Ontology Learning from Folksonomies

CHEN, Wenhao

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong

August 2010

Thesis/Assessment Committee

Professor LYU Rung Tsong (Chair)

Professor LEUNG Ho Fung (Thesis Supervisor)

Professor FU Wai Chee (Committee Member)

Professor CHEUNG Shing Chi (External Examiner)

Abstract

This thesis deals with the problem of ontology learning. Ontology, specification of the objects, properties and relations that one would encounter in a particular domain of discourse, is the basis component of the Semantic Web. Since constructing ontologies is a time consuming job for domain experts, much research is conducted on automatically extracting ontologies from texts. With the development of folksonomy or collaborative tagging system, more and more researchers realize that folksonomy is a better knowledge source for constructing ontologies than texts. Although some works have already been proposed to extract ontologies from folksonomies, they consider little on what is a more acceptable and applicable ontology for users and lack an principle to supervise the ontology extraction from a human's perspective. In cognitive psychology, there is a family of concepts named basic level concepts which are frequently used by people in their daily life, and most human knowledge is organized with basic level concepts. In this thesis, inspired by studies in cognitive psychology, we try to extract ontologies with basic level concepts from folksonomies. To the best of our knowledge, it is the first work on discovering basic level concepts in folksonomies and using them to construct ontologies. Using Open Directory Project (ODP) as the benchmark, we demonstrate that the ontology extracted by our method is reasonable and consistent with human thinking. In addition, we also discuss the impact of context in ontology learning. In cognitive psychology, context plays an important role in human cognitive process including basic level concepts detection. The basic level concepts in the same domain are different under different contexts. We demonstrate the existence of context effect on categorization and concept learning in folksonomies through different evaluation methods. The effectiveness of our method in modeling context is also discussed in this thesis. Our motivation is to model human cognitive process especially ontology learning process so that we can explore the implicit semantics in folksonomies.

摘要

本論文探討了關於本體學習的問題。本體是語義網的基本組成部分。由於構造本體對於領域專家來說是一項相當耗時的工程，目前的許多研究都致力於從文件中直接提取本體。隨著大眾分類法的發展，越來越多的研究者意識到對於本體學習來說大眾分類系統是一個比文本文件更好的知識庫。雖然已經有許多關於從大眾分類系統中提取本體的方法，他們基本沒有考慮什麼樣的本體才是一個更容易被用戶接受和重用的本體，同時他們缺少一種從人類思維方式出發的監督本體學習的原理。在認知心理學中，存在一系列被稱為基本概念的概念，他們被人們頻繁的應用於日常生活中，同時大部分人類的知識都可以用基本概念來組織。在本論文中，通過對認知心理學的學習，我們嘗試從大眾分類系統中提取由基本概念構成的本體。根據我們的了解，這是第一次有人嘗試從大眾分類系統中提取基本概念，並用它們來構造本體。我們通過使用開放式分類目錄搜索系統作為基本標準，證明了通過我們的方法構造的本體符合人們的預期。另外，我們也討論了語境對本體學習的影響。在認知心理學中，語境對人類的認知過程有及大的影響，特別是在基本概念的學習中。在同一個領域中，不同的語境，會產生不同的基本概念。通過不同的實驗，我們證明了語境對分類和概念學習的影響是確實存在的。

我們同樣也討論了我們方法在模擬語境上的有效性。我們研究的根本動機就是通過對人類認知過程的模擬特別是本體學習過程的模擬達到提取大眾分類系統中的語義並加以應用的目的。

Acknowledgments

Special thanks will be given to my supervisor Prof. Ho-Fung Leung and other members of my research group: Cai Yi, She Yingni and Sui Xin.

Contents

1 Introduction 1

1.1 Ontologies and Folksonomies 1

1.2 Motivation 3

1.2.1 Semantics in Folksonomies 3

1.2.2 Ontologies with basic level concepts 5

1.2.3 Context and Context Effect 6

1.3 Contributions 6

1.4 Structure of the Thesis 8

2 Background Study 10

2.1 Semantic Web 10

2.2 Ontology 12

2.3 Folksonomy 14

2.4 Cognitive Psychology 17

2.4.1 Category (Concept) 17

2.4.2 Basic Level Categories (Concepts) 17

2.4.3 Context and Context Effect 20

2.5 F1 Evaluation Metric 21

2.6 State of the Art 23

2.6.1 Ontology Learning 23

2.6.2 Semantics in Folksonomy 26

3 Ontology Learning from Folksonomies 28

3.1 Generating Ontologies with Basic Level Concepts from Folksonomies . . . 29

3.1.1 Modeling Instances and Concepts in Folksonomies . . . 29

3.1.2 The Metric of Basic Level Categories (Concepts) . . . 30

3.1.3 Basic Level Concepts Detection Algorithm . . . 31

3.1.4 Ontology Generation Algorithm . . . 34

3.2 Evaluation . . . 35

3.2.1 Data Set and Experiment Setup . . . 35

3.2.2 Quantitative Analysis . . . 36

3.2.3 Qualitative Analysis . . . 39

4 Context Effect on Ontology Learning from Folksonomies 43

4.1 Context-aware Basic Level Concepts Detection . . . 44

4.1.1 Modeling Context in Folksonomies . . . 44

4.1.2 Context Effect on Category Utility . . . 45

4.1.3 Context-aware Basic Level Concepts Detection Algorithm 46

4.2 Evaluation . . . 47

4.2.1 Data Set and Experiment Setup . . . 47

4.2.2 Result Analysis . . . 49

5 Potential Applications 54

5.1 Categorization of Web Resources . . . 54

5.2 Applications of Ontologies . . . 55

6 Conclusion and Future Work 57

6.1 Conclusion . . . 57

6.2 Future Work . . . 59

Bibliography 63

List of Figures

1.1 The Ontology Generated by Our Approach. 8

2.1 The Layered Structure of the SemanticWeb proposed by Berners-Lee 12

2.2 Definitions written in RDF [1] 13

2.3 An example of categorization or clustering [2] 21

3.1 An example of algorithm 1 34

3.2 The impact of threshold q. 37

3.3 F1-scores of the category structure built by different algorithms in the three data sets. 39

3.4 Percentage of Different Relations 40

4.1 Comparison of different methods without context information . 50

4.2 Result of our method with (out) context information 51

List of Tables

3.1 Statistics of the extracted ontologies 37

3.2 Similar Concepts between ODP and Ontology extracted by Our Method 41

4.1 Basic level concepts detected in different contexts 53

4.2 Evaluation of basic level concepts in C_{pl} and C_{os} 53

Chapter 1

Introduction

1.1 Ontologies and Folksonomies

Metadata, known as "data about data", is structured information of resources such as documents, books, articles, photographs or other items. It helps systems and users find relevant and useful information. The generation of metadata can be approached in three ways [3]: firstly, metadata is created by professionals traditionally. In libraries and other organizations, creating metadata has been the domain of professionals working with complex rule sets and vocabularies. Generally ontologies constructed by domain experts, describing a certain reality with specific vocabulary are considered as this type of metadata. An alternative approach is creating metadata by authors. Original creators of the resources provide the metadata. Both of these approaches have the same problem: the intended and unintended users of the information are disconnected from the process, and it is hard for them to use the metadata without special training. The third approach which appears in recent years is creating metadata by its users. In folksonomies, users create and manage tags to annotate web resources. These tags are considered as user created metadata. Folksonomies have many advantages over controlled vocabularies or formal taxonomy [4]. There are no complicated vocabularies need to be learned. Users simply create and apply tags freely. In addition, folksonomies are inherently open-ended and therefore respond quickly to changes and innovations in the way users categorize and describe resources. Al-Khalifa et al. [5] demonstrated that folksonomy tags agree more closely with human thinking than those automatically extracted from texts.

The advantages of folksonomies are as follows. There are no complicated, hierarchically organized vocabularies need to be learned in folksonomies. Users simply create and apply tags freely. Folksonomies directly reflect the vocabulary of users which can be used in further study of the community. It can also help you find the users with the same interesting and useful resources. Browsing the system and its interlinked related tag sets is wonderful for finding things unexpected in a general area. However, the problems inherent in an uncontrolled vocabulary lead to a number of limitations and weaknesses in folksonomies, such as ambiguity, synonyms and noise tags. The biggest problem of folksonomies is that there is no hierarchy, and no directly specified parent-child or sibling relationships between tags. It cannot be used for machines and implementation of knowledge representation systems.

On the other hand, with the development of semantic web, ontology plays an important role in providing a way to give semantics to web resource. Ontologies with a hierarchical structure which is similar to a taxonomy are the basis and enabling technology of semantic web, for information sharing and manipulation. However, as we know, the weaknesses of ontologies are that the data users are disconnected from the design of ontology, and it is hard for them to use it without special training. Extracting ontologies from folksonomies is a way to combine the advantages of ontologies and folksonomies. These ontologies represent most users' latest opinions about how to describe a web resource. These ontologies will benefit both social tagging systems and the development of semantic web.

Some researches have been already conducted on automatically extracting ontologies from folksonomies. For examples, Mika [6] extract broader/narrower tag relations using set theory. Zhou et al. [7] apply Deterministic Annealing for clustering tags and build tags hierarchical structure. However, these works focus on hierarchy construction only and they consider little on what is a better ontology for users. As an ontology provides a way to model a domain of human knowledge, it is necessary to take people's thinking and cognitive process into consideration.

1.2 Motivation

1.2.1 Semantics in Folksonomies

After inspecting the tags in folksonomies, people can find that tags have a lot of information about the resource. Actually, there is an assumption that user defines a tag to annotate the resources based on some special purpose. User will use tags which they think are important to identify the resource for themselves. Consequently, different types of tags can be identified depending on its purpose [8]:

- Identify "what or who it is about". These tags are used to identify what the content is or who the content is about.
- Identify "what it is". These tags indicate the type of the annotated resource such as blog, book, etc.
- Identify "who owns it". These tags are used to establish who is the author or the proprietary of the content.
- Identify "categories". Some users use particular tags to simulate hierarchies such as some numbers.
- Identify "Qualities and characteristics of the content". These tags are usually adjectives (funny, bored, etc.) representing the opinion of the user who annotates the content.

- Self reference. Such tags represent the relationship between the user and the content. Usually they begin with "my". For example, mythings, myjob, mycomments, etc.
- Organize "tasks". Such tags are used to simulate content classification in order to organize the work. Tags that fit into this class are toread, todo, search-work, etc.

Through the discovery of different purposes or semantic of tags, there is an assumption about the frequency of different tags associated with the same resource. If the frequency of a tag is higher, which means more people have some purpose to use the tag and think the tag is useful and important to identify the resource, the tag is more important to the resource in the community. Most types of tags can be considered as identifying different features of the resource for the community. Other types of tags (identify "categories", self reference and organize "tasks") appear rarely. They can be considered as noise in the features space of the resource. Actually, because of the uncontrolled vocabulary, there are many different kinds of noise tags. If we consider the tags associated with a resource as its features, we can use the frequency of tags to reduce the impact of the noise, because the frequency of the noise tags is very low. In addition, in the experiment on delicious.com data, Golder [8] demonstrated that the combined tags of many users' bookmarks gave rise to a stable pattern in which the proportions of each tag were nearly fixed (after about 100 bookmarks). The reason is that the number of ideas or the features of the web page that are represented through tags and the importance of the feature is stable. The stable proportion also demonstrates that the community's linguistic custom is stable where the proportion of users used different synonyms to tag the resource is approximately fixed.

In addition, Al-Khalifa [5] constructed a system to automatically compare the overlap between folksonomy tags, Yahoo TE [9]which is a famous keyword extraction tool and human indexer keywords. The result of their experiments shows that folksonomy tags agree more closely with the human generated words than those automatically generated. Folksonomy tags have more semantics, and then are considered as a potential source for generating semantic metadata for web resources.

In conclusion, we find that folksonomies, including implicit semantics, is a potential knowledge source. How to extract the implicit semantics and make use of them deserve further research. Our objective is extracting the semantics in folksonomies and use them to build ontologies.

1.2.2 Ontologies with basic level concepts

Although some results have already been reported on generating ontologies from folksonomies, most of them do not consider what a more acceptable and applicable ontology for users should be. Previous research on ontology generation from folksonomies focused on hierarchy construction of tags and lacked a principle for supervising the process from a human's perspective [6]. Since an ontology provides a vocabulary shared by users to model a domain, it is necessary to construct ontologies from users' perspective (i.e., taking how people define and use concepts into consideration). In cognitive psychology, psychologists find that there is a family of categories named basic level categories. People can identify category members faster and easier in basic level categories, and such a level most faithfully mirrors natural kinds [10]. These categories represent the most natural level; neither too general nor too specific. People most frequently prefer to use basic level concepts which is the abstraction of basic level categories in their daily life. For example, when people see a car, most people would call it as a "car", even though we also can call it as a "vehicle" or a "sedan". Thus, we consider that constructing an ontology with basic level concepts for a domain can be more acceptable and applicable for users (more consistent with human thinking and reused easily).

1.2.3 Context and Context Effect

Inspired by studies in cognitive psychology, we try to model human cognitive process in folksonomies. Context plays an important role in cognitive process of human especially in basic level concepts detection. The basic level concepts in the same domain are different in different contexts [11]. For example, for all computer science conferences, people may consider "data mining conferences", "semantic web conference", "graphics conferences" and so on as the basic level concepts in the context of submitting a paper. However, in the context of measuring a researcher's publications, the basic level concepts for all computer science conferences may be "rank one conferences", "rank two conferences" and so on. Hence, it is necessary to take context into consideration while detecting basic level concepts. In this thesis, our objective is to demonstrate the existence of context effect in human categorization process and basic level concepts detection process. We want to discuss the importance of taking context into consideration.

1.3 Contributions

This thesis presents our research work which investigates the problem of ontology learning, and proposes a novel idea to explore the implicit semantics in folksonomies and use them to build ontologies. Our work combines thorough background research, psychology analysis and experiments on real world data sets. We summarize the contributions of our research work as follows.

- We carry out a thorough study of folksonomies including the main components of folksonomies, the advantages of folksonomies and semantics in folksonomies. In addition, we also investigate current research in folksonomies.

- We investigate the nature of human cognitive process and concept learning process. We mainly study the research on basic level categories (concepts) and their implementation.

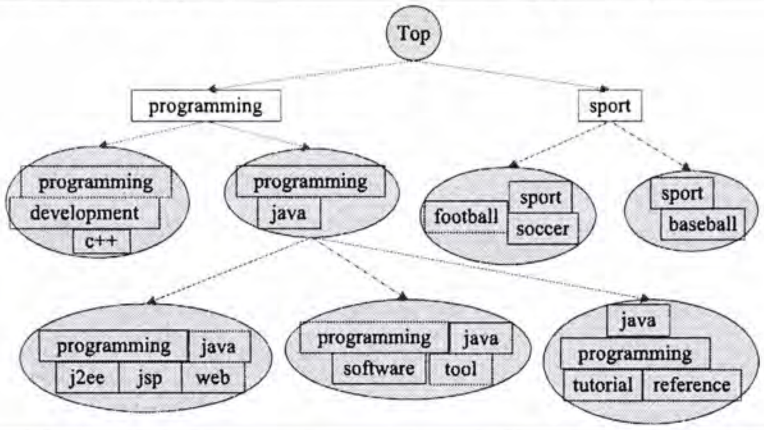
- We propose an algorithm for constructing ontology with basic level concepts. To the best of our knowledge, it is the first work on discovering basic level concepts in folksonomies and using them to construct ontologies. We conduct experiments to evaluate our method using del.icio.us data set and compare the extracted ontology with ODP concept hierarchy. Experiments show that the ontologies extracted using our method are more consistent with human thinking than that of other compared methods.

- We propose a novel basic level concepts detection algorithm to take context into consideration. We model context in folksonomies and demonstrate the existence of context on basic level concept detection. We also ask students and experts to evaluate the results we get from different context. The evaluation results justify the context effect on basic level concepts detection.

- We also discuss the metric to characterize basic level categories. The original metric is category utility. Based on category utility, we take the effect of folksonomies into consideration and give a modified category utility. Similarly, we also give a contextual category utility to consider context effect.

In conclusion, inspired by studies in cognitive psychology, we try to extract ontologies with basic level concepts from folksonomies. An algorithm for constructing ontology with basic level concepts is proposed. In addition, we also discuss the context and context effect on ontology learning. While previous methods represent a concept in an ontology by only one tag, our method provides a novel way to represent a concept by a set of tags. Figure 1.1 gives an example of the ontology explored through our approach. In the ontology, concepts are represented by the common tags of a category of resources. The tags of a concept are inherited by its sub-concepts and a concept has all instances of its descendants. Such a representation can keep more information and properties of concepts. We expect that this work can benefit the future development of ontology learning and folksonomies, and can be used to enhance knowledge representation in the Semantic Web. We hope that our work can invoke future research in combining cognitive psychology and data mining technology.

Figure 1.1 The Ontology Generated by Our Approach.



1.4 Structure of the Thesis

This thesis is structured as follows:

Following this introductory section, Chapter 2 reviews the basic knowledge of the topics involved in this thesis firstly. These include the background of the Semantic Web, ontologies, folksonomy and related concepts in cognitive psychology. Then the chapter also mentions previous research on the topic of ontology learning and semantics in folksonomy.

Chapter 3 describes the details of the approach of ontology extraction from folksonomies. We start from the basic ideas in modeling of instances, concepts and properties, and then go on to describe the metric of basic level categories. Experimental results are given out in this chapter. Through quantitative analysis and qualitative analysis, we demonstrate the effectiveness of our method in generating ontologies from folksonomies. In the experimental part, we use Open Directory Project (ODP) [1] as the gold standard.

Chapter 4 mainly discusses the context effect on our method. We investigate the problem of context and the contextualization of ontologies and model context in folksonomies. We also describe how different contexts constitute different results. A new approach of basic level concepts learning taking context into consideration is presented in this chapter. In the experiment part, we mainly use questionnaires (asking students and experts to evaluate the results) to discuss the existence of context in ontology learing and its effect. Experiments demonstrate the importance of taking context into account and effectiveness of our method.

Finally, Chapter 5 discusses the potential applications of the approach discussed in this thesis. This approach can be used in categorization of web resources and benefit the development of semantic web. Chapter 6 draws conclusions, highlights the main research issues and major contributions of this research work. In addition, we present some future research directions.

[1] http://www.dmoz.org/

Chapter 2

Background Study

2.1 Semantic Web

The Semantic Web is an extension of the World Wide Web which derives from W3C director Tim Berners-Lee's vision of the Web as a universal medium for data, information and knowledge exchange where web resources are made not only for humans to read but also for machines to understand and automatically process [12]. Through using technical standards and ontological markup languages to describe semantics of a certain web resource, resources in the Semantic Web are machine-readable.

Currently, web pages are mainly marked up by HTML (Hypertext Markup Language). HTML is limited in describing the content of a document. Unless using advanced natural language processing algorithms [13], the semantics of the web pages cannot be understood without human inspection. Hence, it is difficult to let agents such as search engines process the documents and extra useful information for users.

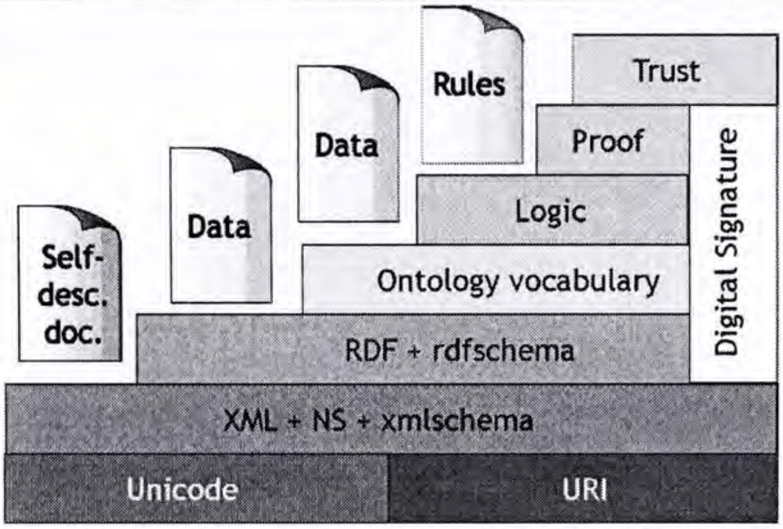
The Semantic Web addresses the problem by using ontologies [14] which are specified in descriptive languages such as RDF (Resource Description Framework) and OWL (Web Ontology Language)[15]. These descriptive languages are based on the customizable markup language XML (eXtensible Markup Language). The standardized machine readable descriptions allow content managers to add meaning to the content, thereby facilitating automatic information gathering and research.

Figure 2.1 shows the architecture of the Semantic Web proposed by Berners-Lee.[1] The layers refer to different components of the Semantic Web. In the Semantic Web, each resource is given an URI (Uniform Resource Identifier, a compact string of characters used to identify or name a resource). The URL of a web site (e.g. http://www.semanticfocus.com) is a popular example of a URI. URI and Unicode consist of the bottom level of the architecture. Unicode is the universal standard encoding system and provides a unified system for representing textual data. On the top of bottom level, we find XML which allows users to define their own vocabulary, and RDF which allows users to specify relations between resources. As we go up the layer, there are more expressive and powerful ontology languages, and also a logic framework which provides reasoning services on the concepts and properties defined in ontologies. Finally the trust layer implements components, such as digital signature, which is used to ensure security and quality.

Although the Semantic Web facilitates automatic information gathering and research, it faces many different challenges. Existing technology has not yet been able to eliminate all semantically terms. There are many logical contradictions which will arise during the development of large ontologies, and when ontologies from separate sources are combined. In addition, the producer of the information sometimes is intentionally misleading the user of the information. There is also a challenge from imprecise concepts such as "young" or "tall". These concepts impede the process of matching query terms of users to provider terms and trying to combine different knowledge bases with overlapping. This challenge is considered as vagueness and the most common technique for dealing with vagueness is fuzzy logic.

[1] http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html.

Figure 2.1 The Layered Structure of the SemanticWeb proposed by Berners-Lee



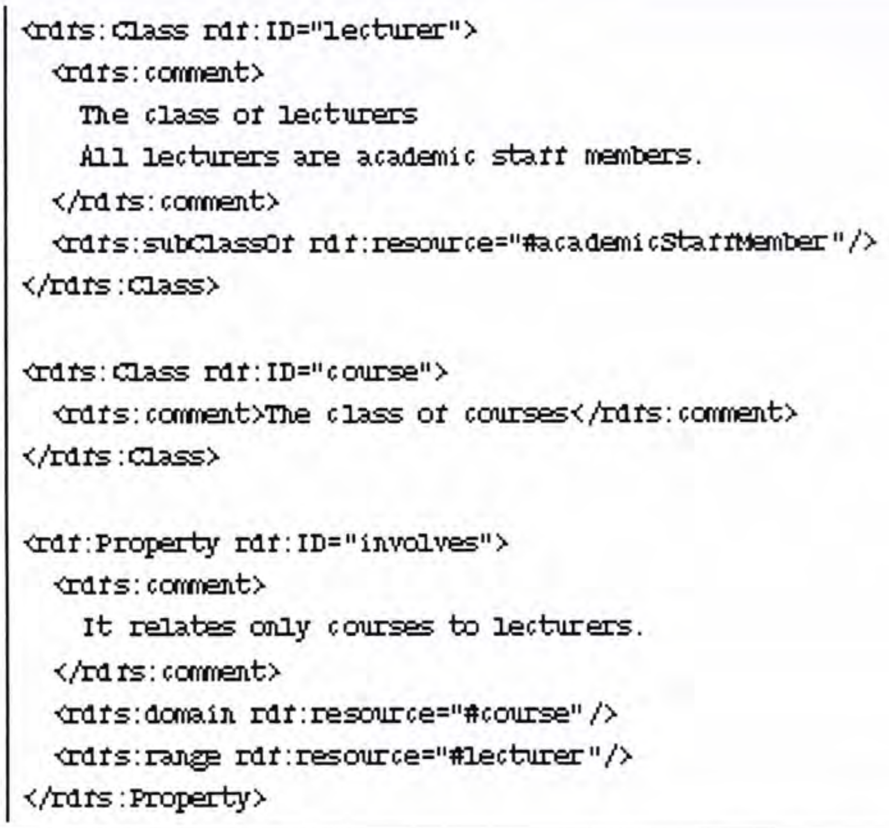
Since this thesis concerns the problem of ontology learning from folksonomies, we mainly deal with the ontology layer of the Semantic Web. More about knowledge representation in the Semantic Web and ontologies will be presented in the next section.

2.2 Ontology

"Ontology" is originally a philosophical term, a major and fundamental branch of metaphysics, which studies the problem of being or existence and their basic categorizations and relationships [15]. The term "ontology" has been adopted into computer science, especially by researchers in artificial intelligence and knowledge management, to refer to the specification of the objects, properties and relations that one would encounter in a particular domain of discourse [12].

Considering description logic as the theoretical support of logical reasoning services provided in ontologies, an ontology in computer science is defined as an explicit and formal specification of conceptualization. An ontology consists of a hierarchical taxonomy of concepts. The hierarchy is indeed a taxonomic (subclass) hierarchy [15]. In other words, if concept A is a subclass of concept

Figure 2.2 Definitions written in RDF [1]



```
<rdfs:Class rdf:ID="lecturer">
  <rdfs:comment>
    The class of lecturers
    All lecturers are academic staff members.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#academicStaffMember"/>
</rdfs:Class>

<rdfs:Class rdf:ID="course">
  <rdfs:comment>The class of courses</rdfs:comment>
</rdfs:Class>

<rdf:Property rdf:ID="involves">
  <rdfs:comment>
    It relates only courses to lecturers.
  </rdfs:comment>
  <rdfs:domain rdf:resource="#course"/>
  <rdfs:range rdf:resource="#lecturer"/>
</rdfs:Property>
```

B, every instances of A must be an instance of B and every property statement holds for instances of B must also apply to instances of A. Throughout the history of the development of ontologies, there have been quite a number of definitions of ontology [16]. To facilitate the discussions in this thesis, referring to [17], we formally define an ontology as a four-tuple:

Definition 2.1 *An* **Ontology** *is a tuple $O = (C, P, I, S)$ where C, P and I are finite sets, whose elements are called concepts, properties and instances, respectively, and S is a set of rules, propositions or axioms that specify the relations among concepts, properties and instances.*

Ontologies can be used in the Semantic Web to provide semantics to resources making them machine-readable. Agents are then able to access resources and communicate with one another based on the shared specification of concepts.

In the Semantic Web, different markup languages, such as RDF and RDF Schema, DAML+OIL and OWL, are available for coding of ontologies [15]. RDF stands for Resource Description Framework. It is a recommendation of the W3C and is intended for describing resources on the World Wide Web with meta-data. RDF is based on the idea that every object is related to each other through a binary relation. For example, referring to figure 2.2 which shows an ontology adapted from [1], it includes a relation between a course and a lecturer.

For more details on ontology development, readers can refer to the thorough review paper [16]. In conclusion, ontology is an engineering artifact which describes a certain reality with specific vocabulary. Ideally, ontology is constructed by domain experts with markup language so as to make ontology as a acceptable vocabulary for machine and human users. However, it is a time consuming job for human to construct an ontology. Accordingly, some researches are conducted on automatically extracting ontologies from texts, which we will discuss in section 2.6.1.

2.3 Folksonomy

The term folksonomy is generally attributed to Smith Gene [18]. It is a portmanteau of the words folk and taxonomy, so a folksonomy is a user generated taxonomy. Recently, folksonomies have become more and more popular on the Web as part of social annotation systems such as social bookmarking (e.g., delicious.com)[2] and photograph annotation (e.g., flickr)[3] . A folksonomy is generally considered as a system of classification derived from the practice of collaboratively creating and managing tags to annotate and categorize resources. There are millions of users in these systems recently. According to

[2] http://delicious.com

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

[19], a folksonomy is defined as follows:

Definition 2.2 *A* **folksonomy** *is a tuple* $\mathbb{F} := (U, T, R, Y)$ *where* U, T *and* R *are finite sets, whose elements are called users, tags and resources, respectively, and* Y *is a ternary relation between them, i.e.* $Y \subseteq U \times T \times R$.

In the definition, users are typically described by their user ID, and tags may be arbitrary strings. The type of resources in a folksonomy depends on the social annotation system. In delicious.com, resources are web pages and in Flickr resources are pictures. As an example, in the social annotation web site delicious.com, when a user bookmarks a web page, he can use any word to annotate it. These words are named tags and this action is defined as a post. The formal definition of a post is as follows:

Definition 2.3 *A* **post** *is a triple* $(u, T_{u,r}, r)$ *with* $u \in U, r \in R$, *and a set* $T_{u,r} := \{t \in T | (u, t, r) \in Y\}$.

Actually, a folksonomy consists of a set of posts. In a post, a user u assigns some related tags to a resource r.

Folksonomies have many advantages. A social annotation system allows its user to search for the resources that the user has tagged based on his vocabulary. Because users with similar interests tend to have a shared vocabulary, tags created by one user may be useful to others. In addition, collaborative tagging systems assist navigation through providing dynamic hyperlinks among tags, documents and users that help overcome searches' limitations. For instance, navigation allows casual browsing and leads to serendipitous discoveries. Through tag-based navigation users can discover who created a given tag and see the other tags this person has created. In this way a folksonomy user can discover other users with similar interests and other useful resources.

Accordingly, it helps users in not only retrieving information but also socializing with others. Furthermore, when a user is tagging a resource, the tags for this resource from other users can be provided for references. As soon as users assign a tag to a resource, he can see the cluster of resources with the same tag. If that is not what he expected, the user can change the tag or add another tag.

However, the problems in an uncontrolled vocabulary lead to a number of limitations and weaknesses in folksonomies. Ambiguity of the tags can emerge as users apply the same tag in different ways. On the other hand, the lack of synonym control can lead to different tags being used for the same concept. Additionally, there are many noise tags which have no meaning in folksonomies, such as "todo".

An important aspect of folksonomies which is very different from ontologies is that they are comprised of terms in a flat namespace [18]: there is no hierarchy, and no directly specified parent-child or sibling relationships between these terms. This is unlike formal taxonomies and classification schemes where multiple kinds of explicit relationships between terms exist. These relationships include broader, narrower, as well as related terms. Accordingly, the metadata of folksonomy is hard for machines to use. If ontologies can be built based on folksonomies, these ontologies will represent most users' latest opinion. As a result, the metadata of folksonomies becomes useful for machines as the form of ontologies.

2.4 Cognitive Psychology

2.4.1 Category (Concept)

Concepts are abstract representation of objects. The general view of concept held among psychologists suggested that concepts are defined by singly necessary and jointly sufficient properties. This view is now generally referred to as the classical view [20]. The idea of this view can actually be traced back to the time of Aristotle's philosophically oriented studies of categories [21], which requires instances of concepts to meet a set of pre-defined conditions. The classical Aristotelian view claims that categories are discrete entities characterized by a set of properties which are shared by their members. In analytic philosophy, these properties are assumed to establish the conditions which are both necessary and sufficient conditions to capture meaning. For example, the truth or falsity of "Rachel is a wildebeest" is something that can be determined by referring to the definition: Does Rachel have all the properties listed in the definition - four legs, horns and so on?

2.4.2 Basic Level Categories (Concepts)

In cognitive psychology, in a hierarchical category structure such as a taxonomy of plant, there is one level named the basic level at which the categories are cognitively basic. The basic level categories, defined by Rosch et al. [10], carry the most information and are the most differentiated from one another. They are the categories easier than others to be learned and recalled by human as concepts. In psychology, generally speaking, a concept holds the common features of a category of instances and is the abstraction of that category. Basic level concepts are the abstraction of basic level categories. Objects are identified as belonging to basic level categories and recognized as basic level concepts faster than others. For example, in classifying life forms, basic level categories tend to be at the level of the genus (maple, dog etc.). When we see a tree, we could call it a "plant", a "maple" and a "sugar maple", but most people will identify it as "maple". The concept "maple" is a basic level concept.

To characterize basic level categories, psychologists give the metric named category utility [22]. Through many experiments, they demonstrate that the character of basic level categories is that they have the highest category utility. It provides a normative information-theoretic measure of the predictive advantage gained by the person who possesses knowledge of the given category structure over the person who does not possess this knowledge.

Category utility can be viewed as a function that rewards traditional virtues held in clustering generally: similarity of objects within the same category and dissimilarity of objects in different categories [23]. Category utility is a tradeoff between intra-class similarity and inter-class dissimilarity of objects, where objects are described by a set of properties. Intra-class similarity is reflected by conditional probabilities of the form $p(f_i|c_k)$ where f_i is a feature and c_k is a category. The larger this probability, the greater the proportion of category members sharing the property. Inter-class similarity is a function of $p(c_k|f_i)$. The larger this probability is, the fewer the objects in contrasting categories that share this value. These probabilities are dispositions of properties, and they can be combined to give an overall measure of partition quality, where a partition is a set of mutually-exclusive object categories, $c_1, c_2, ..., c_m$. The combination of intra-class and inter-class similarity is as follows:

$$\sum_{k=1}^{m} \sum_{i=1} p(f_i)p(f_i|c_k)p(c_k|f_i) \tag{2.1}$$

It is a tradeoff between intra-class similarity (through $p(f_i|c_k)))$ and inter-class dissimilarity (through $p(c_k|f_i)$) that is summed across all categories (k) and properties (i). According to Bayes rule ($p(f_i)p(c_k|f_i) = p(c_k)p(f_i|c_k)$), it can be changed to:

$$\sum_{k=1}^{m} p(c_k) \sum_{i=1} p(f_i|c_k)^2 \qquad (2.2)$$

In other words, $\sum_{i=1} p(f_i|c_k)^2$ is the expected number of propertie that can be correctly guessed for an arbitrary member of category c_k. This expectation assumes a guessing strategy that is probability matching, meaning that a property is guessed with a probability equal to its probability of occurring. Thus, it assumes that a property is guessed with probability $P(f_i|c_k)$ and that this guess is correct with the same probability. [4]

Finally, category utility is considered as the increase in the expected number of properties that can be correctly guessed ($P(c_k)\sum_{i=1} p(f_i|c_k)^2$) given a partition $c_1, ..., c_m$ over the expected number of correct guesses with no such knowledge ($\sum_{i=1} p(f_i)^2$). In addition, It can also be considered as the increase in the inter-class and intra-class similarity when people do the categorization. The formal definition of category utility is as follows:

$$cu(C,F) = \frac{1}{m} \sum_{k=1}^{m} p(c_k) \left[\sum_{i=1}^{n} p(f_i|c_k)^2 - \sum_{i=1}^{n} p(f_i)^2 \right] \qquad (2.3)$$

where C is the set of categories, F is the set of features, f_i is a feature, $p(f_i|c_k)$ is the probability that a member of category c_k has the feature f_i, $p(c_k)$ is the probability that an instance belongs to category c_k, $p(f_i)$ is the probability that an instance has feature f_i, n is the total number of features, m is the total number of categories. The denominator, m, is the number of categories in a partition. Averaging over categories allows comparison of different size

[4] Probability matching can be contrasted with probability maximizing. The latter strategy assumes the most frequently occurring f_i is always guessed. While this strategy may seem superior at a cursory level, it is not sensitive to the distribution of all properties and is not as desirable for heuristically ordering object partitions. Psychologist demonstrate probability matching are the best strategy in human categorization process [10].

partitions.

2.4.3 Context and Context Effect

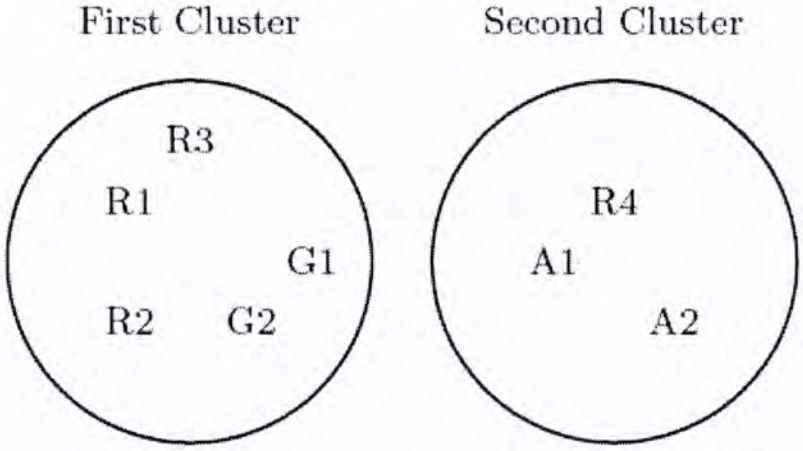
Context refers to the general conditions (circumstances) in which an event or action takes place. The context of something consists of the ideas, situations, judgments, and knowledge related to it. In cognitive psychology [24], the term "context effect" is used to refer to the influence of context in different cognitive tasks. For example, Roth and Shoben [25] investigate the effect of context in categorization, and suggest that, if the prototype view of concepts is applied, context causes a reweighing of the importance of the properties of a concept, thus resulting in a different categorization and concepts. In addition, Tanaka and Taylor [11] find out that the domain knowledge in different context has an effect on finding basic level concepts. The experts with particular domain knowledge tend to treat different concepts as basic level concepts compared with non-experts.

Elements of context can be classified into *internal context* and *external context* [26] [27]. Internal context refers to the subjective aspects of an agent (user). For example, in the categorization, the goal of using a concept and knowledge of the user are some subjective aspects of users. These aspects have a strong effect on forming perspectives to handle tasks. When a particular context is perceived by an agent, the agent forms a certain perspective. A perspective is a certain viewpoint on the concepts and objects encountered by the agent. It refers to a set of relevant aspects that one takes into consideration when accomplishing a particular task. The user will use such a perspective to handle a specific task. Thus, for different users, they may form different perspectives based on their subjective aspects for a particular task, and the results of handling the task may be different. For internal context, its effects on a task are achieved by applying perspectives to the task. External context refers to objective aspects in the environment, i.e., the ground facts (e.g., concepts and objects) that happen to exist in a situation. External context has an effect on a task because it can impose constraints for obtaining more relevant information (i.e., information of the task context) for the task. For different external contexts, their relevant information for the task is different.

There is numerous work on context in AI community. For instance, among all of these, McCarthy [28] is the first one to promote formalizing context in intelligent systems. He introduces the notation $ist(c, p)$ to denote the assertion that the proposition p is true in context c. [29] [30] and so on are subsequent efforts in formalizing context in logics. Guha et al. [31] present a context mechanism for the Semantic Web that is adequate to handle the data aggregation tasks. Besides, contexts are critical and useful in many other tasks [32].

2.5 F1 Evaluation Metric

Figure 2.3 An example of categorization or clustering [2]



In this thesis, to compare the generated ontology with the standard, we chose to use the F1 score as the evaluation metric [33]. F1 score is a measure of a categorization result's accuracy according to the standard. It is the harmonic mean of precision and recall. In this thesis, precision and recall are computed over pairs of resources or instances. F1 score is used to compare the category structure of ontologies.

An example is given in figure 2.3 [2]. Two categories are shown, and each resource is denoted by its category: A for "Arts", G for "Games", R for "Recreation". For example, A2 denotes a resource which is in the category "Arts" and the clustering algorithm has decided to put it in the second cluster. We think of pairs of resources as being either the same category or differing category (according to our standard), and we think of the clustering algorithm as predicting whether any given pair has the same or differing cluster. The clustering result shown in figure 2.3 has predicted that (A1,A2) are in the same cluster and that (R2,R4) are in different clusters. In figure 2.3, we find out that there are four cases of different resource pairs :

- **True Positives (TP)**: The clustering algorithm placed the two resources in the pair into the same cluster, and our standard has them in the same category. For example, (R1,R3). There are 5 true positives.
- **False Positives (FP)**: The clustering algorithm placed the two resources in the pair into the same cluster, but our standard has them in differing categories. For example, (R1,G2). There are 8 false positives.
- **True Negatives (TN)**: The clustering algorithm placed the two resources in the pair into differing clusters, and our standard has them in differing categories. For example, (R2,A1). There are 12 true negatives.
- **False Negatives (FN)**: The clustering algorithm placed the two resources in the pair into differing clusters, and our standard has them in the same category. For example, (R2,R4). There are 3 false negatives.

Then we can calculate precision, recall and F1 score. $Precision = \frac{TP}{TP+FP} = \frac{5}{13}$. $Recall = \frac{TP}{TP+FN} = \frac{5}{8}$. Precision can be considered as a measure of exactness or fidelity, whereas Recall is a measure of completeness. F1 score which is the harmonic mean of precision and recall takes advantages of precision and recall. $F1 = \frac{2 \times precision \times recall}{precision + recall} \approx 0.476$. F1 balances the need to place similar resources together while keeping dissimilar resources apart.

2.6 State of the Art

2.6.1 Ontology Learning

Ontology research is primarily concerned with the definition of concepts and relations between them [34]. As constructing ontologies by human is a time consuming and tough job, much research is conducted on ontology learning. Ontology learning also known as ontology extraction, ontology generation or ontology acquisition is a subtask of information extraction. The objective of ontology learning is to (semi-) automatically extract relevant concepts and relations from a given text or other kinds of data sets. Ontology learning actually contains six different aspects of learning tasks:

- **Terms**: Terms are linguistic realization of domain-specific concepts. Terms extraction is a prerequisite for all aspects of ontology learning. Previous research provides many examples of terms extraction methods that could be used as a first step in ontology learning from text. Most of these are based on information retrieval methods for term indexing [35]. Other methods take inspiration from terminology [36].

- **Synonyms**: The synonym learning addresses the acquisition of semantic term variants in and between languages, where the latter in fact concerns the acquisition of term translations. Much of the work in this area has focused on the integration of WordNet [5] for the acquisition of English synonyms [37]. In contrast to using available synonym sets, researchers

[5] WordNet is freely accessible from http://wordnet.princeton.edu

have also worked on algorithms for the dynamic acquisition of synonyms by clustering and related techniques [38]. There seems to be a current trend to use statistical information measures to detect synonyms [39].

- **Concepts**: The extraction of concepts from text is controversial as it is not clear what constitutes a concept. Most of the research in concept extraction addressed the question from a linguistic perspective, regarding concepts as clusters of related terms. Concepts learning includes the extraction or acquisition of formal and informal definitions. An informal definition might be a textual description. A formal definition includes the extraction of concept properties, part of which is the extraction of relations between a particular concept and other concepts.

- **Taxonomy**: Taxonomy is a hierarchical structure of concepts. The relationship between different level concepts is the is-a relation. There are currently three main paradigms exploited to induce taxonomies from textual data. The first one is the application of lexico-syntactic patterns to detect hyponymy relations as proposed by [40]. In the second paradigm, people mainly exploited hierarchical clustering algorithms to automatically derive term hierarchies from text, e.g. [41], The third paradigm stems from the information retrieval community and relies on a document-based notion of term subsumption, for example [42].

- **Relations (non-hierarchical)**: Relations extraction from text, other than the is-a relation discussed above, has been addressed primarily within the biomedical field as there are very large text collections available for this area of research. The goal of this work is to discover new relationships between known concepts by analyzing large quantities of biomedical scientific articles [43].

- **Rules**: The extraction of rules is probably the least addressed research area in ontology learning. Initial blueprints for this task can be found in [44].

The main component of ontology learning is the taxonomy part. As the reason that in the second paradigm of taxonomy learning, people mainly exploited hierarchical clustering algorithms to automatically derive term hierarchies from text, hierarchical clustering algorithm plays an important role in ontology learning which is the main focus of this thesis. In general, hierarchical clustering algorithm can be further classified into agglomerative and divisive hierarchical clustering approaches, depending on whether the hierarchy is formed in a bottom-up or top-down manner. The agglomerative approach, such as AGNES [45], is a bottom-up method. It begins with each object forming a separate group and then merges the most similar groups, until all of the groups are merged into one, or a termination condition holds, e.g. the similarity between the most similar groups is lower than a threshold. There are many different methods to compute the similarity between groups for example single linkage, centroid, complete linkage, etc. The divisive approach, such as DIANA [46], is a top-down approach. It starts with all objects in the same cluster and then splits a cluster into smaller clusters in each iteration until termination condition holds, e.g. K clusters remain.

In addition, Fisher [23] presented an incremental conceptual clustering algorithm, COBWEB, which creates a hierarchical structure in the form of classification tree through maximizing an evaluation measure called category utility in every incremental step. In every incremental step, the algorithm adds an instance or objects into the classification tree. There are four basic operations COBWEB employs in building the classification tree depending on the category utility of the classification achieved by applying it. The operations include merging two nodes, splitting a node, inserting a new node and passing an object down the hierarchy (a node is a cluster of objects). This method is extended to CLASSIT, by Gennari et al. [47], which is used for incremental clustering of continuous data.

Considering folksonomies as another source for ontology learning, next section will introduce previous research on folksonomies and discuss the implicit semantics in folksonomies.

2.6.2 Semantics in Folksonomy

In these years, folksonomy or social annotation becomes a hot topic, on which much research has been conducted. There are many research areas to enhance the capability of folksonomy, such as community identification, user and document recommendation, ontology learning and so on.

Community identification means to find the interests of people or which community one user belongs to. Diederich [48] used some tags related to a user to build the user's profile and feed them to a recommendation system, especially to identify related persons in the community. Wu [4] presented a spectral method to identify global communities using authorship and usage of tags and documents. All documents, tags and users are considered as nodes in a network. A link is added from each tag to every associated document. A link is also added from each user to every tag the user has created or accessed, and the documents accessed through the tag.

The ability to find high-quality resources, whether documents or people, is important to overcoming information overload. Recommendation systems identifying high quality resources and related users based on individual's knowledge are very useful. Hotho etc [49] gave out an algorithm called FolkRank based on the PageRank algorithm [50] to retrieve topically related items for any given set of highlighted tags, users or resources. Abbasi [51] presented a system T-ORG, which provides a mechanism to organize web resources by classifying the tags attached to them into predefined categories.

Making use of semantics under the tag space of folksonomies is an important research topic. Ramage et al. [2] compared the clustering results of using traditional words extracted from the text and using folksonomy tags. Their experiments demonstrated that using folksonomy tags can improve the clustering result. Au Yeung et al. [52] developed an effective method to disambiguate tags by studying the tripartite structure of folksonomies. He also proposed a k-nearest-neighbor method [53] for classifying web search results based on the data in folksonomies. Moreover, some researches focus on combining ontologies and folksonomies. Specia et al. [54] presented an approach for making explicit the semantics and hierarchy behind the tag space through mapping folksonomies to existent ontologies so that this collaborative organization can emerge in the form of groups of concepts and partial ontologies. Mika [6] extracted broader/narrower tag relations using set theory and proposed an approach to extend the traditional bipartite model of ontologies with the social annotations. Jaschke et al. [55] [49] [56] defined a new data mining task, the mining of frequent tri-concepts, and presented an efficient algorithm to discover these implicit shared conceptualizations. Zhou et al. [7] proposed a method to build the hierarchical structure of tags in a top-down way using Deterministic Annealing algorithm.

Chapter 3

Ontology Learning from Folksonomies

In this chapter, we discuss how to generate ontologies with basic level concepts from folksonomies. To the best of our knowledge, it is the first work on discovering basic level concepts from folksonomies and using them to construct ontologies [57]. We perform experiments to evaluate our method using delicious.com data set and compare the generated ontology with ODP concept hierarchy. Experiments show that ontologies generated using our method are more consistent with human thinking than that of other compared methods. In our approach, concepts are represented by the common tags of a category of resources. For example, tags "java" and "programming" together represents a concept about java programming. The tags of a concept are inherited by its sub-concepts and a concept has all instances of its descendants. Such a representation can keep more information and properties of concepts and is consistent with the definition of concepts in psychology.

3.1 Generating Ontologies with Basic Level Concepts from Folksonomies

3.1.1 Modeling Instances and Concepts in Folksonomies

In folksonomies, tags are given by users to annotate a resource and describe its characters. Naturally, the tagged resources are considered as instances in the definition of ontology. For the reason that each resource is described and represented by tags, we consider these tags as properties of instances. Accordingly, an instance is represented as a vector of tag-value pairs:

Definition 3.1 *An* **instance**, *r_i, is represented by a vector of tag:value pairs, $r_i = (t_{i,1} : v_{i,1}, t_{i,2} : v_{i,2}, \ldots, t_{i,n} : v_{i,n})$ with $t_{i,k} \in T, 0 < v_{i,k} \leq 1, 1 \leq k \leq n$.*

where n is the number of unique tags assigned to resource r_i, $v_{i,k}$ is the weight of tag $t_{i,k}$ in resource r_i. The weight $v_{i,k}$ determines the importance of the tag $t_{i,k}$ to resource r_i. We consider that a tag assigned by more users to a resource is more important because more users think the tag is useful to describe the resource. Although different users may annotate a resource in different aspects and some may even randomly assign tags, Golder [8] demonstrated that, in delicious.com, in a resource the occurrence frequency of a tag becomes a nearly fixed number after enough bookmark. The fixed number reflects the importance of a tag in the resource. Accordingly, the weight of a tag $t_{i,k}$ is defined as:

$$v_{i,k} = \frac{N_{t_{i,k}}}{N_{r_i}} \tag{3.1}$$

where $N_{t_{i,k}}$ is the number of users using the tag $t_{i,k}$ to annotate the resource r_i and N_{r_i} is the total number of users assigning tags to r_i. In the case that all users annotate r_i with $t_{i,k}$, the weight $v_{i,k}$ is 1.

A concept is the abstraction of a category of instances and holds the common properties of them. Accordingly, we construct a concept through extracting common tags of a category of instances. These common tags are considered as the properties of the concept. The weights of these tags are their mean values among all instances in a category. Accordingly, the definition of a concept is as follows:

Definition 3.2 *A* **concept**, c_i, *is represented by a vector of tag:value pairs,* $c_i = (t_{i,1} : v_{i,1}, t_{i,2} : v_{i,2}, \ldots, t_{i,n} : v_{i,n})$ *with* $t_{i,k} \in T, 0 < v_{i,k} \leq 1, 1 \leq k \leq n$.

where n is the number of unique tags, $t_{i,k}$ is a common tag of a category of resources, $v_{i,k}$ is the weight of the tag $t_{i,k}$.

3.1.2 The Metric of Basic Level Categories (Concepts)

To characterize basic level categories, psychologists [22] give a metric named category utility. Through many experiments, they demonstrate that the character of basic level categories is that they have the highest category utility. Category utility was intended to supersede more limited measures of category goodness such as cue validity. It provides a normative information-theoretic measure of the predictive advantage gained by a person who possesses knowledge of the given category structure over a person who does not possess this knowledge. Given a set C of categories and a set F of features, the category utility is defined as follows:

$$cu(C, F) = \frac{1}{m} \sum_{k=1}^{m} p(c_k) \left[\sum_{i=1}^{n} p(f_i|c_k)^2 - \sum_{i=1}^{n} p(f_i)^2 \right] \tag{3.2}$$

where $p(f_i|c_k)$ is the probability that a member of category c_k has the feature f_i, $p(c_k)$ is the probability that an instance belongs to category c_k, $p(f_i)$ is the probability that an instance has feature f_i, n is the total number of features, m is the total number of categories.

Features of instances are represented by tags in folksonomies. Accordingly, in the definition of category utility, the tag set T is used as the feature set F and a tag t_i is used as a feature f_i. As we model, the importance of tags are different in folksonomies. To take the differences of tag importance into account, we modify the definition and add the weight w_i of tag t_i into the definition:

$$cu(C,T) = \frac{1}{m}\sum_{k=1}^{m} p(c_k)\left[\frac{\sum_{i=1}^{n_k} w_i p(t_i|c_k)^2}{n_k} - \frac{\sum_{i=1}^{n} w_i p(t_i)^2}{n}\right] \tag{3.3}$$

where w_i is the weight of the tag t_i, n_k is the number of unique tags in cluster c_k, n is the number of all unique tags. To reflect the mean weight of a tag, w_i is defined as:

$$w_i = \frac{1}{N_{t_i}}\sum_{j=1}^{N_{t_i}} v_{j,i} \tag{3.4}$$

where N_{t_i} is the number of resources annotated by tag t_i and $v_{j,i}$ is the weight of the tag t_i in resource r_j. To differentiate it from the original definition, we consider it as the weighted category utility.

3.1.3 Basic Level Concepts Detection Algorithm

Because basic level categories (and concepts) have the highest category utility, the problem of finding basic level categories (and concepts) becomes an optimization problem using category utility as the objective function. The value of category utility is influenced by the intra-category similarity which reflects the similarity among members of a category. Categories with higher intra-category similarity have higher value of category utility. Accordingly, we put the most similar instances together in every step of our method until the decrease of category utility. While it is possible to have different functions for similarity measure of two instances r_i and r_j, we argue that the function $sim(r_i, r_j)$ should satisfies the following axioms:

Axiom 3.1 $0 \leq sim(r_i, r_j) \leq 1$,$sim(r_i, r_j) = 0$ *if* r_i *and* r_j *have no common tags,* $sim(r_i, r_j) = 1$ *if* $v_{i,k} = v_{j,k}$, *for all k=1,...,n.*

Axiom 3.2 $sim(r_i, r_j) > sim(r_i, r_l)$, *if* $0 < v_{i,k} < v_{j,k} < v_{l,k}$ *or* $0 < v_{l,k} < v_{j,k} < v_{i,k}$ *and* $v_{l,m} = v_{j,m}$ *for all* $m \neq k$.

Axiom 3.1 specifies the boundary cases of similarity measure. Axiom 3.2 specifies the influence of tag weight. The deviation of the weight of tag is larger, the similarity is lower. In commonly used methods of computing similarity between two vectors, cosine coefficient is a suitable method to satisfy these conditions, which computes the cosine angle between two vectors. In addition, we find that tags appearing in fewer documents are more important for categorization than those appearing in more documents [58]. Accordingly, the similarity measure metric is defined as follows:

$$sim(r_i, r_j) = \frac{\sum_{k=1}^{n} idf(t_k) \cdot v_{i,k} \cdot v_{j,k}}{\sqrt{\sum_{k=1}^{n} v_{i,k}^2} \cdot \sqrt{\sum_{k=1}^{n} v_{j,k}^2}} \tag{3.5}$$

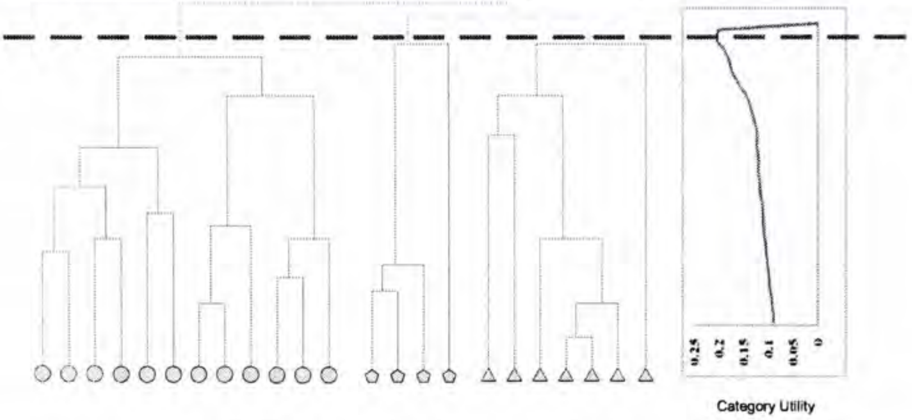
where r_i, r_j are two instances, n is the total number of unique tags describing them, and $v_{i,k}$ is the value of tag $t_{i,k}$ in instance r_i, if r_i does not have the tag, the value is 0. $idf(t_k)$ is the inverse document frequency of the tag t_k.

$$idf(t_k) = \log_N(\frac{N}{N_{t_k}}) \tag{3.6}$$

where N is the total number of resources and N_{t_k} is the number of resources annotated by tag t_k, $0 \leq idf(t_k) \leq 1$. $idf(t_k)$ gets the value 0, when the tag t_k is assigned to all resources. In this case, all resources have this tag, the tag is useless for categorization and identification. $idf(t_k)$ gets the value 1, when only one resource annotated by tag t_k.

In our algorithm [59], firstly, we consider every single instance itself as a concept. This type of concept which only includes one instance is considered as the bottom level concepts. Secondly, we compute the similarity between each pair of concepts and build the similarity matrix. Thirdly, the most similar pair in the matrix is identified and merged into a new concept. The new concept contains all instances of the two old concepts and holds their common properties. After that we reconsider the similarity matrix of the remaining concepts. We apply this merging process until only one concept is left or the similarity between the most similar concepts is 0. We then determine the step where the categories have the highest category utility which is the local optimum of category utility. These categories are considered as the basic level categories and the concepts are considered as the basic level concepts. For example, in the left part of figure 3.1, 23 instances are classified into 3 categories (concepts) represented by circles, pentagons and triangles respectively. In every step, the most similar instances are merged into one concept. In figure 3.1, finally all instances are merged into one concept and the process is similar to building a dendrogram. The category utility of the result after every merging step is shown in the right part of figure 3.1. The category utility gets the highest value when only 3 concepts left as shown by the red dashed line, which is the result of our algorithm. The detail of this algorithm is given in algorithm 1, and the time complexity is $O(N^2 \log N)$ where N is the number of resources.

Figure 3.1 An example of algorithm 1



3.1.4 Ontology Generation Algorithm

Using algorithm 1, we can extract basic level concepts from a set of instances. For the reason that basic level concepts are considered cognitively basic (learned by human easily and quickly), building the ontology with basic level concepts is our objective. The ontology built through our method has the psychological character that every concept in the ontology is basic level concept, which differentiates the ontology built through our method to the ontology built in previous ontology learning research. To achieve our goal, we build the ontology in a top-down way based on algorithm 1. We first generate a root concept including all instances. After using algorithm 1 to find the basic level concepts, we add the basic level concepts to the ontology as sub-concepts of the root. Then, we apply algorithm 1 iteratively to the instances of those sub-concepts and add their sub-concepts until they are the bottom level concepts. After several iteration, the ontology are built. The detail of this ontology generation method is given in algorithm 2.

Algorithm 1 Basic Level Concepts Detection

```
Input: R, a set of instances (resources)
Initialize C, C is an n dimensions vector $C = (c_1, c_2, ..., c_n)$ where its element $c_i$ is the bottom level concept. $C_{size}$ is equal to the number of elements in $C$. Set sim[n][n] as the similarity matrix of C, $sim[i][j] = sim(c_i, c_j)$. $S = (s_1, s_2, ..., s_n)$, $s_i$ is used to record the clustering result of step i.
Set $s_1 = C$, step=1,
while $C_{size} > 1$ do
    step++
    Find the most similar concepts in C and define a new concept include all instances of them.
    Delete the most similar concepts from C, and add the new concept into C.
    Update the similarity matrix.
    $C_{size} = C_{size} - 1$
    Record the result, $s_{step} = C$
    Compute the category utility of this step $cu_{step}$
end while
Find the step with the highest category utility $cu_{max}$, define the record of this step $s_{max}$ as the basic level categories.
Define the concept of each basic level category. The concept include all instances of the category and the properties of the concept are the common features (tags) of the instances.
Output these concepts.
```

3.2 Evaluation

3.2.1 Data Set and Experiment Setup

Experiments are performed on three genres of real world data : PROGRAMMING LANGUAGE, SPORT and GAME. The PROGRAMMING LANGUAGE data set consists of 1087 resources. The SPORT data set consists of 552 resources. The GAME data set consists of 645 resources. These data sets are crawled from delicious.com. As Golder [8] demonstrated, in delicious.com, each tag's occurrence frequency become fixed after a resource is bookmarked 100 times. The fixed frequency reflects the importance of a tag. To make sure that the frequency is nearly fixed, the web pages in our data sets are the ones which

Algorithm 2 Ontology Generation

```
Input: Concept $c$
Use algorithm 1 to explore basic level concepts from instances in $c$.
if the size of $s_{max} > 1$ then
    for every element $c_i$ in $s_{max}$ do
        Set $c_i$ as the sub-concept of $c$
        Use algorithm 2 with input $c_i$.
    end for
end if
```

are bookmarked more than 100 times in delicious.com. In addition, the web pages in our data sets must appear in both delicious.com and Open Directory Project (ODP) [1] because we use ODP as the gold standard to evaluate the ontology built by our method.

ODP is a user-maintained hierarchical web directory. Each directory in ODP has a label describing its name (e.g. "Arts" or "Python") and is a category of web pages. To derive the gold standard category structure from ODP, we first choose a category in the hierarchy of ODP, for example "Programming Languages" and then include all its sub-categories and their descendants into the category structure. These categories in ODP are created, verified and edited by thousands of users. ODP is considered as an user-generated ontology. The label of each directory is the name of the concept and the web pages in the directory are considered as the instances of this concept.

Furthermore, to filter the noise tags, we preprocessed each data set by (a) removing stop words and tags whose weight is less than the threshold q; (b) down casing the obtained tags.

3.2.2 Quantitative Analysis

Using ODP as the gold standard for evaluation, we apply F1 score [33] to compare the ontology built by our approach with ODP. F1 score is a measure

[1] http://www.dmoz.org/

Table 3.1 Statistics of the extracted ontologies

Data Set	#Resources	#Tags	#Users	#Concpets	#Levels
PROGRAMMING LANGUAGE	1087	39475	57976	422	6
SPORT	552	18776	31741	273	5
GAME	645	20352	39224	313	5

of a categorization result's accuracy according to the standard, which is the harmonic mean of precision and recall. If the ontology is more similar to ODP, the F1 score will be higher, which means the ontology is more consistent with human thinking. Experiments are first carried out on the PROGRAMMING LANGUAGE data set with different values of threshold q. Figure 3.2 presents the F1 scores of the results obtained by using different values of q. We find that if we do not filter any tags ($q = 0$), the clustering results will be the worst (0.011). Among different values of q, 0.02 gives the best result. Accordingly, we set $q = 0.02$ in our experiments first.

Figure 3.2 The impact of threshold q.

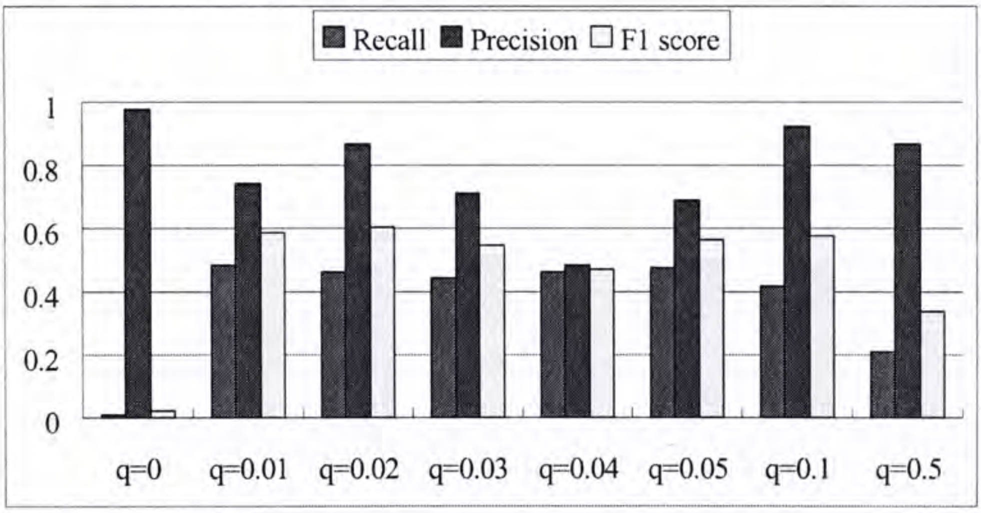


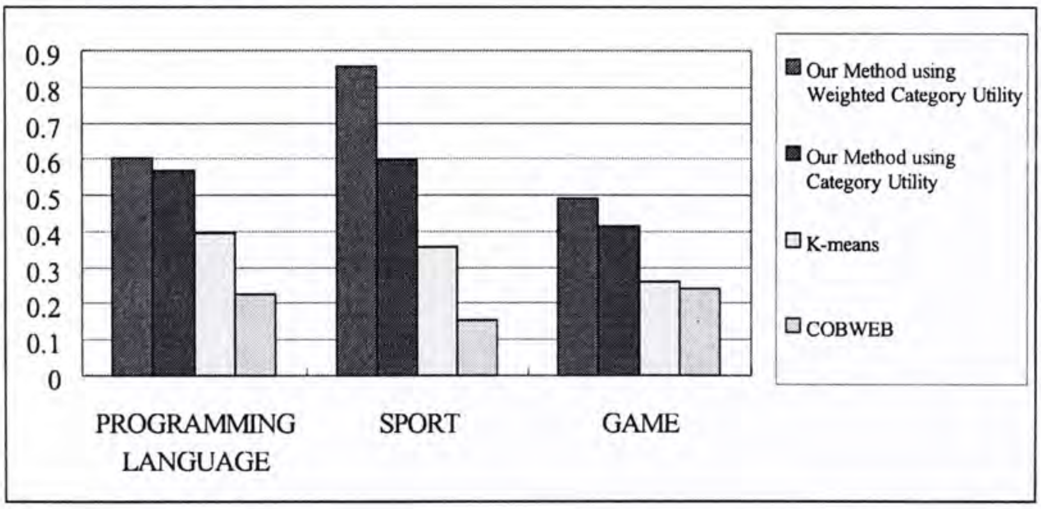
Table 3.1 shows the statistics of the ontologies extracted from the three data sets. The hierarchy of the ontology extracted from the PROGRAMMING LANGUAGE data set has 6 levels from the root concept to the bottom level concepts and contains 422 concepts (except bottom level concepts). The hierarchy of the ontology extracted from the SPORT data set has 5 levels and contains 273 concepts (except bottom level concepts). The hierarchy of the

ontology extracted from the GAME data set has 5 levels and contains 313 concepts (except bottom level concepts).

In previous research of ontology learning from folksonomies [6][7], researchers ignore the instances and categories. They define tags as concepts and only explore the relationship between these tags. There is not any category structure in the ontology generated by previous approaches. Their methods cannot organize instances into a category structure as ours. Accordingly it is impossible to compare the category structure of the ontology generated by our method with them. As commonly used clustering methods, K-means and concept clustering algorithm COBWEB can cluster instances into different categories. We compare the category structure built by our method with that built by K-means (when K is equal to the number of categories in ODP and Euclidean metric is used to determine the distance of two instances) and COBWEB to demonstrate the effectiveness of our approach on categorization.

In Figure 3.3, we show F1 scores of the results using different algorithms in the three data sets (PROGRAMMING LANGUAGE, SPORT and GAME). It is observed that our algorithm performs better than others especially in the sports data set (0.855) that means the category structure built by our method is more consistent with ODP than others. In sports domain, the basic level categories are explicit so that they can easily be detected. Basketball, football, running and other types of sports form the basic level categories in this domain (referring to table 3.2). In addition, the content of web pages in sports domain is unambiguous and the noise tags are fewer than in other domains. The result in the GAME data set is not as good as others because the ODP categories in this domain do not lay on the basic levels in our opinion. The F1 score of the results using our approach in the PROGRAMMING data set is 0.604 which is about 50% higher than the results using K-means. K-means has problems when clusters are of differing sizes, densities and non-globular shapes which is the situation of real world data set especially web resources. In this sense, our approach is much better than K-means. We also compare our approach with COBWEB [23] which is an incremental conceptual clustering algorithm also aiming to maximize category utility as our approach. In COBWEB, they use a incremental strategy to add instances to the category structure one by one. Although this strategy is flexible, the limitation is that the structure determined in previous steps cannot be rebuild later. Accordingly, the order of the instances will impact the quality of the result which make the quality uncertain. To solve this problem and improve the quality, our approach consider the whole data set first and always merge the most similar ones together. This strategy makes sure that we are finding the basic level categories in the whole data set. In addition, our method performs better using weighted category utility as the metric than using category utility in the three data sets because weighted category utility considers the difference of tags which is the situation in folksonomies.

Figure 3.3 F1-scores of the category structure built by different algorithms in the three data sets.



3.2.3 Qualitative Analysis

In this section, we will discuss the quality of the ontologies generated by our method. The ontology generated by our method is similar to ODP ontology.

Figure 3.4 Percentage of Different Relations

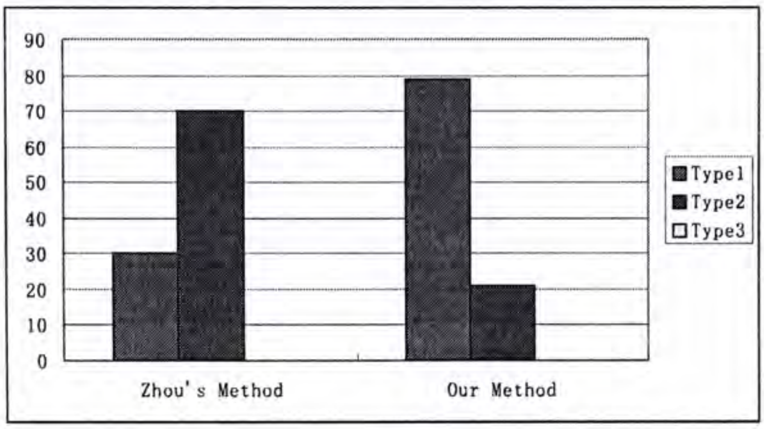


Table 3.2 shows the similar pairs between ODP concepts and concepts in the ontologies generated by our method. Concepts generated by our method are described in the form *(tag:value,...,)*. Concepts in ODP are described in the form *(label)*. The tags from super-concepts are not shown in the table because of the limit of space, e.g. the concept *(.net:0.349)* should be *(programming:0.415, .net:0.349)*. Most sub-concepts of *(programming:0.3)* are about programming languages in this data set, such as Java, Python and Ruby. This is consistent with the basic level concepts of programming language domain in human thinking. As shown in table 2, Properties of these concepts are related with labels of ODP concepts. There are totally 15 similar pairs (47% of the sub-concepts). In addition, in the SPORT data set, there are 12 similar pairs (23% of the sub-concepts) and in the GAME data set, there are 6 similar pairs (37.5% of the sub-concepts). Table 3.2 also shows the similar concepts in different levels of the ontology such as the sub-concepts of the concept *(java:0.730)*. These concepts do not seem as good as previous ones because in these levels the number of resources or the instances are not enough to support the ontology. These similar concepts and the relations between concepts demonstrate that our method is effective on generating ontologies with basic level concepts and the generated ontologies are meaningful and consistent with human thinking.

According to the research of Zhou et al. [7], we notice that the relations

Table 3.2 Similar Concepts between ODP and Ontology extracted by Our Method

	ODP	Ontology extracted by Our Method
	sub-concepts of (programming)	sub-concepts of (programming:0.3)
1	(c-sharp)	(.net:0.349)
2	(assembly)	(assembly:0.508, asm:0.244, assembler:0.256, development:0.105)
3	(c++)	(c++:0.641, development:0.155)
4	(c)	(c:0.522)
5	(pl-sql)	(database:0.450, development:0.100)
6	(sql)	(erlang:0.889)
7	(java)	(java:0.730)
8	(javascript)	(javascript:0.704)
9	(lisp)	(lisp:0.661)
10	(perl)	(perl:0.800)
11	(php)	(php:0.745)
12	(python)	(python:0.853)
13	(ruby)	(ruby:0.690)
14	(scripting)	(scripting:0.280)
15	(delphi)	(software:0.173, development:0.178, delphi:0.743)
	sub-concepts of (sports)	sub-concepts of (sport:0.498)
1	(Baseball)	(baseball:0.736)
2	(Basketball)	(basketball:0.535)
3	(Boxing)	(boxing:0.695)
4	(Cricket)	(cricket:0.698)
5	(Cycling)	(cycling:0.425, bike:0.395)
6	(football)	(soccer:0.397, football:0.459)
7	(golf)	(golf:0.809)
8	(hockey)	(hockey:0.603)
9	(Martial_Arts)	(martialart:0.299, martial_art:0.136)
10	(Motorsports)	(racing:0.325, new:0.215, motorsport:0.266)
11	(running)	(running:0.708, fitness:0.229)
12	(Water_Sports)	(surf:0.448, surfing:0.454)
	sub-concepts of (games)	sub-concepts of (game:0.417)
1	(online)	(free:0.184, online:0.065)
2	(gambling)	(gambling:0.337)
3	(card_games)	(poker:0.883)
4	(roleplaying)	(rpg:0.442)
5	(puzzles)	(puzzle:0.421)
6	(board_games)	(chess:0.802)
	sub-concepts of (python)	sub-concepts of (python:0.853)
1	(WWW)	(web:0.320)
2	(Development Tool)	(development:0.162, software:0.115, tool:0.109)
	sub-concepts of (java)	sub-concepts of (java:0.730)
1	(FAQs, Help, and Tutorials)	(tutorial:0.204, reference:0.153)
2	(Development Tools)	(software:0.107, tool:0.288)
3	(Applications)	(opensource:0.317, software:0.152)
4	(Personal Pages)	(development:0.138, blog:0.347)
	sub-concepts of (Soccer)	sub-concepts of (soccer:0.397, football:0.459)
1	(Video Games)	(video:0.491)
2	(Statistics)	(statistic:0.245, stat:0.136)
3	(News and Media)	(news:0.284)
	...	...

between different tags or concepts mainly include three types. (1) B is the sub-type of A. (e.g. "java" is sub-type of "programming") (2) B is a related aspect of A. (e.g. "development" is related with "programming") (3) B is parallel to A. (e.g. "java" is parallel to "python"). According to the definition of ontologies, the relations between concepts of different levels should be type 1. To demonstrate the effectiveness of our approach on generating hierarchical structure of ontologies, we compare the relations between first level concepts and second level concepts in the ontology generated by our method with that generated by Zhou's method. The result is shown in figure 3.4. The result shows that the percentage of type 1 (sub-type) relation in the ontology generated by our method (79%) is much higher than that generated by Zhou's method (30%). The percentage of type 2 relation is 21% and 70% respectively. In addition in this situation, there is no type 3 relation. The result demonstrates that the hierarchical structure in the ontology generated by our method, to some extent, is better than that generated by Zhou's method.

Chapter 4

Context Effect on Ontology Learning from Folksonomies

Inspired by studies in cognitive psychology, we try to model human cognitive process in folksonomies so that we can explore the implicit semantics and build more human acceptable and applicable concepts (ontology). In cognitive psychology, basic level concepts are frequently used by people in daily life, and most human knowledge is organized with them. In addition, contexts play an important role in concept learning. The basic level concepts will shift based on different contexts and categorization. Taking contexts into consideration will make our proposed method more completed and applicable.

In this chapter, we discuss the context effect on ontology learning from folksonomies especially basic level concepts detection and a metric named contextual category utility is proposed to take context into account [60]. Based on the contextual category utility, we propose a method to detect basic level concepts in different contexts. To the best of our knowledge, it is the first work on detecting basic level concepts in different contexts from folksonomies. We conduct experiments to evaluate our method using a real-world data set and compare the detected concepts with ODP concepts. Experiment results demonstrate that our method can detect basic level concepts in different contexts effectively.

4.1 Context-aware Basic Level Concepts Detection

4.1.1 Modeling Context in Folksonomies

According to the studies in cognitive psychology, contexts play an important role in human cognitive process. In such a process, there is a set of persons in a context and some subjective aspects of them should be considered as a part of context (e.g. the goal of using a concept, the knowledge of persons). According to the research finished by Tananka and Taylor [11], there is one very interesting cognitive psychology phenomenon: the shifting of the basic level. People with different domain knowledge have different considerations of the basic level. The domain knowledge has an effect on where the basic level lies. This difference is considered as the effect of contexts. As we mentioned above, a folksonomy consists of a set of resources, a set of tags and a set of users. Users with different domain knowledge annotate the resources with different tags. These tags naturally represent users subjective aspects including purposes and knowledge. Thus, we define a context x as a collection of relevant subjective aspects of users.

Definition 4.1. *A* **context**, *denoted by x, is a tuple, which consists of a subset of users and tags, $x =< N_u, N_t >$* , where N_u is a set of users and N_t is a set of tags which represents the subjective aspects of users.

In a particular context, some tags are more important than others [26]. In our model, the importance of each tags is indicated by a real number (i.e., weight of a tag) whose value is between 0 and 1. If a tag is absolutely important for a task in a specific context, then its weight is 1. If a tag is not important at all for a task in a specific context, then its weight is 0. We define a tag weight vector which reflects importance of tags in a context.

Definition 4.2. *A* **tag weight vector** *in a context x, denoted by V^x, is represented by a vector of tag:value pairs, $V^x = (t_1 : v_1^x, t_2 : v_2^x, \ldots, t_n : v_n^x)$,* $0 \leq v_{i,k} \leq 1$, where n is the number of relevant tags and v_i^x is the weight of tag t_i in context x.

Based on subjective aspects, users can form a perspective so as to obtain a set of weights for tags in a context. We formally define a perspective as follows:

Definition 4.3. *A* **perspective**, *denoted by π^x, maps a set of users and a set of tags to a tag weight vector, $\pi^x(N_u, N_t) = V^x$*, where V^x is a tag weight vector, N_u is a set of users and N_t is a set of tags.

For the reason that a perspective is formed based on subjective aspects of users, we consider that such a mapping is accomplished by the users in a context and the weight vector is given by the users. For example, people who are interested on programming languages may give a tag weight vector as: $V^x = (\textit{java} : \textit{1}, ..., \textit{css} : \textit{0.5})$. It means that the tag "java" is absolutely important and "css" is less important in the context. People may have different perspectives in contexts and give different property weight vectors with respect to their own perspectives.

4.1.2 Context Effect on Category Utility

In folksonomies, features of instances are represented by tags. Accordingly, in the definition of category utility, the features set F should be changed to the tags set T, and feature f_i should be changed to tag t_i, where $f_i \in F$, $t_i \in T$. In cognitive psychology, under different contexts the basic level concepts are different. Accordingly, we should consider the effect of contexts on category utility. The importances of tags are different in folksonomies under different contexts. To consider the differences in tag importance, we add the tag weight vector V^x of context x to the definition of category utility. Considering the context, the metric of predicting performance should be positively correlated with the tag weight in a certain context. So we change the metric of predicting performance from the correctness $p(t_i)^2$ to $v_i^x \cdot p(t_i)^2$. Furthermore, in folksonomies each resource has different number of tags, and we hope category utility will not be affected by this difference. As a result, we consider the impact of one tag on average in category utility and $\sum_{i=1}^{n} p(f_i)^2$ is changed to $\frac{\sum_{i=1}^{n} v_i^x \cdot p(t_i)^2}{n}$. Accordingly, the contextual category utility is then defined as follows:

$$cu(C,T,x) = \frac{1}{m}\sum_{k=1}^{m} p(c_k)\left[\frac{\sum_{i=1}^{n_k} v_i^x p(t_i|c_k)^2}{n_k} - \frac{\sum_{i=1}^{n} v_i^x p(t_i)^2}{n}\right] \tag{4.1}$$

where C is the set of categories, T is the set of tags, x is the context. n_k is the number of unique tags in cluster c_k and n is the number of all unique tags. v_i^x is defined as the value of tag t_i in V^x which is the tag weight vector of context x.

4.1.3 Context-aware Basic Level Concepts Detection Algorithm

Referring to algorithm 1, to detect the basic level concepts we put the most similar instances together in every step of our method until the decrease of category utility. To compute the similarity, we use the cosine coefficient which is a commonly used method of computing similarity between two vectors in information retrieval. In addition, taking the context effect into consideration (the metric should be positively correlated with the tag weight), we add the tag weight into the definition of cosine coefficient. Accordingly, the similarity metric is defined as follows:

$$sim(a,b,x) = \frac{\sum_{k=1}^{n} v_k^x \cdot v_{a,k} \cdot v_{b,k}}{\sqrt{\sum_{k=1}^{n} v_{a,k}^2} \cdot \sqrt{\sum_{k=1}^{n} v_{b,k}^2}} \qquad (4.2)$$

where a, b are two concepts, n is the total number of unique tags describing them, and $v_{a,k}$ is the value of tag $t_{a,k}$ in concept a, if a does not have the tag, the value is 0. v_k^x is defined as the value of tag t_k in V^x which is the tag weight vector of context x.

The algorithm of context-aware basic level concepts detection is similar to algorithm 1. Firstly, we construct bottom level concepts where each concept only includes one instance. Secondly, we compute the similarity between each pair of concepts and build the similarity matrix. Thirdly, the most similar pair in the matrix is generated and merged into a new concept. The new concept contains all instances of the two old concepts and holds their common properties. We apply this merging process until the decrease of category utility. Taking context into consideration, the detail of this method is shown in algorithm 3.

4.2 Evaluation

4.2.1 Data Set and Experiment Setup

Our experiments are conducted on a real world data set (the PROGRAMMING LANGUAGE data set in chapter 3): 1087 web pages which are associated with 39475 tags and 57976 users. These web pages are all in the programming domain. Golder [8] demonstrated that, in delicious.com, in a resource each tag's frequency becomes a nearly fixed proportion of the total frequency of all tags after the resource is bookmarked 100 times. The fixed proportion reflects the real value of a tag in the resource. To make sure that the proportion is nearly

Algorithm 3 Context-aware Basic Level Concepts Detection

```
Input: $R$, a set of instances (resources); $V^x$, the tag weight vector of context $x$
Initialize C, C is an n dimensions vector $C = (c_1, c_2, ..., c_n)$ where its element $c_i$ is the bottom level concept. $C_{size}$ is equal to the number of elements in $C$. Set sim[n][n] as the similarity matrix of C, $sim[i][j] = sim(c_i, c_j, x)$. $S = (s_1, s_2, ..., s_n)$, $s_i$ is used to record the clustering result of step i.
Set $s_1 = C$, step=1,
while $C_{size} > 1$ do
    step++
    Find the most similar concepts in C and define a new concept include all instances of them.
    Delete the most similar concepts from C, and add the new concept into C.
    Update the similarity matrix.
    $C_{size} = C_{size} - 1$
    Record the result, $s_{step} = C$
    Compute the contextual category utility of this step $cu_{step}$
end while
Find the step with the highest category utility $cu_{max}$, define the record of this step $s_{max}$ as the basic level categories.
Extract concepts of basic level categories. A concept includes all instances of a category and the properties of the concept are the common features (tags) of the instances.
Output these concepts.
```

fixed, the web pages in our data sets are the ones which are bookmarked more than 100 times in delicious.com. In addition, the web pages in our data sets appear in both delicious.com and Open Directory Project (ODP) [1] because we use ODP as the gold standard. ODP is a user-maintained web directory. Each directory is considered as a concept in ODP. To derive the gold standard concepts from ODP, we first choose a certain directory (e.g. programming) in ODP and then consider all its sub-directories as the gold standard concepts. These concepts in ODP are created, verified and edited by experts around the world and accepted by many users. For evaluation, we apply F1 score which is the aggregation of recall and precision [33] to compare concepts detected by

[1] http://www.dmoz.org/

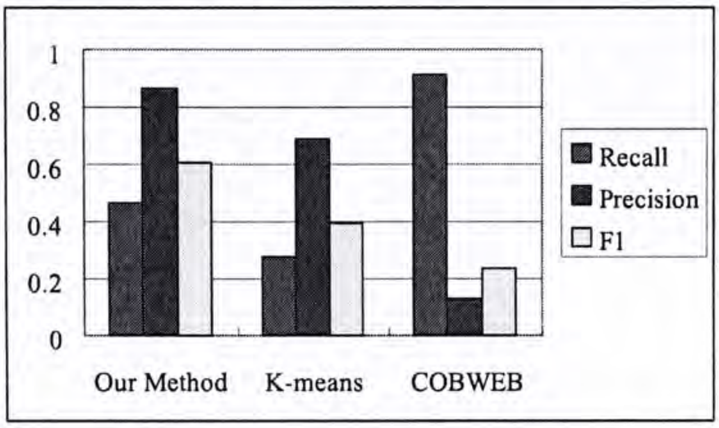
our approach with ODP concepts on their category structures. In addition, we ask people to evaluate our experiment results through questionnaires. To filter the noise tags, we also preprocess the data set by (a) removing tags whose weight is less than the threshold $q = 0.02$; (b) down casing the obtained tags.

4.2.2 Result Analysis

As we mentioned, we model a context in folksonomies through a tag weight vector V^x in which different tags have different values according to their importance in the context. In our experiments, we use questionnaires to get people's consideration on tag weights in different contexts. In the questionnaire, we ask 20 people to give weights to different tags given the context information (we ask them to give marks to tags where "0" means the tag is not related to the context, "1" means a little bit related to the context, "2" means moderate and "3" means highly related). The value of a tag in the tag weight vector of a context is the average mark of the 20 people after normalizing to the range from 0 to 1. If we are not given any information about the context (i.e., we do not take context into consideration), the weights of all tags are the same and equal to 1. We use two traditional categorization methods as baselines, which are K-means clustering and a concept clustering algorithm named COBWEB [23]. For the reason that the traditional categorization method do not take context into consideration and no information about the context are given, we compare our method with traditional methods without context information first.

In figure 4.1, we show the results obtained by different methods in our data set without context information. The F1 score of the result using the traditional K-means algorithm (when K is equal to the number of categories in ODP and Euclidean metric is used to determine the distance of two instances) is 0.393. Our approach outperforms K-means by about 50% and the F1 score

Figure 4.1 Comparison of different methods without context information



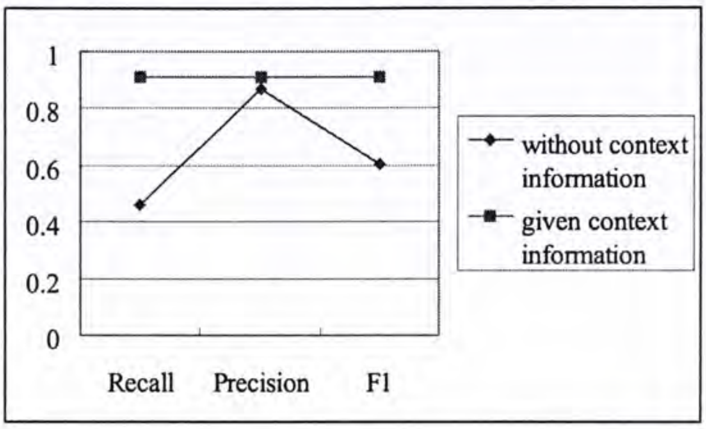
of our method is about 0.599. In addition, our method outperform COBWEB by more than 100% on F1 score. In the result of COBWEB, most of web pages are classified to one concept which is not reasonable so the value of recall is nearly 1 but the precision is only 0.127 and the F1 score is only 0.237. On precision, our proposed method also has the highest value (i.e., about 0.463).

In our method, we take context into consideration and make our method to be context-aware for categorization and concept learning. To indicate our method is context-aware, we discuss two contexts (which are denoted by C_{pl} and C_{os} respectively) in our experiments for the same 1087 web pages which are in the programming domain. In the context C_{pl}, people whose interests are on programming languages are trying to classify these web pages. In the context C_{os}, people whose interests are on operation systems are trying to classify these web pages.

Result Analysis for Context C_{pl}.

In context C_{pl}, users try to classify web pages based on the interest of themselves. As mentioned, the interest of users in this context is programming languages. To model this context, we ask 20 students majoring in computer science to give weights to tags based on the interests in C_{pl}. The tag weight vector of this context is *(java:0.9, mac:0.1, unix:0.3, c:1.0, .net:0.75, ruby:0.6,*

Figure 4.2 Result of our method with (out) context information



window:0.3, web:0.4, blog:0.0, ...). Tags which are related with programming languages have high weights such as "java" and "c" whose weight are 0.9 and 1.0 respectively. Tags which are not quite related with programming languages have low weights such as "unix" whose weight is only 0.3.

We compare the detecting basic level concepts using our method in this context with sub-concepts of "programming languages" in ODP. According to figure 4.2, while we take the context information into consideration, the categorization results and the concepts will be improved. The F1 score is 0.599 without the context information, and while given the information the F1 score increase to 0.912. For our method, the F1 score obtained by given the context information outperform that without context information about 50%. In addition, given the context information, our method also dominates on recall and precision. As discussed in previous chapter, our method without context information is already an effective one especially on putting similar resources together. Accordingly, the precision score is already good (0.865). However, without context information it is hard to detect which kind of basic level concepts we are expecting or which clusters should be combined together. Many small clusters are not merged without context information so that the value of recall is low (0.463). The value of recall is greatly improved (from 0.463 to 0.911) given context information that means more similar clusters are combined, which is consistent with the standards. The detected concepts through our method are almost the same as gold standard concepts which demonstrates our assumption. We show the detected basic level concepts using our method in table 4.1. In table 4.1, concepts are represented by the form *(tag: value)*, for example *(java: 0.680)* where 0.680 is the average weight of the tag "java" in instances of the concept. In addition, we also ask 20 students to evaluate the basic level concepts detected in C_{pl} and the result is shown in table 4.2. People evaluate the results using the score from 0 to 10 where 10 means that people think the result is perfect under certain context. According to table 4.2, the average evaluation score given by people on the result of our method in context C_{pl} is 8.16. Such a score means that, given the tag weight vector in C_{pl}, our method can detect the basic level concepts which is consistent with people's expectation. We also can find that the detected concepts are not good with a much smaller evaluation score 4.22 without the context information. Such a result demonstrates the rationality of our context modeling approach and the efficiency of our context-aware basic level concept detection method.

Result Analysis for Context C_{os}.

In context C_{os}, users try to classify web pages based on the interest of them and their interest is operation systems. The context is also given by the 20 students and the basic level concepts detected in this context are shown in table 4.1. *(linux:0.406), (windows:0.362), (mac:0.410, osx:0.393, macosx:0.092)* are all concepts about operation systems include Linux, Windows and MacOS. Under this situation, the evaluation results in table 4.2 show that given the context information (i.e., the tag weight vector), we can build the concepts which is consistent with people's expectation and the evaluation score with context information is 7.88 which is much better than the result without the information (2.13).

Table 4.1 Basic level concepts detected in different contexts

Context	Basic Level Concepts
context C_{pl}	(xml:0.635), (javascript:0.599), (smalltalk:0.651), (html:0.252), (delphi:0.743), (sql:0.502, database:0.476), (cocoa:0.354, mac:0.213,apple:0.212,osx:0.226), (haskell:0.753), (python:0.812), (basic:0.185), (perl:0.751), (java:0.680), (lisp:0.633), (ruby:0.652), (php:0.651), (c:0.238), (c++:0.687, cpp:0.047), (fortran:0.181)
context C_{os}	(linux:0.406), (windows:0.362), (mac:0.410, osx:0.393, macosx:0.092)

Table 4.2 Evaluation of basic level concepts in C_{pl} and C_{os}

	C_{pl}	C_{os}
given context information:	8.16	7.88
without context information:	4.22	2.13

These experiments demonstrate the existence of context and its effect on human concept learning process especially basic level concepts learning. In different contexts, the basic level concepts are different. In addition, they also demonstrate that our method outperforms previous methods in detecting basic level concepts. The concepts detected by our method are approximate to human's expectation. What is more, our method can detect different basic level concepts in different contexts while previous methods cannot.

Chapter 5

Potential Applications

5.1 Categorization of Web Resources

The exponentially increasing size of web pages creates a need of an efficient method to categorize and organize them. However categorizing web pages is a time consuming job for human as the reason that the content of web pages is various. In addition, there are many different types of resources in Internet such as photos (e.g., flickr[1]), videos (e.g., youtube[2]) and movies (e.g. imdb[3]). An efficient approach to categorize and organize web resources will benefit their future use.

In this thesis, we provide an automatical approach to organize resources as a hierarchical category structure. The hierarchical category structure is actually a taxonomic (subclass) hierarchy. In other words, if category A is a sub-category of category B, every instance of A must be an instance of B. This type of category structure is consistent with human thinking and an efficient structure for future searching. Another character of our approach is that we do not require any training data. Given a set of resources, we can efficiently organize them in different categories. Our approach provides a possible solution of categorizing and organizing web pages and other resources

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

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

[3] http://www.imdb.com

such as photos, books and movies in Internet.

5.2 Applications of Ontologies

Ontologies have a lot of applications in the Semantic Web, multiagent systems, information retrieval systems, etc. The ontology generation approach proposed in this thesis is designed to improve previous ontology learning methods and enhance the applicability of ontologies. In addition, ontologies generated by our approach have many immediate applications, such as collaborative tagging, tag aided search and tag recommendation.

Firstly, ontologies play an important role in the Semantic Web. The Semantic Web is a technological movement which towards a more structured Web where resources are described by machine-readable ontologies. Accordingly, in the Semantic Web agents can access information automatically, resulting in more efficient and effective information processing. With ontologies, searching information and resources from the Web will become much more efficient and effective because agents are able to understand the semantics of resources. In the Semantic Web, searching of information is actually an action of querying an ontology to retrieve resources which satisfy some conditions[61].

Secondly, the semantic relations between tags defined in ontologies can specify the searching and crawling process. As an example, if a search engine is asked to find some web pages about programming languages, according to the ontologies generated in the PROGRAMMING LANGUAGE data set, the engine will notice that the sub-concepts of "programming language" such as "Java", "C" and "PHP" are related with its target. These ontologies can also be used for knowledge representation in B2B interaction among sites and multi-agents communication.

Thirdly, ontologies built based on semantics in folksonomies will benefit folksnomies and improve the performance of collaborating systems. In folksonomies, when user are choosing a tag to annotate web resources or looking for resources related to a certain aspect (tag), through querying an ontology we can recommend some tags and resources for them to consider.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Ontology is essential for the Semantic Web and knowledge representation of Artificial Intelligence. As ontology building is a time consuming job for human, much research is conducted on automatically extracting ontologies from texts and other resources. In this thesis, inspired by cognitive psychology especially basic level categories theory, we explore the major problem of ontology learning from folksonomies. This thesis presents a novel idea to make use of implicit semantics in folksonomies. We present an algorithm to generate ontologies with basic level concepts. This type of ontology is considered as cognitive basic and more acceptable and applicable by users. Moreover we take context into consideration and successfully model the effect of context on cognitive progress of human, especially concept learning.

In our approach, we generate ontologies based on folksonomy tags which agree more closely with human thinking than those automatically extracted from text. Folksonomies have many advantages over formal taxonomies [4]. No complicated vocabularies need to be learned for users. They create and apply tags freely. In addition, folksonomies are open-ended and therefore respond quickly to changes in the way users describe objects. These advantages attract a lot of users. Ontologies generated from folksonomy tags may represent most users opinion about how to describe a web resource. We believe that these ontologies are easy to be accepted by others. However exploring ontologies from folksonomies faced many challenges due to following reasons:

- Most existing ontology learning methods focus on learning from text of well-defined terms. However, tags are given by users freely which may not appear in a standard vocabulary. This uncontrolled nature of folksonomy tags causes many problems. One is ambiguity. People may use the same word to present different meanings. Another is synonym. Different words can express the same meaning. In addition, there are redundant noise tags that don't have any meaning such as "todo".

- An ontology has a hierarchical structure. There are many relations between ontology terms such as hyponymy and associative relations. However, tags in folksonomies are considered as in a flat space. There are not any hierarchical relations among tags.

Exploring ontologies from folksonomies is not only valuable but also a challenging task. Much research has been conducted on this topic as we discussed in the background study chapter. However, their work focuses on hierarchy construction only, and they lack a principle for supervising the ontology extraction from a human's perspective. In other words, they consider little on what is a more acceptable and applicable ontology for users. For the reason that an ontology provides a users' shared vocabulary to model a domain, we consider that it is necessary and benefit to construct ontologies from users' perspective (i.e., taken how people thinking and using concepts into consideration). Compared with the previous research, our approach has three advantages as follows:

- Our method provides an effective hierarchical categorization approach to organize large amount of web resources.

- Our method builds ontologies under a cognitive psychology theory, basic level categories [10]. The generated ontologies will be more consistent with human thinking and reused easily.
- Our method takes context into consideration. We formally model context in folksonomies and give a novel context aware method for ontology learning. Context is an important part in human cognitive process and have an effect on human cognitive tasks.

To the best of our knowledge, it is the first work on discovering basic level concepts in folksonomies and using them to construct ontologies. In experiments, ontologies generated from three real-world data sets demonstrate the effectiveness of our approach on generating ontologies with basic level concepts. In addition, we consider the effect of context on concept learning and present a context-aware category utility to consider context in folksonomies. Through doing experiments on real world data set, we demonstrate not only the existence of context effect but also the effectiveness of our method on concept learning.

6.2 Future Work

This thesis presents a novel ontology learning approach. There are many potential future directions of this work.

Firstly, in this thesis, we present an ontology learning method which is consistent with human cognitive behaviors. This method is inspired by the basic level category theory in cognitive psychology. The psychology character differentiates our method from previous ontology learning methods. There still are a number of issues in cognitive psychology that can be used to enhance ontology learning and knowledge representation. As an example, it is obvious that properties of concepts are correlated to each other and there are different types of relations among properties. Empirical findings in cognitive psychology [20] [62] demonstrate that people do make use of this kind of information in cognitive tasks. As a future research direction, we can further investigate this issue and improve our approach. Moreover, the model of context in our algorithm requires further development and enhancement. How to get contexts automatically should be taken into consideration.

Secondly, taking fuzzy theory into consideration in our approach is valuable. Previous research on ontology has discussed the formal model of fuzzy ontology [63]. According to fuzzy theory, an instance is not only categorized to one concept but has different typicality degree to different concepts. This consideration of ontologies is flexible and will benefit the searching process on the Web. In some domains, such as searching for resources about fishes kept in an aquarium, user may not only be interested in fishes, but may also want to access information about other fish-like marine animals such as dolphins and whales, which strictly speaking are not classified as fishes.

Thirdly, using ontologies learned from folksonomies in real world applications is another important research topic. As an example, it would be useful to design a web search engine using these ontologies to assist its web searching task. The semantic relations between tags can specify the searching task by recommending related super-concepts and sub-concepts for users when they input some concepts into the search engine. Designing new searching and ranking algorithms to use ontologies is essential for this type of application. We also can use context-aware method to develop recommendation systems which can recommend different web resources to users with different contexts.

Finally, our algorithm presented in this thesis is set to find out the local optimal of category utility and it would be interesting to use a global optimization algorithm such as genetic algorithm and evolution algorithm to optimize category utility in the ontology learning process. Because finding the global optimal result is a extremely time consuming job, how to design an efficient algorithm to get close to the optimal result is also an interesting topic for future research.

Publication List

1 **Wen-hao Chen**, Yi Cai, Ho-fung Leung and Qing Li, Context-aware Basic Level Concepts Detection in Folksonomies. In: *Proceedings of The 11th International Conference on Web-Age Information Management*, Jiuzhaigou, China, 15-17 July 2010. Springer Verlag. (To Appear)

2 **Wen-hao Chen**, Yi Cai, Ho-fung Leung and Qing Li, Generating Ontologies with Basic Level Concepts from Folksonomies. In: *Proceedings of The 10th International Conference on Computational Science*, Amsterdam, The Netherlands, 31 May - 2 June 2010. Elsevier. (To Appear)

3 **Wen-hao Chen**, Yi Cai and Ho-fung Leung, An Unsupervised Method of Exploring Ontologies from Folksonomies. In: *Proceedings of The 10th International Conference on Computational Science and Its Applications*, Fukuoka, Japan, 23-26 March 2010. IEEE Computer Society.

Bibliography

[1] G. Antoniou and F. van Harmelen, *A Semantic Web Prime: Cooperative Information Systems*, Cambridge , Mass MIT Press, 2004.

[2] D. Ramage, P. Heymann, C. D. Manning, and H. G. Molina, Clustering the tagged web, in *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 54–63, New York, NY, USA, 2009, ACM.

[3] A. Mathes, Folksonomies-cooperative classification and communication through shared metadata, in *Computer Mediated Communication, LIS590CMC (Doctoral Seminar), Graduate School of Library and Information Science, University of Illinois Urbana-Champaign*, 2004.

[4] 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, ACM New York, NY, USA, 2006.

[5] H. Al-Khalifa and H. Davis, *Exploring the value of folksonomies for creating semantic metadata*, volume 3 of *International Journal on Semantic Web & Information Systems*, IGI Global, 2007.

[6] P. Mika, *Ontologies are us: A unified model of social networks and semantics*, volume 5 of *Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, 2007.

[7] M. Zhou, S. Bao, X. Wu, and Y. Yu, An unsupervised model for exploring hierarchical semantics from social annotations, in *Lecture Notes in Computer Science*, volume 4825, page 680, Springer, 2007.

[8] S. Golder and B. Huberman, Arxiv preprint cs/0508082 (2005).

[9] R. Kraft, F. Maghoul, and C. Chang, Y! q: contextual search at the point of inspiration, in *CIKM*, volume 5, pages 816–823, Citeseer, 2005.

[10] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem, *Basic objects in natural categories*, volume 8 of *Cognitive Psychology*, 1976.

[11] J. Tanaka and M. Taylor, *Object categories and expertise: Is the basic level in the eye of the beholder*, volume 23 of *Cognitive Psychology*, 1991.

[12] T. Lee, J. Hendler, and O. Lassila, *The semantic web*, volume 284 of *Scientific American*, 2001.

[13] A. Bharati, V. Chaitanya, R. Sangal, and K. Ramakrishnamacharyulu, *Natural Language Processing*, PHI, 2000.

[14] N. Guarino and R. Poli, *Formal ontology, conceptual analysis and knowledge representation*, volume 43 of *International Journal of Human Computer Studies*, Citeseer, 1995.

[15] G. Antoniou and F. Van Harmelen, *A semantic web primer*, MIT press, 2004.

[16] Y. Ding and S. Foo, *Ontology research and development*, volume 28 of *Journal of information science*, CILIP, 2002.

[17] C. M. Au Yeung and H. F. Leung, Ontology with likeliness and typicality of objects in concepts, in *Lecture Notes in Computer Science*, volume 4215, page 98, Springer, 2006.

[18] G. Smith, Atomiq/information Architecture [blog] (2004).

[19] C. Cattuto, D. Benz, A. Hotho, and G. Stumme, Semantic grounding of tag relatedness in social bookmarking systems, in *Proceedings of ISWC*, Springer, 2008.

[20] E. Smith and D. Medin, *Categories and concepts*, Harvard University Press Cambridge, MA, 1981.

[21] J. Ackrill, *Aristotle's Categories and De interpretatione*, Oxford University Press, USA, 1963.

[22] M. Gluck and J. Corter, Information, uncertainty, and the utility of categories, in *Proceedings of the seventh annual conference of the cognitive science society*, pages 283–287, 1985.

[23] D. Fisher, *Knowledge acquisition via incremental conceptual clustering*, volume 2 of *Machine learning*, Springer, 1987.

[24] K. M. Galotti, *Cognitive Psychology In and Out of the Laboratory*, Belmont, CA: Wadsworth, third edition, 2004.

[25] E. M. Roth and E. J. Shoben, *The effect of context on the structure of categories*, volume 15 of *Cognitive Psychology*, 1983.

[26] G. L. Murphy, *The big book of concepts*, MIT Press, 2002.

[27] P. Ozturk and A. Aamodt, Towardsa model of context for case-based diagnostic problem solving, in *Context '99*, pages 198–208, 1997.

[28] J. McCarthy, Notes on formalizing contexts, in *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 555–560, 1986.

[29] S. Buvac and I. A. Mason, Propositional logic of context, in *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 412–419, 1993.

[30] F. Giunchiglia and L. Serafini, *Multilanguage hierarchical logics, or: how we can do without modal logics*, volume 65 of *Artif. Intell.*, 1994.

[31] R. Guha, R. McCool, and R. Fikes, Contexts for the semantic web, in *Proceedings of the 3rd International Semantic Web Conference*, volume 3298, pages 32–46, 2004.

[32] V. Akman and M. Surav, *Steps Toward Formalizing Context*, volume 17 of *AI Magazine*, 1996.

[33] C. Manning, P. Raghavan, and H. Schtze, *Introduction to information retrieval*, Cambridge University Press, 2008.

[34] P. Buitelaar, P. Cimiano, and B. Magnini, Ontology learning from text: Methods, evaluation and applications , 3 (2005).

[35] G. Salton and C. Buckley, *Term-weighting approaches in automatic text retrieval*, volume 24 of *Information processing & management*, Elsevier, 1988.

[36] P. Qiu, *Recent advances in computational promoter analysis in understanding the transcriptional regulatory network*, volume 309 of *Biochemical and Biophysical Research Communications*, Elsevier, 2003.

[37] M. Lesk, Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone, in *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, ACM New York, NY, USA, 1986.

[38] Z. Harris, *Mathematical structures of language*, Interscience publishers, 1968.

[39] M. Baroni and S. Bisi, Using cooccurrence statistics and the web to discover synonyms in a technical language, in *Proc. of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Citeseer, 2004.

[40] M. Hearst, Automatic acquisition of hyponyms from large text corpora, in *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545, Association for Computational Linguistics Morristown, NJ, USA, 1992.

[41] P. Cimiano, A. Hotho, and S. Staab, *Learning concept hierarchies from text corpora using formal concept analysis*, volume 24 of *Journal of Artificial Intelligence Research*, AI Access Foundation, 2005.

[42] M. Sanderson, Deriving concept hierarchies from text, in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213, ACM New York, NY, USA, 1999.

[43] T. Rindflesch, L. Tanabe, J. Weinstein, and L. Hunter, EDGAR: extraction of drugs, genes and relations from the biomedical literature, in *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, page 517, NIH Public Access, 2000.

[44] D. Lin and P. Pantel, DIRT-discovery of inference rules from text, in *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328, Citeseer, 2001.

[45] L. Kaufman and P. Rousseeuw, *Agglometarive nesting (program AGNES), Volume 1 of 1, Chapter 5*, volume 14 of *New York: Wiley Inter-Science*, 1990.

[46] L. Kaufman and P. Rousseeuw, *Divisive analysis (program DIANA), Volume 1 of 1, Chapter 6*, volume 9 of *New York: Wiley Inter-Science*, 1990.

[47] J. Gennari, P. Langley, D. Fisher, C. U. I. D. O. INFORMATION, and C. SCIENCE, *Models of Incremental Concept Formation.*, Defense Technical Information Center, 1988.

[48] J. Diederich and T. Iofciu, Finding communities of practice from user profiles based on folksonomies, in *Proceedings of the 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice*, Citeseer, 2006.

[49] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme, *Information retrieval in folksonomies: Search and ranking*, volume 4011 of *Lecture Notes in Computer Science*, Springer, 2006.

[50] S. Brin and L. Page, *The anatomy of a large-scale hypertextual Web search engine*, volume 30 of *Computer networks and ISDN systems*, Elsevier, 1998.

[51] R. Abbasi, S. Staab, and P. Cimiano, *Organizing resources on tagging systems using t-org*, volume 2 of *Bridging the Gap between Semantic Web and Web*, 2007.

[52] C. M. Au Yeung, N. Gibbins, and N. Shadbolt, Tag meaning disambiguation through analysis of tripartite structure of folksonomies, in *Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences*, pages 3–6, 2007.

[53] C. M. Au Yeung, N. Gibbins, and N. Shadbolt, A k-nearest-neighbour method for classifying web search results with data in folksonomies, in *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference*, volume 1, pages 70–76, 2008.

[54] L. Specia and E. Motta, *Integrating folksonomies with the semantic web*, volume 4519 of *Lecture Notes in Computer Science*, Springer, 2007.

[55] R. Jäschke, A. Hotho, C. Schmitz, B. Ganter, and G. Stumme, *Discovering shared conceptualizations in folksonomies*, volume 6 of *Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, 2008.

[56] R. Jaschke, A. Hotho, C. Schmitz, and G. Stumme, *Analysis of the publication sharing behaviour in BibSonomy*, volume 4604 of *Lecture Notes in Computer Science*, Springer, 2007.

[57] W. H. Chen, Y. Cai, and H. F. Leung, An unsupervised method of exploring ontologies from folksonomies, in *Proceedings of The 10th International Conference on Computational Science and Its Applications*, IEEE Computer Society, 2010.

[58] J. Schultz and M. Liberman, Topic detection and tracking using idf-weighted cosine coefficient, in *Broadcast News Workshop'99 Proceedings*, page 189, 1999.

[59] W. H. Chen, Y. Cai, H. F. Leung, and Q. Li, Generating ontologies with basic level concepts from folksonomies, in *Proceedings of 10th International Conference on Computational Science*, Elsevier, 2010.

[60] W. H. Chen, Y. Cai, H. F. Leung, and Q. Li, Context-aware basic level concepts detection in folksonomies, in *Proceedings of the 11th International Conference on Web-Age Information Management*, Springer Verlag, 2010.

[61] N. Stojanovic, R. Studer, and L. Stojanovic, *An approach for the ranking of query results in the semantic web*, Lecture Notes in Computer Science, Springer, 2003.

[62] B. Cohen and G. Murphy, *Models of concepts*, volume 8 of *Cognitive Science*, Lawrence Earlbaum, 1984.

[63] Y. Cai and H. F. Leung, A Formal Model of Fuzzy Ontology with Property Hierarchy and Object Membership, in *Proceedings of the 27th International Conference on Conceptual Modeling*, page 82, Springer, 2008.