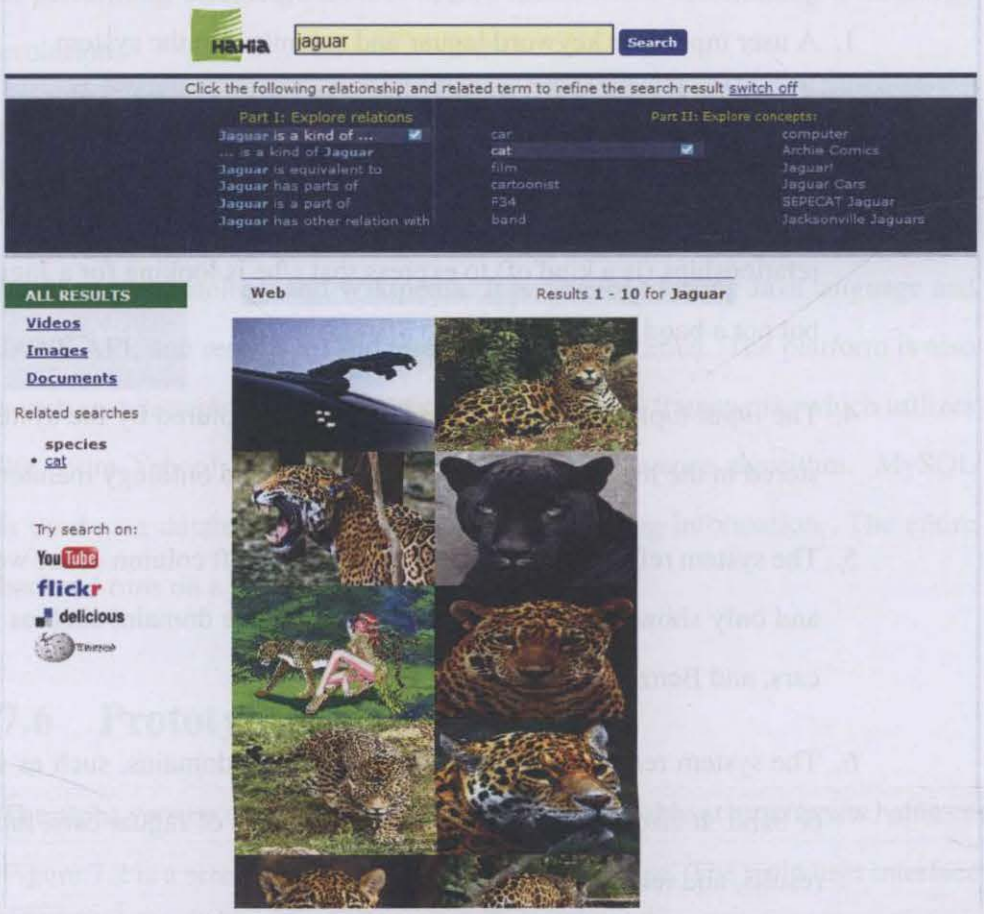
3. The user clicks one of the related terms – car – and then clicks one of the relationships (is a kind of) to express that s/he is looking for a Jaguar car, but not a band or anything else.

4. 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.

5. 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, and Benz.

6. The system removes all the results from other domains, such as species or band. It also expands the search with models of Jaguar cars, ranks the results, and returns them to the user.

Figure 7.2: Screenshot of the OntoAssist display at www.hahia.com. (The data in the image refers to Flickr.com)



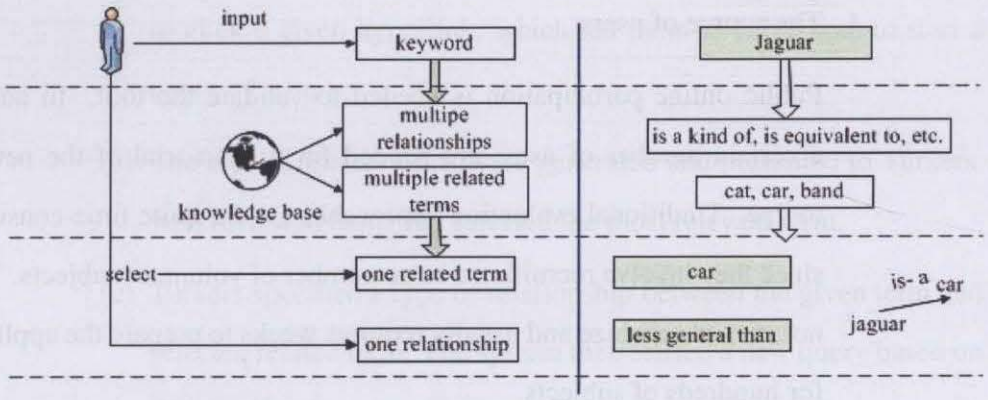


Figure 7.3: An example search of jaguar (car)

7.7 Experiment

In this section, we show how OntoAssist and crowdsourcing can discover new terms and facilitate rapid ontology evolution.

7.7.1 Experimental Setup

To validate the model and our approach, we chose a partial ontology in the computer domain to be our test ontology. We manually queried "computer" in Flickr and received a collection of tags. Applying the algorithms described in chapter 5, we extracted a subset ontology containing 85 terms, including 6 synonyms, 4 hypernyms, one term that has an "is a part of" relationship, 24 terms that fit the "has parts of" distinction, and another 50 "is a kind of" computer terms .

In our experiment, users were only allowed to issue queries with one of these terms.

1. The source of users

   Public online participation is needed to validate the tool. In addition, a certain number of users are needed for a short trial of the new prototype. Traditional evaluation approaches can be quite time-consuming, since they involve recruiting a large number of volunteer subjects. This is not easy to organize and usually requires weeks to prepare the application for hundreds of subjects.

   In our experiment, Amazon Mechanical Turk (MTurk) was introduced as a tool to source public users. To simulate the public at large, Turkers were not told of the purpose of this experiment. They only knew that they were performing normal queries using a search engine with an additional search assistance plugin.

2. Design of tasks

   The task design needs to satisfy three main goals. First, we have to make sure that our search engine is used for each HIT result. Second, the interface design should be simple. Finally, we need a way to measure the quality of submitted work.

   Our MTurk task was titled "select a related term and specify a type of relationship" and it was designed to get human knowledge about a specific term. With a click on the task link, the Turker could see the full description of the job. The sequence was the following:

   (a) A term x was selected from the test ontology. Turkers were asked

to click a given hyperlink, which led them to hahia.com to start a query of term x.

(b) The top 12 related terms were generated and presented to Turkers, who reviewed them and selected the most relevant term.

(c) Turkers specified a type of relationship between the given term and selected related term. The system then started a new query based on the specification. A refined search result was displayed.

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

For example, given a keyword "redhat", the Turker would click through to a list of candidate terms at hahia.com. S/he might select the term "operating system" and assign the relationship "is a kind of" to "redhat". The Turker's input, (redhat, is a kind of, operating system), is called a judgment. We also included an optional field that let Turkers write their comments/suggestions on the use of our OntoAssist platform.

7.8 Results and Evaluation

The experiment was completed in about three hours. Each participant was required to have an MTurk account and he or she could participate only once for a particular group of terms. We collected 1935 judgments from HITs completed by 225 individual Turkers. The Turkers came from eight countries. Most

submissions were from India, and other participating countries included the United States, Romania, the United Arab Emirates, and Macedonia. Figure 7.4 shows the top 100 contributing Turkers, where each bar represents an individual Turker. The numbers below the graph indicate how many judgments were submitted in the experiment.

It is important to assess the quality of inputs and only collect the meaningful ones. We employed the following quality control strategies (as described in chapter 6). First, we put five gold standard tasks in the work pool to assess the quality of Turkers' work. Special keywords (PC, dedicated file server, bulletin board system, analog computer, and CRT) randomly appeared in the queries that were presented to the Turkers. Each of these gold standard keywords has a complete set of correct relationships, and if a user chooses one of the correct relationships, that submission is considered accurate. With this we were able to identify the untrusted Turkers and then exclude those inputs from the final data collection. Turkers who had less than a 40% accuracy rate were recognized as untrusted Turkers in the experiment.

Figure 7.5 shows that Turkers completed the jobs with an average of 68% accuracy against the gold standards. The figure also shows that the trusted Turkers have a significantly higher accuracy rate, 96% on average, than the Turkers who were classified as untrusted, who only had 22% accuracy. In Figure 7.6, we have a further look at each of the five gold standards. It shows that each of the five keywords has almost the same level of accuracy. Finally, the 777 judgments that were made by untrusted Turkers were excluded from the results, leaving 1158 trusted inputs.

Figure 7.4: Judgment per worker

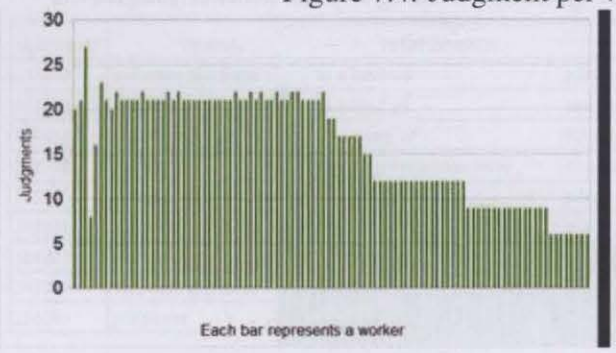
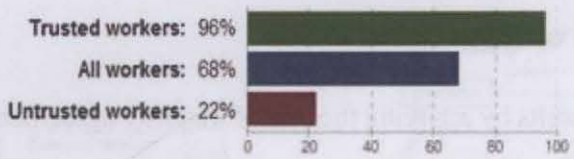


Figure 7.5: Details of trusted and untrusted inputs



While gold standards are helpful in removing untrusted judgments from the collection, agreement is also an important parameter to aggregate the trusted judgments and work out a common concept from them. In our experiment, at least nine different Turkers queried each term. Figure 7.7 shows that the majority of judgments from different users were in agreement.

Figure 7.6: Percent correction golden data

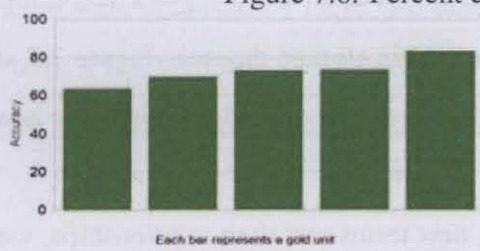


Figure 7.7: Agreement among MTurk worker judgments

7.8.1 Performance of Aggregation

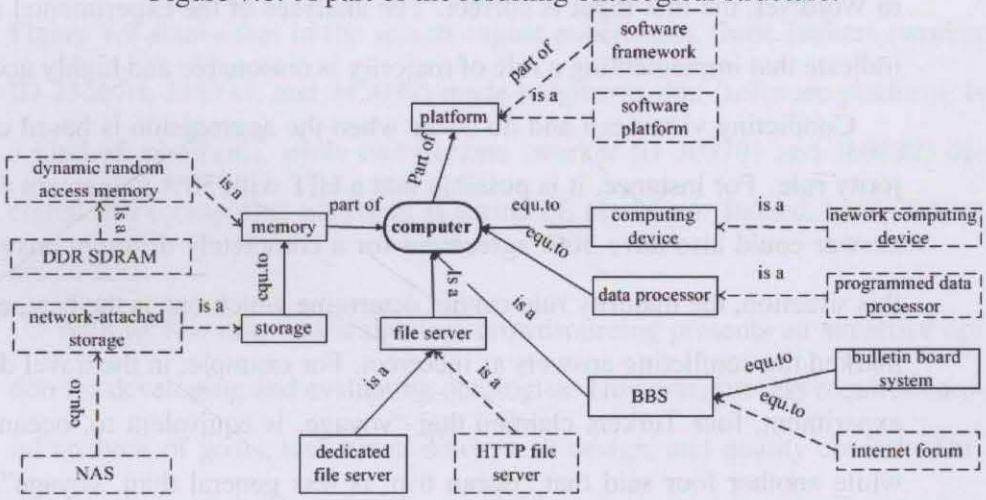
We then aggregated the results by applying the rule of majority agreement. This gives us a composite view. By comparing the combined results with the original ontology, we discovered that 173 additional domain terms from Wikipedia were collected together with their relationships. These additional terms indicate that new concepts are emerging in the computer domain. Some examples are Logitech G51, flash memory, and NAS. Furthermore, the relationships show connections to existing terms in WordNet. Some were marked as "is equivalent to" relationships. Here are some examples: (network-attached storage, is equivalent to, NAS); (dynamic random-access memory, is a kind of, memory); and (floppy disk, is a kind of, removable storage device). Figure 7.7 shows a part of the resulting ontological structure. The dashed lines indicate new terms and relationships that came from user inputs.

After manually reviewing the new terms and their relationships, we found that 89% of them were applicable to the computer domain. The accuracy of the

Table 7.1: An example of user inputs for 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 to	Computing platform

Figure 7.8: A part of the resulting ontological structure.



computer term relationships was much lower, about 62%. It appears that some users were confused by these two descriptions: "... is a kind of term x" and "term x is a kind of ..." The accuracy might be improved if we reword these two types of relationships to "less general" and "more general". See Figure 7.8.

7.9 Discussion

Because different people contributed different knowledge or perspectives to the same question, we received a range of judgments for many HITs. The majority rule was applied to the assortment of judgments made on a specific HIT. For instance, in the travel domain experiment, 50% of Turkers said that "sales booth, is less general than, shop", while 30% of Turkers said that "sales booth, is a part of, shop", and 20% said that "sales booth, is equivalent to, shop". According to WordNet, the first input is correct. The analyses of the experimental results indicate that implementing a rule of majority is reasonable and highly accurate.

Conflicting views can and do occur when the aggregation is based on majority rule. For instance, it is possible that a HIT with 50% agreement for one answer could also have 50% agreement for a completely different answer. In this situation, the majority rule cannot determine which one is the best, and we marked the conflicting answers as incorrect. For example, in the travel domain experiment, four Turkers claimed that "voyage, is equivalent to, ocean trip", while another four said that "ocean trip, is less general than, voyage". The number of conflicts may increase with the growth of the datasets. A further iterative process may be needed to resolve these conflicts. A possible solution would be to automatically publish conflicting HITs and request a new group of Turkers to make fresh judgments.

From our analyses of the experimental data, we noted that the agreement rate was much lower for vague or difficult terms. For instance, there were varying judgments on the relationship between "landing field" and "flight line".

Some said that "landing field, has part of, flight line", which WordNet designates as correct. But others said that "landing field is more general than, flight line". There was also one Turker who argued that "Landing field and flight line are two different parts of an airport, not sure how they are related". Robust strategies must be developed to handle such cases.

We also note that the knowledge expressed by the minority is not necessarily wrong. Some judgments may have fewer agreements simply because the terms have different relationships with multiple related terms. For example, Figure 7.9 shows that in the search engine experiment, three Turkers (worker ID 235670, 248921, and 245099) made judgments that (software platform, is a kind of, platform), while two Turkers (worker ID 305701 and 169892) declared that (computing platform, is a kind of, platform). Indeed, both of them are correct.

With its low cost and scalability, crowdsourcing presents an attractive option for developing and evaluating ontologies. However, success requires careful analysis of goals, task break down, task design, and quality control. This requires the participation of experts and the cost may become significant when advanced skills and intensive knowledge are necessary in more complex domains.

7.10 Summary

In this chapter, we have developed a framework for blending ontology evolution tasks seamlessly with public users' daily search activities. To demonstrate the

framework, we built OntoAssist, a semantic navigation tool. It enhances the native search in CTS, giving users a smart and user-friendly search engine. In particular, the disambiguation feature helps users to search more effectively. At the same time, user input to clarify term meanings is collected to help evolve the underlying ontology. On top of that, OntoAssist can be integrated with third-party commercial search engines and portals such as Google Search, Bing, or Yahoo! Search, using their APIs. As an example, the OntoAssist tool was implemented based on Yahoo! BOSS and released at www.hahia.com. It thus has the ability to provide semantic search and explore most existing resources in CTS.

Chapter 8

Conclusions

8.1 Introduction

We begin with a recapitulation of the significant findings and conclusion of our research. We then proceed to discuss the potential applications and uses of the methods presented and suggest areas where this research can be employed. Based on our observations, we note the limitations and outline future work that is needed to further develop the semantic web by integrating folksonomies and ontologies.

8.2 Research Questions and the Findings

The World Wide Web has undergone significant evolution in the past decade. Web 2.0 (Social Web) introduced the critical feature of user participation and contribution. Its impact has been massive and has led to the rise of a vast array

of social media sites and applications. Semantic web refers to a web of data that makes it possible for machines to understand the meaning of information. This is done through the semantic annotation of web content, building ontologies, and developing reasoning based on those ontologies. In this thesis, we have made a modest contribution to the evolution of a Social Semantic Web vision for the future of the Internet. The emerging Social Semantic Web has a goal of achieving a balanced integration of services provided by Web 2.0 with semantic web technologies. In the Social Semantic Web, aspects of Web 2.0 and semantic web will be complementary to each other, rather than in competition.

Towards this perspective and as partial demonstration and validation of the concept, we introduced a prototype semantic search application for collaborative tagging systems (CTS). CTS has recently emerged as one of the rapidly growing Web 2.0 applications. The informal social classification structure found in CTS, known as a folksonomy, provides a convenient way to annotate resources by allowing users to tag content with any keyword that they find relevant. However, the flat, non-hierarchical structure of the folksonomy, with its unsupervised vocabularies, yields low search precision and poor resource navigation and retrieval. This drawback has created the need for ontological structures that provide shared vocabularies and semantic relations for translating and integrating different sources of online information.

In designing a semantic search application that can overcome the problems in CTS, we dealt with two major research questions: (1) How can ontological structures be extracted from folksonomies in a way that supports effective search and exploration? and (2) How can we gather information and design a

model that enables the ongoing evolution of such a structure?

This thesis proposes an integration of machine computation and crowd-sourcing methods to extract an ontological structure from a folksonomy. A human-machine combination will make it possible for the ontology to evolve automatically as usage patterns change. In this way, the resulting structure can greatly facilitate semantic search and improve the retrieval and navigation of information on the Internet.

Our research was carried out in three phases.

First, an integrated automatic computational method was employed to extract the ontological structures from folksonomies. This method exploits the power of low support association rules mining supplemented by an upper ontology such as WordNet. The machine-based algorithms were applied to four kinds of word-formations found in folksonomies: standard tags, jargon tags, compound tags, and nonsense tags. In CTS, more than half of the tags are in the form of jargon and compound terms. Existing ontologies are not comprehensive enough to determine the relationships among all the tags in a folksonomy. Association rules mining is an unsupervised data mining method used to find interesting associations between datasets. We used association rules to find semantically related tags, which formed the basis for further ontology building. Next, we simplified the Apriori algorithm to find two-item set rules and introduced a new cosine coefficient, which significantly improved the efficiency in low-support mining. Using association rules mining and other techniques, such as token-based similarity, we were able to match tags and incorporate nonstandard terms into ontological structures. Our approach has produced promising

initial results using two datasets from Flickr and CiteULike.

We then introduced the human component, and explored the possibility of using non-experts to help build ontologies. Our experiments attempted to aggregate the knowledge of web users in general, using Amazon Mechanical Turk (Mturk) as a crowdsourcing platform. We assessed the ability of non-experts to solve two main labour-intensive tasks in ontology development – domain term selection and relationship assignment – by asking workers from MTurk to reproduce a variety of ground-truth ontologies. These ontologies for the vehicle, travel, and computer domains, were all subsets taken from WordNet. The experiments were completed in a short time, at low cost, with more than 90% accuracy.

While the two ontology development activities were driven by the same motives (i.e., fast, cheap, and involving user contribution), they presented distinctly different methods. The first computational approach (described in chapter 5) was characterised by power of machine, consisting of a set of algorithms (i.e., association rules mining, natural language processing, and ontology mapping). The second method (explained in chapter 6) employed human knowledge from the crowd (i.e., users from Amazon Mechanical Turk).

Finally, we presented a hybrid human-machine system, called OntoAssist, which allows a systematic approach for this emerging area. Regarding our second question about ontology evolution, maintaining ontologies over time has proven to be a task that is difficult for experts alone. Contributions are needed from a large number of participants. The task also needs to be complemented with computational support provided by the machine. With the integration of

computational and crowdsourcing methods, we achieve more accurate knowledge acquisition quickly and with reduced maintenance costs.

The core of this human-machine system is the ontological service, which enables a semantic search of CTS. OntoAssist is the medium for machine-human integration, by which Internet users participate while they go about doing normal search activities. We elicit their knowledge by presenting them with related terms generated from Wikipedia, and then aggregate their inputs. This allows the ontologies to evolve. A benefit of this model is that it offers sustainable motivation for continued input from the crowd. By integrating purpose-designed HITs into the daily activities of web users, who are searching with a major application like Yahoo!, we can complete our work without the need to pay money. In lieu of monetary reward we offer users a better search experience. Results of experiments using the prototype are presented as evidence of the value and efficacy of the concept.

8.3 Implications for Social Semantic Web Research

There is now considerable interest in the possibilities offered by what has come to be known as the Social Semantic Web. It builds on the developments achieved with Web 2.0 and attempts to integrate the structures from semantic web technologies. However, many significant challenges have to be overcome before we can achieve reliable and mature semantic search as described above. It is not easy to develop ontologies as a semantic backbone for large quantities of user-generated content, since most content is created and annotated based on a

user's own experiences and linguistic styles and preferences.

For researchers, the integration framework presented in this thesis contributes to the theory of Social Semantic Web in general, and provides valuable insights into the development of ontologies as a means to enable semantic search of CTS in particular.

We began with an investigation into the nature of user-generated tags in CTS. The tag collections from folksonomies were divided into four kinds of word-formations, i.e., standard tags, jargon tags, compound tags, and nonsense tags. We developed appropriate ways to extract the semantic relationships from each of these. The ontological service from the tag-based ontologies has resulted in a Social Semantic Web application with better search and exploration capabilities. Researchers can also make use of tag classification to cope with variance in user-generated content, especially the keywords that are used to query search engines and tags in most of the Web 2.0 applications.

This study uses folksonomies as resources for extracting formal ontologies. Our research could potentially help to create robust ontological resources that can speed up the maturation of the semantic web by advancing the state of the art with respect to semantic search. Using a folksonomy as a weak knowledge base to build an ontology provides significant coverage and depth in the relevant domains, and adds valuable annotations to the ontology.

Our work emphasises the importance of people and communities of users as a means to develop and maintain ontologies. Instead of relying entirely on automated machine-based algorithms or teams of experts, we suggest that online users are a good alternative. The promising results we achieved demonstrate

that web users can contribute to ontology development, and it is possible to engage large numbers of human agents through crowdsourcing. Our results show that MTurk can be a source of on-demand labour to build ontologies at a reduced cost, in limited time, without significant reduction in the quality of the ontologies. There are good reasons to believe that an online user-oriented approach offers great potential.

8.4 Implications for Crowdsourcing Research

For researchers, the experiments in ontology development that use crowdsourcing represent a starting point for further research. They provide early insights into a field that will become increasingly important as its benefits become apparent.

The idea of drawing large numbers of people into solving problems is hardly new. For example, the open-source software movement has been successful for many years. The difference is that today's Web 2.0 technologies make it possible to easily gather ever-larger numbers of nonprofessional people to do more complex problems quickly and at reduced cost. However, the work submitted from people online often comes with noise. It is difficult to distinguish the correct answers. This poses a variety of new challenges in interacting with workers and ensuring standards for quality control.

This study investigated several mechanisms that can be applied to avoid or limit distortions in the data. We applied a gold standard test that posed questions in the HITs for which we knew the answers, and prevented Turkers from

continuing the work if they were unable to correctly answer most or all of these questions. Another technique used was a measure of agreement, which collected redundant inputs and assumed that a large number of Turkers agreeing on an answer meant it was correct. In addition to these two solutions, we also applied other techniques to normalise the datasets, such as soliciting comments from Turkers and continuously monitoring the input results while the HITs were occurring. We organised these mechanisms and functions provided by MTurk into a quality control workflow, which will be useful to others who are doing crowdsourcing research.

We developed and implemented the concept of service as a motivation for crowd-based solutions to complex tasks and problems. The success of any crowdsourcing approach relies on strong and sustainable motivation to attract a sufficient number of human agents. Monetary reward can attract all sorts of participants, but it is only feasible for short term projects, such as the early stages of ontology building. Voluntary mass collaboration is the next stage in the evolution of crowdsourcing models. Davis (2011) coined the term crowdservicing. Crowdservicing emphasises applications maintained by the users themselves.

8.5 Implications for Practice

For practitioners, the application of this conceptual framework can improve the query performance of folksonomy-based systems and enhance the organisation of resources. Also, a query disambiguation tool – expressing the intent of a user's query keyword as a tuple – can be implemented in most search engines

to improve search results. A significant effort is needed to develop a backend ontology for a semantic search engine than can be scaled up for general online use. It is hoped that the crowdsourcing approach will complement the computational methods to help create robust ontological resources that can advance the state of the art with respect to semantic search.

The advent of crowdsourcing is revolutionising data collection, knowledge acquisition, and other traditionally labour-intensive processes. Crowdsourcing supplies highly accurate work quickly and at low cost. This study establishes a basic foundation for understanding principles, platforms, and the potential application of crowdsourcing to the development of semantic web applications.

Our OntoAssist application, which works with Yahoo! Search and can be adapted to other major search engines like Bing or Google, also demonstrates the usefulness of crowdservicing. 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). It allows complex problem solving and task execution to take place outside the boundaries of business firms and other institutions (Davis and Lin 2011; Davis 2011). As cognitive technologies such as cloud computing develop further, crowdservicing also offers new startups and other enterprises the opportunity to scale up very rapidly and achieve striking results in short order, becoming 'flash companies' (Woods 2010).

Our implementation of the hybrid human-machine system shows that the integration of these elements achieves a level of service quality that cannot be attained by each alone. The integrated approach used in this research can be

further developed for other new systems that combine human intelligence with automated, machine-processed computation to deliver innovative functionality and services. The techniques we have developed are quite general and should allow overall performance improvement in several areas. Examples include query categorisation, judging search relevance, evaluating the output of language translation texts, annotating training data, handling cases that are difficult for automated systems, offering real-time customer service, performing human query processing to complement database query processing, and more.

8.6 Limitations and Future Work

We note that the ontological structures we obtained could be enriched and deepened by using larger tag datasets, other semantic relations provided by WordNet, and more specialized semantic, lexical resources such as thesauri and subject-specific dictionaries. Since the extraction process takes time, we currently do not provide a live ontology generation online but instead use an ontology extracted from offline Flickr datasets. In the future, we will consider a mechanism for automatically generating an ontology from a folksonomy. A real-time ontology can dynamically evolve itself from community input. Additional work on ranking the generated results will also be useful when conducting searches of large datasets.

We observe that ontologies in some domains such as vehicles, computer,

food and travel are quite straightforward and the relevant folksonomies are usually attributed to new, popular, and colloquial tags which can be relatively easily handled by machine learning techniques. However, in some specialized domains such as healthcare and biology, ontologies can be more complex because of the rapidly expanding terminologies and the emerging relationships of the new terms with existing terms. Folksonomies in these domains may not be discovered as new, popular, and colloquial tags but new terms emerging from research and professional publications, and later adopted in mainstream areas. Identifying these new terms and their relationships with existing terms is a challenging problem because the frequency of their occurrence is sparse. Moreover, the ontology development process requires active involvement of domain experts who provide the relevant conceptual knowledge. These problems suggest a variety of research directions that need to be pursued to make such an integrated ontology extraction system feasible.

One such direction would be to investigate allowing automatic identification of terms that have very low support value in association rule mining process but may have high impact to their domains. The current framework requires that all terms and associations meet certain support and confidence values. If the minimum support is set too high, those rules that involve rare items will not be found. If it is set too low, some uninteresting associations may appear. It would be preferable that an initial model be suggested and the framework be allowed to adapt or extend it so as to best fit the data. Liu et al. (1999) argued that the single minimum support for the whole database is inadequate because it cannot capture the frequency differences of the items in the database. They suggested

a technique that allowed the user to specify multiple minimum supports to accommodate the different frequencies in the database. Instead of using the user specified minimum support, Selvi and Tamilarasi (2009) calculated minimum support for each item set generation and for rule generation. The minimum support threshold was calculated by analysing the frequency of items and their associations in the database at each level, thereby more relevant and meaningful rules were generated.

Social Network Analysis can be also helpful to solve this problem. It provides a mechanism to identify institutions as well as researchers playing major roles as central hubs or located at critical network cut-points in the domain community. New terms and associations emerging from these institutions or researchers would be regarded as high impact ontological terms and be selected for further evaluation. Algorithms proposed by Newman and Girvan (2004) can reliably and sensitively extract community structure including central hubs and critical network cut-points from research communities.

Another possibility would be to attract and to engage sufficient number of qualified experts in a sustainable way. The strength of the crowdsourcing approach relies on the accumulated knowledge from participants. However, expert knowledge is often lacking in online micro-task market such as MTurk since the small rewards and short-term duration of the task posting often limit the variety of expertise available. This makes it difficult to complete certain tasks in complex and specialized domains. It would be helpful to add a new invitation mechanism when we design the crowdsourcing task. For certain domains that

require more expertise to provide input and judgment, we can increase the reward amount to attract the experts and extend the duration of task posting. An invitation indicating an appropriate payment level can be sent to invite them to work on the tasks. A qualification check can be used to limit those workers that carry out the HIT to invitees only.

In the future, it will be helpful to represent the extracted ontologies using RDF data format and the SPARQL query language (Berners-Lee et al. 2001). Then the ontology can be integrated with other semantic web services, and create a collaborative ontology environment that continuously evolves, reflecting the knowledge and usage changes in CTS.

Our experiences indicate that with proper task design, domain knowledge can be elicited rapidly both from paid users, perhaps from online labour markets such as Mechanical Turk, or from volunteer users of an online service such as a search engine. However, our work has been based on a few experiments in the keyword domains of vehicles, computers, food, and travel. Since these domains are general and known to most online users, the labour pool may be inadequate for more specific areas. For specialised topics, such as biology or healthcare, more control over the source of the crowd is necessary. MTurk workers may not meet the requirements since they don't have sufficient knowledge of biology or medicine. One possible solution is to design a qualifications test and only allow those who pass the test to accept the task. This may result in task failure because not enough Turkers accept it, or completion times are too long. Another solution is to integrate crowdsourcing with a special interest community in the domain, to provide a suitable group of contributors.

One major limitation regarding the OntoAssist integrated platform is that the experimental data was collected from a relatively small number of people. Further research is needed that encompasses larger groups. In the future, crowdsourcing experiments with more participants are necessary to collect larger datasets for analysis. In particular, implementing this platform in real scenarios, such as library systems or medical notes management systems, may be a good method to validate the proposed machine-human hybrid model, as well as to test its performance. Moreover, we plan to improve the aggregation techniques by developing strategies to handle the conflicts and disagreements that arise when users evaluate search terms.

We also would like to integrate the OntoAssist application with the API provided by Amazon Mechanical Turk. Then there would be two sources of labour, both paid workers and unpaid users. We could employ paid workers on difficult cases, for example, solving certain conflicts when merging folksonomy terms into the ontology.

8.7 Summary

In summary, we have discussed the theoretical and pragmatic issues concerning integrating Web 2.0 user-generated content with semantic web technologies. Specifically, this thesis described the extraction of ontological structures from collaborative tagging systems, and the subsequent enrichment of an ontology with the addition of semantic search. Our efforts to facilitate semantic search are improving the precision and recall of information in the World Wide Web.

A new model was developed, termed "Ontological Structures Extraction 2.0". It integrated computational and crowdsourcing methods to combine human knowledge from search engine users with the knowledge extracted from folksonomies, using data mining techniques and the relevant terms from an existing upper-level ontology. A prototype hybrid human-machine system was developed as a demonstration that validated the concept.

Our experimental results have demonstrated significant performance improvements in information retrieval through the unification of the seemingly exclusive features of folksonomies and ontologies. They can complement each other by linking to full advantage the colloquial terms from a folksonomy with the semantic relations from an ontology. This combination exploits the semantic relations in the ontological structure to satisfy user queries or navigation requests using terms familiar to them. They can access millions of annotated resources, and translate and integrate them from different sources. Most significantly, we have shown that OntoAssist continues evolving its underlying ontology, based on user inputs. Users can benefit from the improvement of services provided by OntoAssist over a period of time. Finally, these strong experimental results indicate that this approach will make real improvements in the way Social Semantic Web applications are developed.

Bibliography

R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994. 3.3.3, 2, 5.3

R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *the 1993 ACM SIGMOD international conference on Management of data*, 1993. 3.3.3, 2, 5.3

H. Al-Khalifa and H. Davis. Measuring the semantic value of folksonomies. *Innovations in Information Technology*, 2006a. Nov. 3.2

H. Al-Khalifa and H. Davis. Folksannotation: A semantic metadata tool for annotating learning resources using folksonomies and domain ontologies. In *Innovations in Information Technology, 2006*, pages 1–5. IEEE, 2006b. ISBN 1424406749. 3.3.6

Y. J. An, J. Geller, Y.-T. Wu, and S. A. Chun. Automatic generation of ontology from the deep web. *database and Expert Systems IEEE 2007*, 2007. 3.3.5

S. Angeletou, M. Sabou, and E. Motta. Semantically enriching folksonomies with flor. *CISWeb*, pages 65–65, 2008a. 1.6

S. Angeletou, M. Sabou, and E. Motta. Semantically enriching folksonomies with flor. In *CISWeb 2008*, 2008b. 3.3.5

J. Antin and C. Cheshire. Designing Social Psychological Incentives for Online Collective Action. In *Proceedings of Directions and Implications of Advanced Computing; Conference on Online Deliberation (DIAC 2008)*, 2008. 3.4.4

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. *The Semantic Web*, pages 722–735, 2007. 2.4.3

J. Bao and V. Honavar. Collaborative ontology building with wiki@ nt-a multiagent based ontology building environment. 2004. 3.4.1

G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland.* Citeseer, 2006. 3.3.1, 3.3.4

T. Berners-Lee. Information management: A proposal. 1989. 2.2.2

T. Berners-Lee. The world wide web: A very short personal history. *World Wide Web Consortium.[Online]. Available: http://www. w3. orq/People/Berners-Lee/ShortHistory*, 1998. 2.2.2

T. Berners-Lee, R. Cailliau, J. Groff, and B. Pollermann. World-wide web: The information universe. *Internet Research*, 2(1):52–58, 1992. 2.2.2

T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific american*, 284(5):28–37, 2001. 2.4.2, 8.6

K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu. Can all tags be used for search? In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 193–202, Napa Valley, California, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458112. 3.2

K. Bontcheva, J. Davies, A. Duke, T. Glover, N. Kings, and I. Thurlow. Semantic information access. *Semantic Web Technologies. John Wiley and Sons*, 2006. 2.3.4, 2.4.2

D. C. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75–75, 2008. 3.4

S. Braun, A. Schmidt, and A. Walter. Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In *CKC workshop at WWW 07*, 2007. 1.5, 2.4.5, 3.4.3, 3.4.4, 4.3

J. Breslin, A. Passant, and S. Decker. *The social semantic web*. Springer Verlag, 2009. ISBN 3642011713. 2.3, 2.3.4, 2.4.2

P. Buitelaar, P. Cimiano, and B. Magnini. Ontology learning from text: An overview. *Ontology learning from text: Methods, evaluation and applications*, 123:3–12, 2005. 3.3

V. Bush. As we may think. *The Atlantic Monthly (July 1945)*, 1945. 2.2.1

P. Castells, M. Fernandez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, pages 261–272, 2007. 1.4

C. Cattuto, D. Benz, A. Hotho, and G. Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In *The Semantic Web-ISWC 2008*, pages 615–631. Springer, 2008. 5.3

P. Cimiano. *Ontology learning and population from text: algorithms, evaluation and applications*. Springer Verlag, 2006. ISBN 0387306323. 3.3

J. Conklin. Hypertext: An introduction and survey. *Computer supported cooperative work: A book of readings*, pages 423–476, 1988. 2.2.1

J. Davies, A. Duke, and A. Kiryakov. *Semantic Search*, pages 179–209. John Wiley & Sons, Ltd, 2009a. 1.1

J. Davies, A. Duke, and A. Kiryakov. Semantic search. In *Information Retrieval: Searching in the 21st Century*, pages 179–209. John Wiley & Sons, Ltd, 2009b. 2.4.1, 2.4.2, 3.5, 5.7.2.1

J. Davis. From crowdsourcing to crowdservicing. *Internet Computing, IEEE*, 15(3):92–94, 2011. 1.7.2, 3.4, 8.4, 8.5

J. Davis and H. Lin. Web 3.0 and crowdservicing. 2011. 1.6, 8.5

R. Dawkins and L. Pyle. *The Blind Watchmaker.* Penguin Harmondsworth, 1991a. 3.4, 4.2.2

R. Dawkins and L. Pyle. *The blind watchmaker.* Penguin Harmondsworth, 1991b. ISBN 0140144811. 3.4

A. Doan, R. Ramakrishnan, and A. Halevy. Mass collaboration systems on the World Wide Web. *Comm. ACM*, 2010. 3.4

F. Echarte, J. Astrain, A. Córdoba, and J. Villadangos. Ontology of folksonomy: A New modeling method. *Proceedings of Semantic Authoring, Annotation and Knowledge Markup (SAAKM)*, 2007. 1, 2.3.2

K. Eckert, M. Niepert, C. Niemann, C. Buckner, C. Allen, and H. Stuckenschmidt. Crowdsourcing the assembly of concept hierarchies. In *the 10th ACM/IEEE Joint Conference on Digital Libraries*, 2010. 1.5, 3.4.2

D. Engelbart and W. English. A research center for augmenting human intellect. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 395–410. ACM, 1968. 2.2.1, 7.2

J. Euzenat and P. Shvaiko. *Ontology matching.* Springer Verlag, Heidelberg, 2007. 1.3, 2.3.4.3, 3.3

D. Fensel, F. Van Harmelen, I. Horrocks, D. McGuinness, and P. Patel-Schneider. Oil: An ontology infrastructure for the semantic web. *Intelligent Systems, IEEE*, 16(2):38–45, 2005. 1, 2.4.1, 2.4.2

M. Franklin, D. Kossman, T. Kraska, S. Ramesh, and R. Xin. Crowddb: Answering queries with crowdsourcing. *Proceedings of SIGMOD 2011*, 2011. 1.5, 3

J. Gallaugher. Crowdsourcing in information systems: A management guide to harnessing technology. *Flat World Knowledge*, 2010. 3.4.4

D. Gasevic, D. Djurić, B. Selic, and V. Devedžić. *Model driven engineering and ontology development.* Springer-Verlag New York Inc, 2009. ISBN 3642002811. 2.4.2

C. Gentry, Z. Ramzan, and S. Stubblebine. Secure distributed human computation. In *the 6th ACM conference on Electronic commerce*, page 164, 2005. 3.4

F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an algorithm and an implementation of semantic matching. In *ESWS*, pages 61–75. Springer-Verlag Berlin Heidelberg, 2004. 2.4.3, 5.4

S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. *Journal of information science*, 32(2):198, 2006. ISSN 0165-5515. 3.2

S. A. Golder and B. A. Huberman. The structure of collaborative tagging systems. Technical report, HP Labs technical report, 2005. 2.3.4, 2.3.4.1, 2.3.4.1, 2.3.4.2, 2.3.4.3, 2.5

C. Grady and M. Lease. Crowdsourcing document relevance assessment with

Mechanical Turk. In *NAACL Workshop on Creating Speech and Language Data With Amazon¡¯s Mechanical Turk*, 2010. 3.4.2

T. Gruber. Ontology of folksonomy: A mash-up of apples and oranges. *International Journal on Semantic Web & Information Systems*, 3(1), 2007. 3.1, 4

T. Gruber et al. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43(5):907–928, 1995. 2.4.1

T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 1993. 1.4, 2.4.1

N. Guarino, C. Masolo, and G. Vetere. Ontoseek: Content-based access to the web. *Intelligent Systems and their Applications, IEEE*, 14(3):70–80, 2002. ISSN 1094-7167. 1.4

J. Gulla and V. Sugumaran. An interactive ontology learning workbench for non-experts. In *Proceeding of the 2nd international workshop on Ontologies and information systems for the semantic web*, pages 9–16. ACM, 2008. 3.3.6

J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006. ISBN 1558609016. 3.3.4

M. Hepp, D. Bachlechner, and K. Siorpaes. Ontowiki: community-driven ontology engineering and ontology usage based on wikis. In *2006 international symposium on Wikis*, pages 143–144. ACM, 2006. 3.4.1

P. Heymann and H. Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical report, InfoLab, Stanford, 2006. 3.3, 3.3.1, 3.3.4, 5.5.2

T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. 1998. 3.3.1

A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. *LNCS*, 4011:411, 2006. 1.3

J. Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6):1–4, 2006. 1.5, 3.4, 4.2.3

J. Hu, G. Wang, F. Lochovsky, J.-t. Sun, and Z. Chen. Understanding user's query intent with wikipedia. In *the 18th international conference on World wide web*, pages 471–480. ACM, 2009. 7.4.3

M. Huiskes and M. Lew. The MIR flickr retrieval evaluation. In *Proc. ACM Special Interest Group on Multimedia (SIGMM 08), ACM Press*, page 3943, 2008. 5.7.2, 5.7.2.3

M. Hwang, H. Kong, and P. Kim. The design of the ontology retrieval system on the web. In *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, volume 3, pages 1815–1818. IEEE, 2006. ISBN 8955191294. 3.5

E. Jacob. Classification and categorization: A difference that makes a difference. *Library Trends*, 52(3):515–540, 2004. ISSN 0024-2594. 1.1

R. Jasper and M. Uschold. A framework for understanding and classifying ontology applications. In *Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW*, volume 99, pages 16–21. Citeseer, 1999. 5.7.1.1

D. Kirkpatrick. Help wanted: Adults on facebook. *fortune*, 2008. 3.4.4

A. Kittur, E. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 453–456. ACM, 2008. 1.5, 3.4.2

H. Kong, M. Hwang, and P. Kim. A new methodology for merging the heterogeneous domain ontologies based on the wordnet. 2005. 3.3.5

A. Kordon. Artificial vs. computational intelligence. *Applying Computational Intelligence*, pages 3–29, 2010. 4.2.2, 4.2.3

A. Kosorukoff. Human based genetic algorithm. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 5, pages 3464–3469. IEEE, 2001. 4.2.3

A. Kosorukoff and D. Goldberg. Evolutionary computation as a form of organization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002) pp*, pages 965–972, 2002. 4.2.3

D. Laniado, D. Eynard, and M. Colombetti. Using WordNet to turn a folksonomy into a hierarchy of concepts. In *Proc.of 4th Italian Semantic Web Workshop*, 2007. 3.3.5

F. Limpens, F. Gandon, and M. Buffa. Collaborative semantic structuring of folksonomies. pages 132–135. IEEE Computer Society, 2009. 3.4.4

H. Lin, J. Davis, and Y. Zhou. An integrated approach to extracting ontological structures from folksonomies. In *6th European Semantic Web Conference (ESWC)*, pages 654–668. Springer-Verlag, 2009. 1.3

G. Little, L. Chilton, M. Goldman, and R. Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 68–76. ACM, 2010. 3.4.2, 4.6.5

B. Liu, W. Hsu, and Y. Ma. Mining association rules with multiple minimum supports. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 337–341. ACM, 1999. 8.6

B. Liu, Y. Ma, C. Wong, and P. Yu. Scoring the data using association rules. *Applied intelligence*, 18(2):119–135, 2003. 2, 5.3

T. Malone, R. Laubacher, and C. Dellarocas. Harnessing crowds: Mapping the genome of collective intelligence. *Report: CCI Working Paper*, 1, 2009. 4.2.3

C. Mangold. A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, 2(1):23–34, 2007. ISSN 1744-2621. 3.5

B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme. Evaluating similarity measures for emergent semantics of social tagging. In

Proceedings of the 18th international conference on World wide web, pages 641–650. ACM, 2009. 5.3

C. Marlow, M. Naaman, D. Boyd, and M. Davis. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40. ACM, 2006. ISBN 1595934170. 3.3.2

A. Mathes. Folksonomies-cooperative classification and communication through shared metadata. *Computer Mediated Communication*, 2004. Dec. 3.2

J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine intelligence*, 4(463-502):288, 1969. 4.2.1

P. McCorduck and I. Ebrary. *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*. AK Peters, 2004. 4.2.1

P. Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1): 5–15, 2007. 3.3.2

G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995. 1.6, 2.4.3

F. Monaghan and D. Sullivan. Automating photo annotation using services and ontologies. 2006. ISSN 1551-6245. 3.3.2, 3.5

S. Movva, R. Ramach, X. Li, P. Cherukuri, and S. Graves. Noesis: A Semantic Search Engine and Resource Aggregator for Atmospheric Science. 2007. 3.5

T. Nelson. Complex information processing: a file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th national conference*, pages 84–100. ACM, 1965. 2.2.1

M. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004. 3.3.2

M. Newman and M. Girvan. Finding and evaluating community structure in networks. *PHYSICAL REVIEW E Phys Rev E*, 69:026113, 2004. 8.6

M. Niepert, C. Buckner, and C. Allen. Working the crowd: Design principles and early lessons from the social-semantic web. In *Proceedings of the Workshop on Web*, volume 3, 2009. 3.4

N. F. Noy and D. L. Mcguinness. Ontology development 101: A guide to creating your first ontology. Technical report, 2001. 2.4.4

N. F. Noy and M. A. Musen. Algorithm and tool for automated ontology merging and alignment. In *Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, 2000. 2.4.5

N. F. Noy, A. Chugh, W. Liu, and M. A. Musen. A framework for ontology evolution in collaborative environments. *The Semantic Web-ISWC*, pages 544–558, 2006. 3

J. Z. Pan, S. Taylor, and E. Thomas. Reducing ambiguity in tagging systems with folksonomy search expansion. In *6th European Semantic Web Conference*, volume 2009, 2009. 3.3.5, 5.7.2.2

N. Parikh and N. Sundaresan. Inferring semantic query relations from collective user behavior. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 349–358. ACM, 2008. 3.5

J. Patrick, Y. Wang, and P. Budd. An automated system for conversion of clinical notes into snomed clinical terminology. In *Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68*, pages 219–226. Australian Computer Society, Inc., 2007. 3.3.5

C. Petrie. Plenty of room outside the firm. *IEEE Internet Computing*, pages 92–96, 2010. 8.5

A. Plangprasopchok and K. Lerman. Constructing folksonomies from user-specified relations on flickr. In *Proceedings of the 18th international conference on World wide web*, pages 781–790. ACM, 2009. 2.3.2

M. Plasse, N. Niang, G. Saporta, A. Villeminot, and L. Leblond. Combined use of association rules mining and clustering methods to find relevant links between binary rare attributes in a large data set. *Computational Statistics & Data Analysis*, 52(1):596–613, 2007. ISSN 0167-9473. 3.3.3, 2, 5.3

X. Qi and B. Davison. Web page classification: Features and algorithms. *ACM Computing Surveys (CSUR)*, 41(2):1–31, 2009. ISSN 0360-0300. 2.3.1

E. Quintarelli. Folksonomies: power to the people. 2005. 1.1, 2.5, 2.5

G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc. New York, NY, USA, 1986. 1.1, 5.3

C. Schmitz, A. Hotho, R. J

"aschke, and G. Stumme. Mining association rules in folksonomies. In *the 10th IFCS Conference, Studies in Classification, Data Analysis, and Knowledge Organization*, 2006. 3.3, 3.3.3, 2, 5.3

P. Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop (WWW'06)*, Edinburgh,UK, 2006. 3.3.1, 4

A. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga. Ontology-based photo annotation. *IEEE Intelligent Systems,*, pages 66–74, 2001. ISSN 1541-1672. 3.3.6

E. Schwarzkopf, D. Heckmann, D. Dengler, and A. Kroner. Mining the structure of tag spaces for user modeling. In *Data Mining for User Modeling On-line Proceedings of Workshop*, page 63, 2007. 3.3.3

C. Selvi and A. Tamilarasi. Mining association rules with dynamic and collective support thresholds. *International Journal of Engineering and Technology*, 1(3):236–240, 2009. 8.6

H. Simon. *The sciences of the artificial.* the MIT Press, 1996. 4.3.1

E. Simperl, M. Mochol, T. B

"urger, and I. Popov. Achieving Maturity: The State of Practice in Ontology Engineering in 2009. pages 983–991. Springer, 2009. 2.4.2

K. Siorpaes. Lightweight community-driven ontology evolution. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, pages 951–955. Springer-Verlag, 2007. ISBN 3540762973. 3.4.1

K. Siorpaes and M. Hepp. OntoGame: weaving the semantic web by online games. *The Semantic Web: Research and Applications*, pages 751–766, 2008. 3.4, 3.4.3, 4.2.3

R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008. 3.4.2

A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008. 3.4.2, 6.1

L. Specia and E. Motta. Integrating folksonomies with the semantic web. In *European Semantic Web Conference*, Innsbruck, Austria, 2007. 3.3.5, 3.3.6, 5.7.2.2

L. Stojanovic, A. Maedche, N. Stojanovic, and R. Studer. Ontology evolution as

reconfiguration-design problem solving. In *the 2nd international conference on Knowledge capture*, page 171. ACM, 2007. 4.3

H. Stuckenschmidt and F. van Harmelen. *Information Sharing on the Semantic Web*. Springer, Heidelberg, 2005. 3.2, 3.3.3, 3, 5.7.2.3

F. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217, 2008a. ISSN 1570-8268. 2.4.3

F. M. Suchanek, M. Vojnovic, and D. Gunawardena. Social tags: meaning and suggestions. 2008b. 2.3.4, 3.3.5

C. Sunstein. *Infotopia: how many minds produce knowledge*. Oxford University Press, USA, 2006. 7.2

C. Thomas and A. Sheth. Semantic convergence of wikipedia articles. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 600–606. IEEE Computer Society, 2007. ISBN 0769530265. 3.4.1

C. Torniai, J. Jovanovic, S. Bateman, et al. Leveraging folksonomies for ontology evolution in e-learning environments. In *The IEEE International Conference on Semantic Computing*, pages 206–213. IEEE, 2008. 3.2

D. Vallet, M. Fernández, and P. Castells. An ontology-based information retrieval model. *The Semantic Web: Research and Applications*, pages 455–470, 2005. 1.4, 1.5

C. Van Damme, M. Hepp, and K. Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. *Bridging the Gap between Semantic Web and Web*, 2:57–70, 2007. 3.1, 3.3.6

T. Vander Wal. Folksonomy coinage and definition. *http://www.vanderwal.net/folksonomy.html*, 2007. 1, 1.2, 2.3.2

D. Vizine-Goetz, C. Hickey, A. Houghton, and R. Thompson. Vocabulary mapping for terminology services. *Journal of digital information*, 4(4), 2006. 3.3.5

L. Von Ahn. Human computation. In *the 4th international conference on Knowledge capture*, pages 5–6. ACM, 2007. 3.4, 3.4.3, 3.4.3

L. Von Ahn, R. Liu, and M. Blum. Peekaboom: A game for locating objects in images. In *the SIGCHI conference on Human Factors in computing systems*, 2006. 3.4, 3.4.3, 4.2.3

L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321 (5895):1465–1465, 2008. 3.4, 3.4.4, 4.2.3

D. Waltz. Artificial intelligence: realizing the ultimate promises of computing. *AI magazine*, 18(3):49, 1997. 4.2.1

B. Wang, B. Hou, Y. Yao, and L. Yan. Human flesh search model incorporating

network expansion and gossip with feedback. In *2009 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, pages 82–88. IEEE, 2009. 3.4.3

S. Wasserman and K. Faust. *Social network analysis: methods and applications*, volume 8. Cambridge Univ Pr, 1994. 3.3.2

D. Woods. Steps to creating a flash company. *Forbes Magazine*, 2010. 8.5

H. Wu, M. Zubair, and K. Maly. Harvesting social knowledge from folksonomies. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 111–114, Odense, Denmark, 2006a. ACM. ISBN 1-59593-417-0. doi: 10.1145/1149941.1149962. 3.3, 3.3.4

X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *Proceedings of the 15th international conference on World Wide Web*, pages 417–426, Edinburgh, Scotland, 2006b. ACM. ISBN 1-59593-323-9. doi: 10.1145/1135777.1135839. 1.5, 3.3.1

Y. Yang, B. Zhu, R. Guo, L. Yang, S. Li, and N. Yu. A comprehensive human computation framework: with application to image labeling. In *Proceeding of the 16th ACM international conference on Multimedia*, pages 479–488. ACM, 2008. 3.4.4

C. Yeung, N. Gibbins, and N. Shadbolt. Tag meaning disambiguation through analysis of tripartite structure of folksonomies. In *Proceedings of the 2007*

IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops, pages 3–6. IEEE Computer Society, 2007. ISBN 0769530281. 3.3.2

N. Zhong, J. Liu, Y. Yao, and S. Ohsuga. Web Intelligence (WI). In *Computer Software and Applications Conference, 2000. COMPSAC 2000. The 24th Annual International*, pages 469–470. IEEE, 2002. ISBN 0769507921. 1, 2.2.3