Automatic Generation of Ontology for Scholarly Semantic Web

Thanh Tho Quan[1], Siu Cheung Hui[1], A.C.M. Fong[1], and Tru Hoang Cao[2]

[1] School of Computer Engineering, Nanyang Technological University
Singapore
{PA0218164B, asschui, ascmfong}@ntu.edu.sg
[2] Faculty of Information Technology, Hochiminh City University of Technology
Vietnam
tru@dit.hcmut.edu.vn

Abstract. Semantic Web provides a knowledge-based environment that enables information to be shared and retrieved effectively. In this research, we propose the Scholarly Semantic Web for the sharing, reuse and management of scholarly information. To support the Scholarly Semantic Web, we need to construct ontology from data which is a tedious and difficult task. To generate ontology automatically, Formal Concept Analysis (FCA) is an effective technique that can formally abstract data as conceptual structures. To enable FCA to deal with uncertainty in data and interpret the concept hierarchy reasonably, we propose to incorporate fuzzy logic into FCA for automatic generation of ontology. The proposed new framework is known as Fuzzy Formal Concept Analysis (FFCA). In this paper, we will discuss the Scholarly Semantic Web, and the ontology generation process from the FFCA framework. In addition, the performance of the FFCA framework for ontology generation will also be evaluated and presented.

1 Introduction

Semantic Web was introduced as a common framework that allows data to be shared and reused across application, enterprise and community boundaries[1]. And ontology is used to represent knowledge on the Semantic Web. Ontology is a conceptualization of a domain into a human understandable, but machine-readable format consisting of entities, attributes, relationships and axioms[2]. Thus, the knowledge metadata designed in the Semantic Web using ontologies should be sufficiently expressive to represent and model the domain it applies to. As such, programs can then access the knowledge carried by the Semantic Web and use the knowledge for processing information in a semantic manner.

Recently, much research has investigated the use of ontology to represent data. As the source data is usually stored in unstructured, semi-structured or fully structured format (e.g. textual documents or database schemata), it needs to be processed in order to generate the ontology in an appropriate format for representation. Some tools such as Protégé 2000[3] and OLIEd[4] have been

S.A. McIlraith et al. (Eds.): ISWC 2004, LNCS 3298, pp. 726–740, 2004.

developed to help users to edit ontology. However, it is a very difficult and cumbersome task to manually derive ontology from data. Some recent researches have been carried out to tackle this problem through the learning of ontology from free text[5], semi-structured data (e.g., HTML or XML) or structured data from a database[6].

To represent conceptual organization of the corresponding context of data, concepts are usually organized into multiple levels as a hierarchy, in which concepts at lower levels are more specific in terms of meaning than those at higher levels. Generally, to generate ontology from a database automatically, we need to perform the following two steps: (1) to abstract data items as ontology concepts and (2) to construct relations between concepts. Ontology concepts can be extracted quite efficiently from free text documents. However, it remains a hard problem to generate ontology relations automatically due to the complexity of Natural Language Processing (NLP). Relations represented in semi-structured and structured data can be extracted as ontology relations. This can be done using data mining techniques such as association rules mining and clustering[5,7].

Conceptual clustering techniques such as COBWEB[7] were proposed to discover knowledge that is more meaningful and comprehensible. COBWEB can cluster data into conceptual clusters, which are clusters associated with conceptual descriptions. The generated clusters can then be organized as a concept hierarchy. However, as COBWEB uses statistical models to describe clusters conceptually, it is unable to give the "real" conceptual organization, in terms of descriptions and relations. Moreover, since COBWEB is based on the hierarchical clustering technique to generate clusters of hierarchical relations, the concept hierarchy generated has a tree-like form. It means that a subconcept is only inherited from one superconcept. This way of conceptual representation cannot really reflect real-life concept organization. For instance, "fuzzy clustering technique" can be considered as a concept inherited from two superconcepts, "clustering technique" and "fuzzy theory".

Formal Concept Analysis (FCA)[8], which is a data analysis technique based on ordered lattice theory, has been used for conceptual knowledge discovery[9]. FCA offers better conceptual representations compared with traditional conceptual clustering techniques such as COBWEB as it provides formal definitions of concepts and its hierarchical relationships of concepts are organized as a lattice rather than a simple hierarchical tree. However, this technique still suffers from the drawback on its in capabilities on representing vague information and extracting "real" concepts. In this paper, we propose a new formal framework known as Fuzzy Formal Concept Analysis (FFCA) for constructing ontology for Scholarly Semantic Web that supports citation-based retrieval of scientific publications for the Semantic Web. The FFCA framework extends Formal Concept Analysis with fuzzy logic[10] which can represent vague and uncertain information. In addition, we also propose a conceptual clustering technique based on the FFCA framework to construct ontology of "real" concepts.

The rest of this paper is organized as follows. Section 2 discusses the Scholarly Semantic Web. Section 3 discusses the Fuzzy Formal Concept Analysis

framework. Section 4 discusses conceptual clustering based on FFCA. Ontology generation is given in Section 5. The ontology generation process using an experimental citation database is discussed in Section 6. The performance results are given in Section 7. Finally, Section 8 concludes the paper.

2 Scholarly Semantic Web

Scholars are defined as individuals working in scientific areas. They could be researchers, scientists or academics. Obviously, scholars always need to acquire new information related to their academic activities in order to support their work and research. Such information will help researchers to obtain useful knowledge in their research areas, and enable them to make contributions in their scholarly activities, such as publishing papers or attending conferences.

The enormous growth of the Internet in recent years has urged scholars to use the Web for conducting scientific research. Digital libraries[11] and in particular, citation-based retrieval systems such as ISI (Institute for Scientific Information)[12] and CiteSeer (or Research Index)[13] are some of the tools that have been developed to help researchers to search related scientific information over the Web. In scientific documents, other papers or books that are useful for the understanding of its contents are usually cited as references. Citation indices contain references that the documents cite. They provide linking between source documents to the cited documents or papers. Thus, citation indices provide useful information to help researchers when conducting scientific research, such as identifying researchers working on their research areas, finding publications from a certain research area, and analyzing research trends. This helps researchers avoid doing research that had already been done by others. Citation indices are stored in a citation database.

Due to the lack of semantic information in the traditional Web, researchers can only rely on traditional search or retrieval systems to retrieve scientific publications. To provide more advanced and semantic-based search, research has recently been carried out based on Semantic Web for scholarly activities. The E-scholar Knowledge Inference Model (ESKIMO)[14] investigates the use of hypertext links on the traditional Web to develop a scholarly system for the Semantic Web. Using the Semantic Web, ESKIMO can retrieve scholarly information, such as finding researchers and institutions that are related to a particular document, or finding similar journals. However, in ESKIMO, the ontology building approach is still largely based on manual methods.

In this research, we aim to develop a Scholarly Semantic Web to support scholarly activities based on citation databases. Figure 1 gives the architecture of the Scholarly Semantic Web. As shown in Figure 1, scientific publications stored in the citation database are used as the source for automatic construction of ontology. The proposed FFCA technique is then applied to the citation database to construct the Scholarly Ontology. Web Services can be provided to enable the knowledge stored in the Scholarly Ontology be accessible by other programs. Semantic Search Engine can also be developed to support seman-

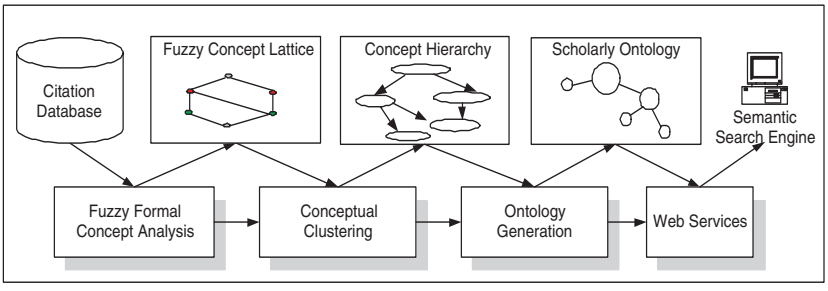

Fig. 1. The proposed approach for automatic generation of concept hierarchy

tic search functions to locate information on the Scholarly Ontology. As such, knowledge stored in the Scholarly Ontology can be shared and retrieved. The major components of the Scholarly Semantic Web are briefly discussed below:

- Fuzzy Formal Concept Analysis: It generates *fuzzy formal context* from the citation database. Citation information extracted from scientific documents is used to generate the necessary fuzzy information in order to construct the fuzzy formal context. In addition, it also generates *fuzzy formal concepts* from the *fuzzy formal context* and organizes the generated concepts as a *fuzzy concept lattice*.
- Conceptual Clustering: It clusters concepts on the *fuzzy concept lattice* and generates a *concept hierarchy*.
- Ontology Generation: It generates the Scholarly Ontology from the *concept hierarchy*.
- Web Services: It enables other programs to access the knowledge stored in the Scholarly Ontology. In addition, advanced search features can also be provided.
- Semantic Search Engine: It enables users to query the ontology stored in the Scholarly Ontology.

This paper focuses only on the ontology generation process that involves components including Fuzzy Formal Concept Analysis, Conceptual Clustering and Ontology Generation. Web Services and Semantic Search Engine will not be discussed in this paper.

3 Fuzzy Formal Concept Analysis

In this section, we propose the Fuzzy Formal Concept Analysis, which incorporates fuzzy logic into Formal Concept Analysis, to represent vague information.

Definition 1. *A fuzzy formal context is a triple* $K = (G, M, I = \varphi(G \times M))$ *where* G *is a set of objects,* M *is a set of attributes, and* I *is a fuzzy set on*

domain $G \times M$. *Each relation* $(g, m) \in I$ *has a membership value* $\mu(g, m)$ *in* $[0, 1]$.

A fuzzy formal context can also be represented as a cross-table as shown in Table 1. The context has three objects representing three documents, namely $D1$, $D2$ and $D3$. In addition, it also has three attributes, "**D**ata Mining" (D), "**C**lustering" (C) and "**F**uzzy Logic" (F) representing three research topics. The relationship between an object and an attribute is represented by a membership value between 0 and 1.The membership values can be generated from linguistic variables assigned by experts[15] or computed automatically (to be discussed in Section 6).

Table 1. A cross-table of a fuzzy formal context.

	D	**C**	**F**
D1	0.8	0.12	0.61
D2	0.9	0.85	0.13
D3	0.1	0.14	0.87

Table 2. Fuzzy formal context in Table 1 with $T = 0.5$.

	D	**C**	**F**
D1	0.8	-	0.61
D2	0.9	0.85	-
D3	-	-	0.87

A *confidence threshold* T can be set to eliminate relations that have low membership values. Table 2 shows the cross-table of the fuzzy formal context given in Table 1 with $T = 0.5$.

Generally, we can consider the attributes of a formal concept as the description of the concept. Thus, the relationships between the object and the concept should be the intersection of the relationships between the objects and the attributes of the concept. Since each relationship between the object and an attribute is represented as a membership value in fuzzy formal context, then the intersection of these membership values should be the minimum of these membership values, according to fuzzy theory[16].

Definition 2. *Given a fuzzy formal context* K*=(*G*,* M*,* I*) and a confidence threshold* T*, we define* A^**=* $\{m \in M | \forall g \in A$*:* μ*(*g*,* m*)* $\geq T\}$ *for* $A \subseteq G$ *and* B^**=* $\{g \in G | \forall m \in B$*:* μ*(*g*,*m*)* $\geq T\}$ *for* $B \subseteq M$*. A fuzzy formal concept (or fuzzy concept) of a fuzzy formal context (*G*,* M*,* I*) with a confidence threshold* T *is a pair* $(A_f$ *=*φ*(*A*),* B*) where* $A \subseteq G$*,* $B \subseteq M$*,* $A^* = B$ *and* $B^* = A$*. Each object* $g \in \varphi(A)$ *has a membership* μ_g *defined as*

$$\mu_g = \min_{m \in B} \mu(g, m)$$

where $\mu(g, m)$ *is the membership value between object* g *and attribute* m*, which is defined in* I*. Note that if* B*=* $\{\}$ *then* $\mu_g = 1$ *for every* g*.*

Definition 3. *Let* (A_1, B_1) *and* (A_2, B_2) *be two fuzzy concepts of a fuzzy formal context* (G, M, I)*.* $(\varphi(A_1), B_1)$ *is the subconcept of* $(\varphi(A_2), B_2)$*, denoted as* $(\varphi(A_1), B_1) \leq (\varphi(A_2), B_2)$*, if and only if* $\varphi(A_1) \subseteq \varphi(A_2) (\Leftrightarrow B_2 \subseteq B_1)$*. Equivalently,* (A_2, B_2) *is the superconcept of* (A_1, B_1)*.*

Definition 4. *A fuzzy concept lattice of a fuzzy formal context K with a confidence threshold T is a set $F(K)$ of all fuzzy concepts of K with the partial order $\leq$ with the confidence threshold T.*

Definition 5. *The similarity of a fuzzy formal concept $K_1 = (\varphi(A_1), B_1)$ and its subconcept $K_2 = (\varphi(A_2), B_2)$ is defined as* $E(K_1, K_2) = \frac{|\varphi(A_1) \cap \varphi(A_2)|}{|\varphi(A_1) \cup \varphi(A_2)|}$.

Figure 2 gives the traditional concept lattice generated from Table 1. Figure 3 gives the fuzzy concept lattice generated from the fuzzy formal context given in Table 2. As shown from the figures, the fuzzy concept lattice can provide additional information, such as membership values of objects in each fuzzy formal concept and similarities of fuzzy formal concepts, that are important for the construction of concept hierarchy.

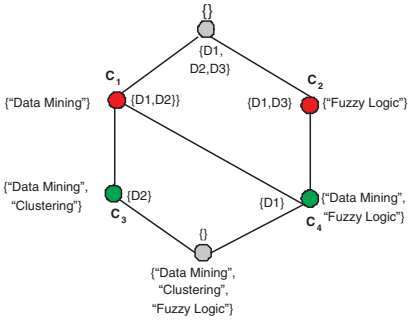

Fig. 2. A concept lattice generated from traditional FCA.

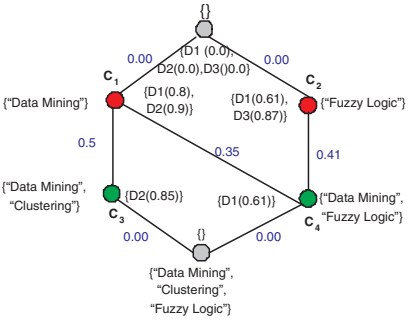

Fig. 3. A fuzzy concept lattice generated from FFCA.

4 Conceptual Clustering

As in traditional concept lattice, the formal concepts are generated mathematically, objects that have small differences in terms of attribute values are classified into distinct formal concepts. At a higher level, such objects should belong to the same concept when they are interpreted by human. Based on this observation, we propose to cluster formal concepts into conceptual clusters using fuzzy conceptual clustering. The conceptual clusters generated have the following properties:

- Conceptual clusters have hierarchical relationships that can be derived from fuzzy formal concepts on the fuzzy concept lattice. That is, a concept represented by a conceptual cluster can be a subconcept or superconcept of other concepts represented by other conceptual clusters.
- A formal concept must belong to at least one conceptual cluster, but it can also belong to more than one conceptual cluster. This property is derived from the characteristic of concepts that an object can belong to more than

one concept. For example, a scientific document can belong to more than one research area.

Conceptual clusters are generated based on the premise that if a formal concept A belongs to a conceptual cluster R, then its subconcept B also belongs to R if B is similar to A. We can use a *similarity confidence threshold* T_S to determine whether two concepts are similar or not.

Definition 6. *A conceptual cluster of a concept lattice K with a similarity confidence threshold T_S is a sublattice S_K of K which has the following properties:*

1. *S_K has a supremum concept C_S that is not similar to any of its superconcepts.*
2. *Any concept $C \neq C_S$ in S_K must have at least one superconcept $C' \in S_K$ such that $E(C, C') > T_S$.*

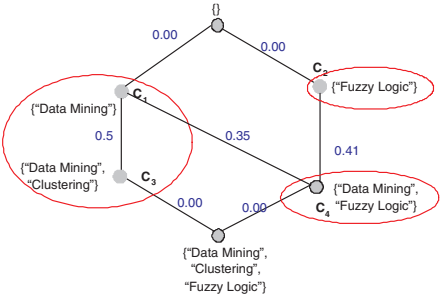

Fig. 4. Conceptual clusters.

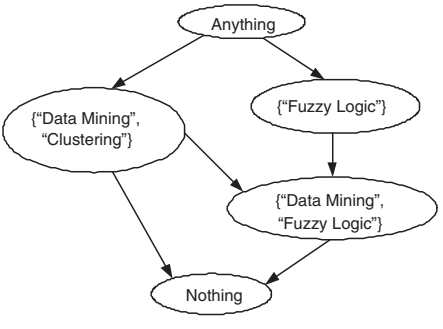

Fig. 5. Concept hierarchy.

Figure 4 shows the conceptual clusters that are generated from the concept lattice given in Figure 3 with the similarity confidence threshold $T_S = 0.5$. Figure 5 shows the corresponding concept hierarchy, in which each concept is represented by a set of attributes of objects from the corresponding conceptual cluster.

Figure 6 gives the algorithm that generates conceptual clusters from a concept C_S which is called the *starting concept* on a fuzzy concept lattice $F(K)$. To generate all conceptual clusters of $F(K)$, we choose C_S as the supremum of $F(K)$, or $C_S = \sup(F(K))$.

5 Ontology Generation

After the construction of the concept hierarchy, we need to convert it into ontology for the Semantic Web. Since ontology strongly supports hierarchical representation, the conversion is done as follows:

Algorithm: Conceptual_Cluster_Generation

Input: Starting concept C_S of concept lattice $F(K)$ and a similarity threshold T_S
Output: A set of generated conceptual clusters S_C
Process:

```
1:  S_C ← {}
2:  F'(K) ← An empty concept lattice
3:  Add C_S to F'(K)
4:  for each subconcept C' of C_S in F(K) do
5:     F'(C') ← Conceptual_Cluster_Generation(C', F(K), T_S)
6:     if E(C_S, C') = |C_S ∩ C'| / |C_S ∪ C'| < T_S then
7:        S_C ← S_C ∪ { F'(C')}
8:     else
9:        Insert F'(C') to F'(K) with sup(F'(K)) as a subconcept of C_S
10:    endif
11: endfor
12:  S_C ← S_C ∪ {F'(K)}
```

Fig. 6. The fuzzy conceptual clustering algorithm.

- Each concept in the hierarchy is represented as an ontology class.
- The concepts' relations are preserved for the corresponding generated ontology classes. That is, if S_1 is the superconcept of S_2, then C_1 is the superclass of C_2 where C_1 and C_2 are the corresponding classes for S_1 and S_2 respectively.
- Each attribute of a concept is represented as a property of the corresponding class.
- Each object in a concept is represented as an instance of the corresponding class.
- The value of an instance's property is the membership value of the corresponding object's attribute. In future research, we will apply fuzzy logic to convert the membership value into linguistic terms, so that the generated ontology will be more informative and comprehensible.

We use DMAL+OIL[17] to annotate the generated ontology. DMAL+OIL is an RDF-based ontology description language that can represent ontology class properties and relations effectively. For illustration, Figure 7 gives the ontology for the concept hierarchy given in Figure 5.

6 Scholarly Ontology for Scholarly Semantic Web

In order for evaluating the proposed Fuzzy Formal Concept Analysis framework for ontology generation for the Scholarly Semantic Web, we have collected a set of 1400 scientific documents on the research area "Information Retrieval" published in 1987-1997 from the Institute for Scientific Information's (ISI) web-site

```
<rdf:RDF>
xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
xmlns:daml="http://www.w3.org/2001/10/daml+oil#"
<daml:Ontology rdf:about="Scholarly Information">
<daml:versionInfo>
$Id: daml+oil-ex.daml,v 1.8 2001/03/27 21:24:04 horrocks Exp $
< /daml:versionInfo>
<rdfs:comment>
An ontology of Scholarly Information
< /rdfs:comment>
<daml:imports rdf:resource="http://www.w3.org/2001/10/daml+oil"/>
<daml:Class rdf:ID="Concept1">
<dmal:label> "Data Mining"</daml:label>
< /daml:Class>
<daml:Class rdf:ID="Concept2">
<dmal:label> Fuzzy Logic</daml:label>
< /daml:Class>
<daml:Class rdf:ID="Concept3">
<dmal:label> "Data Mining, Fuzzy Logic"</daml:label>
<rdfs:subClassOf rdf:resource="#Concept1"/>
<rdfs:subClassOf rdf:resource="#Concept2"/>
< /daml:Class>
<document rdf:ID = "Document3">
<instanceOf>
<resourceRef xlink:href="#Concept2"/>
< /instanceOf>
<FuzzyLogic>
<FuzzyLogicValue>0.87</FuzzyLogicValue>
< /FuzzyLogic>
< /document>
< /daml:Ontology>
< /rdf:RDF>
```

Fig. 7. The ontology for the concept hierarchy given in Figure 5.

[11]. The downloaded documents are preprocessed to extract related information such as the title, authors, citation keywords, and other citation information. The extracted information is then stored as a citation database. We then apply FFCA, conceptual clustering and ontology generation to the citation database as follows.

For each document, we have extracted the 10 most frequent citation keywords. We then construct a fuzzy formal context $K_f = \{G,M,I\}$, with G as the set of documents and M as the set of keywords. The membership value of a document D on a citation keyword C_K in K_f is computed as

$$\mu(D, C_K) = \frac{n_1}{n_2}$$

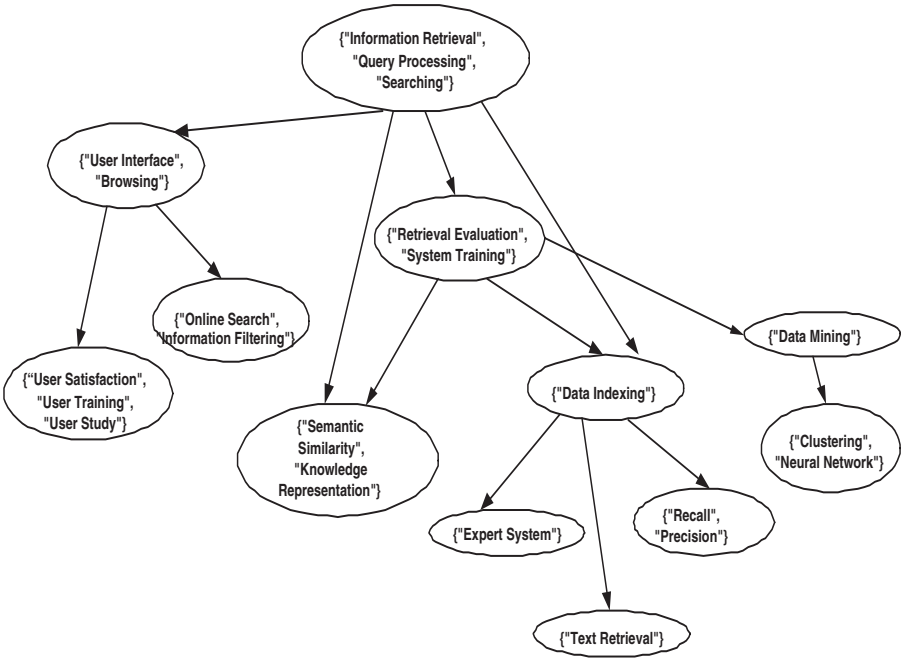

Fig. 8. An example concept hierarchy generated

where n_1 is the number of documents that cited D and contained C_K and n_2 is the number of documents that cited D. This formula is based on the premise that the more frequent a keyword occurs in the citing paper, the more important the keyword is in the cited paper.

Then, conceptual clustering is performed from the fuzzy formal context. As each conceptual cluster represents a real concept, the concepts to be discovered are the research areas. They form a hierarchy of research areas from the main research area on "Information Retrieval".

Figure 8 shows a part of the generated research hierarchy when $T_S = 0.7$. As shown in Figure 8, each research area is represented by a set of most frequent keywords occurring in the documents that belong to that research area. Research areas given by the hierarchy are considered as sub-areas of the research area "Information Retrieval". Hierarchical relationships between research areas including sub-areas and super-areas, which correspond to the definitions of subconcept and superconcept in FFCA, are also given. According to FFCA, sub-areas inherit keywords from their super-areas. Note that the inherited keywords are not shown in Figure 8 when labeling the concepts. Only keywords specific to the concepts are used for labeling.

Then, the generated concept hierarchy is converted into the Scholarly Ontology as discussed in Section 5.

7 Performance Evaluation

To evaluate the performance of FFCA for ontology generation, we can measure the performance of the conceptual clustering technique from FFCA, as it constructs the concept hierarchy for generating the Scholarly Ontology. To do this, we perform the evaluation as follows. First, we use the *relaxation error*[18] and the corresponding cluster goodness measure to evaluate the goodness of the conceptual clusters generated. Typically, the lower the relaxation error is, the better the conceptual clusters are generated. We also show whether the use of fuzzy membership instead of crisp value can help improve cluster goodness. Then, we use the *Average Uninterpolated Precision* (AUP)[19], which is a typical measure for evaluating a hierarchical construct, to evaluate the goodness of the generated concept hierarchy.

7.1 Evaluation Using Relaxation Error

Relaxation error implies dissimilarities of items in a cluster based on attributes' values. Since conceptual clustering techniques typically use a set of attributes for concept generation, relaxation error is quite a commonly used measure for evaluating the goodness of conceptual clusters.

The relaxation error RE of a cluster C is defined as

$$RE(C) = \sum_{a \in A} \sum_{i=1}^{n} \sum_{j=1}^{n} P(x_i) P(x_j) d^a(x_i, x_j)$$

where A is the set of attributes of items in C, $P(x_i)$ is the probability of item x_i occurring in C and $d^a(x_i,x_j)$ is the distance of x_i and x_j on attribute a. In this experiment, $d^a(x_i,x_j) = |m(i,a) - m(j,a)|$ where $m(i,a)$ and $m(j,a)$ are the membership values of objects x_i and x_j on attribute a respectively. The cluster goodness G of cluster C is defined as

$$G(C) = 1 - RE(C)$$

Obviously, the smaller the cluster relaxation error is, the better the cluster goodness is.

The relaxation error and the cluster goodness measure reflect respectively the dissimilarities and similarities of items in clusters. If we use a crisp number (that is, only values of 0 and 1 are used) for items' attributes as in typical FCA methods, we are unable to represent the similarities between items. Instead, the fuzzy membership value can be used to represent such similarities, thereby improving the cluster goodness.

In the experiment, we first measure the *fuzzy cluster goodness* (FCG) of clusters generated using the objects' fuzzy memberships. Then, we replace the objects' fuzzy memberships by crisp values. This is done as follows. If the membership value is greater than 0.5, it is replaced by 1, otherwise it is replaced by 0. Then, we measure the *crisp cluster goodness* (CCG) using the replaced crisp

Table 3. Performance results for fuzzy cluster goodness and crisp cluster goodness.

		T_s=0.2	T_s=0.3	T_s=0.4	T_s=0.5	T_s=0.6	T_s=0.7	T_s=0.8	T_s=0.9
N=2	FCG	0.85	0.77	0.79	0.85	0.86	0.86	0.84	0.85
	CCG	0.7	0.7	0.7	0.7	0.67	0.77	0.76	0.76
N=3	FCG	0.84	0.91	0.81	0.81	0.72	0.88	0.75	0.85
	CCG	0.72	0.72	0.72	0.71	0.65	0.71	0.69	0.69
N=4	FCG	0.9	0.74	0.82	0.76	0.75	0.81	0.73	0.75
	CCG	0.73	0.65	0.63	0.62	0.62	0.65	0.64	0.65
N=5	FCG	0.86	0.78	.075	0.8	0.76	0.75	0.72	0.77
	CCG	0.71	0.63	0.61	0.61	0.61	0.63	0.61	0.61
N=6	FCG	0.91	0.76	0.79	0.76	0.7	0.74	0.8	0.8
	CCG	0.73	0.65	0.61	0.61	0.6	0.62	0.63	0.63
N=7	FCG	0.91	0.79	0.73	0.73	0.73	0.84	0.78	0.74
	CCG	0.76	0.66	0.64	0.63	0.62	0.63	0.65	0.65
N=8	FCG	0.95	0.81	0.73	0.84	0.77	0.82	0.84	0.77
	CCG	0.8	0.71	0.64	0.63	0.62	0.63	0.65	0.65
N=9	FCG	0.91	0.88	0.76	0.84	0.71	0.75	0.77	0.85
	CCG	0.84	0.75	0.66	0.63	0.63	0.65	0.65	0.65
N=10	FCG	0.94	0.85	0.83	0.73	0.74	0.84	0.84	0.73
	CCG	0.85	0.77	0.65	0.64	0.63	0.64	0.66	0.66

values. We vary the number of keywords N extracted from documents from 2 to 10 and the similarity threshold T_S from 0.2 to 0.9 for conceptual clustering. The results are given in Table 3.

As can be seen from Table 3, the FCG obtained is generally better than the CCG. It has shown the advantage of using fuzzy membership values for representing object attributes. Since the relaxation error is an important factor for approximate query answering[20], the experimental results have shown that the use of fuzzy logic in FFCA can potentially improve retrieval performance on conceptual clustering.

In addition, the experimental results have also shown that better cluster goodness is obtained when the number of extracted keywords is small. It is expected because smaller number of keywords will cause smaller differences in objects in terms of keywords' membership values. Therefore, the relaxation error will be smaller. However, as we will see later in Section 7.2, smaller number of extracted keywords will cause poor performance in retrieval.

7.2 Evaluation Using Average Uninterpolated Precision

The Average Uninterpolated Precision (AUP) is defined as the sum of the precision value at each point (or node) in a hierarchical structure where a relevant item appears, divided by the total number of relevant items. Typically, AUP implies the goodness of a hierarchical structure.

We have manually classified the downloaded documents into classes based on their research themes. For each class, we extract 5 most frequent keywords

Table 4. Performance results for AUP^H and AUP^U.

		$\mathbf{T_s}$**=0.2**	$\mathbf{T_s}$**=0.3**	$\mathbf{T_s}$**=0.4**	$\mathbf{T_s}$**=0.5**	$\mathbf{T_s}$**=0.6**	$\mathbf{T_s}$**=0.7**	$\mathbf{T_s}$**=0.8**	$\mathbf{T_s}$**=0.9**
N=2	AUP^H	0.0503	0.0503	0.0503	0.05	0.042	0.043	0.0467	0.0464
	AUP^U	0.0325	0.0325	0.0325	0.0325	0.0251	0.0175	0.0175	0.0175
N=3	AUP^H	0.1088	0.1088	0.1083	0.1076	0.0849	0.0763	0.0818	0.0831
	AUP^U	0.0721	0.0721	0.0721	0.0721	0.0492	0.0305	0.0292	0.0292
N=4	AUP^H	0.1609	0.1552	0.1452	0.1418	0.1373	0.1338	0.1404	0.1461
	AUP^U	0.1458	0.1166	0.1029	0.0985	0.0855	0.0655	0.0636	0.0636
N=5	AUP^H	0.2066	0.1871	0.179	0.1779	0.1761	0.1772	0.1876	0.1965
	AUP^U	0.1924	0.1555	0.1448	0.1429	0.1300	0.1076	0.1065	0.1065
N=6	AUP^H	0.2983	0.2688	0.2625	0.2696	0.2733	0.2737	0.2833	0.2938
	AUP^U	0.2846	0.2424	0.2263	0.2237	0.2077	0.1819	0.1792	0.1797
N=7	AUP^H	0.3534	0.328	0.3149	0.3159	0.3295	0.2737	0.2833	0.2938
	AUP^U	0.352	0.2985	0.2677	0.2651	0.2556	0.2431	0.2316	0.2309
N=8	AUP^H	0.3619	0.3368	0.3215	0.3212	0.3306	0.3366	0.3417	0.3605
	AUP^U	0.36	0.3	0.2687	0.2659	0.2556	0.2422	0.2303	0.2297
N=9	AUP^H	0.3723	0.3459	0.3296	0.3267	0.3353	0.341	0.346	0.3657
	AUP^U	0.3704	0.3114	0.2771	0.2741	0.2635	0.2498	0.2377	0.237
N=10	AUP^H	0.3768	0.351	0.3345	0.3294	0.3381	0.3439	0.3485	0.3686
	AUP^U	0.3747	0.317	0.2824	0.2793	0.2685	0.2548	0.2426	0.242

from the documents in the class. Then, we use these keywords as inputs to form retrieval queries and evaluate the retrieval performance using AUP. This is carried out as follows. For each document, we will generate a set of *document keywords*. There are two different ways to generate document keywords. The first way is to use the set of keywords, known as *attribute keywords*, from each conceptual cluster as the document keywords. The second way is to use the keywords from each document as the document keywords. Then, we vectorize the document keywords and the input query, and calculate the vectors' distance for measuring the retrieval performance.

We refer the AUP measured using the first way to as *Hierarchical Average Uninterpolated Precision* (AUP^H), as each concept inherits attribute keywords from its superconcepts. Whereas the AUP measured using the second way is referred to as *Unconnected Average Uninterpolated Precision* (AUP^U). Table 4 gives the performance results for $\text{AUP}^H$and AUP^Uusing different numbers of extracted keywords N and similarity thresholds T_S for conceptual clustering. Generally, when N gets larger, the performance on both $\text{AUP}^H$and AUP^U gets better. It has shown that the number of keywords extracted for conceptual clustering has affected the retrieval accuracy. In addition, the performance on AUP^H is generally better than that of AUP^U. It means that the attribute keywords generated for conceptual clusters are quite appropriate concepts to be represented in the hierarchical structure.

8 Conclusions

In this paper, we have proposed a new framework called Fuzzy Formal Concept Analysis that extends Formal Concept Analysis for ontology generation for the Scholarly Semantic Web. As compared to traditional FCA, FFCA can deal with uncertainty in data. Moreover, we have also proposed a conceptual clustering technique that can cluster the fuzzy concept lattice to generate a concept hierarchy, which can then be converted into Scholarly Ontology. The proposed framework is used to construct Semantic Scholarly Web from a citation database. The FFCA framework has been evaluated and good performance has been achieved. As for future research, we intend to restructure the generated ontology for supporting multiple-level concepts corresponding to multiple levels of abstractions. Moreover, we will also use fuzzy logic to convert fuzzy property values of the generated ontology into linguistic terms to make the generated ontology more informative and comprehensible.

References

1. T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web", *Scientific American,* Available at: <http://www.sciam.com/2001/0501issue/0501berners-lee.html>, 2001.
2. N. Guarino and P. Giaretta, *Ontologies and Knowledge Bases: Towards a Terminological Clarification. Toward Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, IOS Press, Amsterdam, 1995.
3. N. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", *Report SMI-2001-0880*, Department of Mechanical and Industrial Engineering, University of Toronto, 2001.
4. S. Bechhofer, I. Horrocks, P. Patel-Schneider and S. Tessaris, "A Proposal for a Description Logic Interface", In *Proceedings of the International Workshop on Description Logics*, 1999, pp. 33-36.
5. A. Maedche and S. Staab, "Mining Ontologies from Text", In *EKAW-2000 - 12th International Conference on Knowledge Engineering and Knowledge Management,* Juan-les-Pins, France. LNAI, Springer, 2000.
6. A. Doan, J. Madhavan, P. Domingos and A.Y. Halevy, "Learning to Map Between Ontologies on the Semantic Web", In *Proceedings of the Eleventh International World Wide Web Conference, WWW2002*, Honolulu, Hawaii, USA, 2002, pp. 662-673.
7. A.K Jain and R.C Dubes, *Algorithms for Clustering Data*, Prentice-Hall, 1988.
8. B. Ganter and R. Wille, *Formal Concept Analysis*, Springer, Berlin – Heidelberg, 1999.
9. A. Hotho, S. Staab and G. Stumme, "Explaining Text Clustering Results using Semantic Structures", In *Proceedings of Principles of Data Mining and Knowledge Discovery, 7th European Conference, PKDD 2003*, Croatia, 2003, pp. 217-228.
10. L.A. Zadeh, "Fuzzy Logic and Approximate Reasoning", *Synthese*, Vol. 30, 1975, pp. 407-428.
11. G. Wiederhold, *Digital Libraries, Value, and Productivity*, Communications of the ACM, Vol. 38, No. 4, 1995, pp. 85-96.

12. ISI, *Institute for Scientific Information*, Available at: <http://www.isinet.com>, 2000.
13. K. Bollacker, S. Lawrence and C. Giles, "Discovering Relevant Scientific Literature on the Web", *IEEE Intelligent Systems*, Vol. 15, No. 2, 2000, pp. 42-47.
14. S. Kampa, T. Miles-Board and L. Carr, "Hypertext in the Semantic Web", In *Proceedings ACM Conference on Hypertext and Hypermedia*, Aarhus, Denmark, 2001, pp. 237-238.
15. L.A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning". Part I: *Inf. Science 8,*199-249; Part II: *Inf. Science 8,* 301-357; Part III: *Inf. Science 9,* 43-80, 1975.
16. L.A. Zadeh, "Fuzzy Sets", *Journal of Information and Control*, Vol. 8, 1965, pp. 338-353.
17. DARPA, *DAML-ONT Initial Release,* Available at: <http://www.daml.org/2000/10/daml-ont.html>, 2000.
18. W. Chu and K. Chiang, "Abstraction of High Level Concepts from Numerical Values in Databases", In *Proceedings of AAAI Workshop on Knowledge Discovery in Databases*, 1994, pp. 133-144.
19. N. Nanas, V. Uren and A. de Roeck, "Building and Applying a Concept Hierarchy Representation of a User Profile", In *Proceedings of the 26th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, 2003.
20. W. Chu and Q. Chen, "Neighborhood and associative query answering", *Journal of Intelligent Information Systems*, Vol. 1, No. 3, 1992.