Queensland University of Technology
Brisbane Australia

This may be the author's version of a work that was submitted/accepted for publication in the following source:

Li, Yuefeng & Zhong, Ning
(2006)
Mining Ontology for Automatically Acquiring Web User Information Needs.
IEEE Transactions on Knowledge and Data Engineering, *18*(4), pp. 554-568.

This file was downloaded from: https://eprints.qut.edu.au/225189/

© Consult author(s) regarding copyright matters

This work is covered by copyright. Unless the document is being made available under a Creative Commons Licence, you must assume that re-use is limited to personal use and that permission from the copyright owner must be obtained for all other uses. If the document is available under a Creative Commons License (or other specified license) then refer to the Licence for details of permitted re-use. It is a condition of access that users recognise and abide by the legal requirements associated with these rights. If you believe that this work infringes copyright please provide details by email to qut.copyright@qut.edu.au

Notice: *Please note that this document may not be the Version of Record (i.e. published version) of the work. Author manuscript versions (as Submitted for peer review or as Accepted for publication after peer review) can be identified by an absence of publisher branding and/or typeset appearance. If there is any doubt, please refer to the published source.*

https://doi.org/10.1109/TKDE.2006.1599392

Mining Ontology for Automatically Acquiring Web User Information Needs

Yuefeng Li and Ning Zhong, *Senior Member*, *IEEE*

Abstract—It is not easy to obtain the right information from the Web for a particular Web user or a group of users due to the obstacle of automatically acquiring Web user profiles. The current techniques do not provide satisfactory structures for mining Web user profiles. This paper presents a novel approach for this problem. The objective of the approach is to automatically discover ontologies from data sets in order to build complete concept models for Web user information needs. It also proposes a method for capturing evolving patterns to refine discovered ontologies. In addition, the process of assessing relevance in ontology is established. This paper provides both theoretical and experimental evaluations for the approach. The experimental results show that all objectives we expect for the approach are achievable.

Index Terms—Web intelligence, ontology mining, Web mining, Web user profiles.

1 Introduction

Over the last decade, we have witnessed an explosive growth in the data accessible on the Web. However, there are two fundamental issues regarding the effectiveness of information gathering from the Web: *mismatch* and *overload*. Mismatch means some useful and interesting data has been overlooked, whereas overload means some gathered data is not what users want.

Traditional techniques related to information retrieval (IR) have touched upon the fundamental issues [1], [8]. However, IR-based systems neither explicitly describe how the systems can act like users nor discover exotic knowledge from very large data sets to answer what users really want. This issue has challenged the artificial intelligence (AI) community to address "what has information gathering to do with AI" [14]. For a short while, many intelligent agent-based approaches have been grappling with this challenge. Unfortunately, agent-based approaches can only show us the architectures of information gathering systems. They cannot provide strategies for finding interesting and useful knowledge from data to overcome the fundamental issues.

Web intelligence (WI) [51] is a new direction which can provide a new approach to solve this problem. Currently, the application of data mining techniques to Web data, called Web mining, is used to discover patterns from data (e.g., user feedback or user log data). A Web mining system can be viewed as the use of data mining techniques to automatically retrieve, extract, generalize, and analyze Web information [3], [36]. Web mining can be classified into four categories: Web usage, Web structure, Web content, and Web user profiles [7], [28], [43].

- *Y. Li is with the School of Software Engineering and Data Communications, Queensland University of Technology, Australia. E-mail: y2.li@qut.edu.au.*
- *N. Zhong is with the Department of Systems and Information Engineering, Maebashi Institute of Technology, Japan. E-mail: zhong@maebashi-it.ac.jp.*

Manuscript received 14 June 2005; revised 13 Sept. 2005; accepted 19 Sept. 2005; published online 17 Feb. 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0232-0605.

Earlier work on accessing usage logs can be found in [3], [31], [48]. The primitive objective of Web usage mining is the discovery of Web server access patterns. It can be used to obtain some interesting patterns about user behaviors from usage logs. Web structure mining is the discovery of hypertext/linking structure patterns [39]. Web content mining is the discovery of Web document content patterns. It can be used to analyze text and graphic contents on the Web [2], [26]. There are two diagrams in Web user profile mining: the data diagram and information diagram. The former is the discovery of registration data and customer profile portfolios [43]. The latter is the discovery of interesting topics [24] for Web user information needs. In this paper, we present contributions for Web user profile mining in the information diagram.

Currently, a major challenge is to build communication between search engines and Web users. However, most search engines can only use queries rather than Web user profiles due to the difficulty of automatically acquiring Web user profiles. The first reason for this is that Web users may not know how to represent their topics of interest [33], [19]. The second reason is that Web users may not wish to invest a great deal of effort to dig out a few relevant pages from hundreds of thousands of candidates provided by search engines.

The simplistic approach of acquiring user profiles is to describe the profiles through term vector spaces (e.g., a set of keywords) by using machine-learning techniques [12], [40]. The main disadvantage of the simplistic approach is the poor interpretation of user profiles to the users. To obtain an explicit specification to the users, user profiles can be represented in some predefined categories [33], [20].

There are two main drawbacks in using these approaches to acquire Web user profiles. The first one is that the effectiveness largely depends on the numbers of labeled training data. However, we may only obtain some positive documents. The second one is that it is hard to distinguish noninteresting topics from interesting topics since they may have a similar representation using the above methods.

```
<topic>
    <number> 101 </number>
    <title> Economic espionage </title>
    <description> What is being done to counter economic espionage internationally?
    </description>
    <narrative> Documents which identify economic espionage cases and provide action(s)
        taken to reprimand offenders or terminate their behaviour are relevant. Economic espionage
        would encompass commercial, technical, industrial or corporate types of espionage.
        Documents about military or political espionage would be irrelevant.
    </narrative>
</topic>
```

Fig. 1. A specified topic.

These methods fail to describe correlations between concepts and, hence, have to use incomplete concept spaces for user profiles.

In this paper, we develop an ontology mining technique to overcome the above drawbacks. In the beginning, we assume that the training set only includes positive documents and that the system can discover some patterns from the training set. During the execution, the system might select a small amount of documents and require users to label them as either positive or negative (user feedback). We also assume that user interests (compound classes) can be constructed from some primitive objects (e.g., keywords).

Syntactically, an ontology in this research consists of two parts: the top backbone and the base backbone. The former illustrates the linkage between compound classes of the ontology. The latter illustrates the linkage between primitive classes and compound classes that was normally used in information retrieval [29]. The initial ontology can be automatically built according to the set of discovered patterns. A mathematical model, called the association set, is set up to represent the correlation between compound classes. The ontology can be updated based on the user feedback. We also present a novel technique for capturing evolving patterns in the ontology to refine its association set. As a result, some noninteresting topics (patterns) can be removed or uncertainties in some inadequate topics (patterns) can be weakened. In addition, we establish a method for automatically learning how to assess relevance in the ontology.

The remainder of the paper is structured as follows: We begin by describing the problem and introducing some new definitions in Section 2. In Section 3, we discuss what sort of ontology we can discover from a set of positive documents. We also present an ontology mining algorithm. In Section 4, we propose a novel technique for capturing evolving patterns. Using this technique, we can detect and decline uncertainties in some inadequate patterns and even remove some of them from the ontology. In Section 5, we present formal definitions for learning two dimensions (specificity and exhaustivity of topics) for relevance assessment. We also present a method to synthesize the two dimensions into one for the purpose of efficiency. In Section 6, we discuss our experiments to show the performance of the proposed approach for the automatic acquiring of Web user profiles. Section 7 discusses related work and the final section presents the conclusions and gives an outlook on further work.

2 Background

In this section, we first use an example to describe the problem. We also introduce some new definitions that are used throughout the remainder of the paper.

2.1 The Problem

We classify Web user profiles into two diagrams: the data diagram and information diagram. The former diagram is the discovery of interesting registration data and customer profile portfolios. In general, the registration data or customer profile portfolio can be described as a database or a set of transactions, e.g., user log data. The meaning of data values in each record (or transaction) is understandable.

The latter diagram is the discovery of interesting topics for Web user information needs. Compared to the data diagram, there are two significant differences on the data:

1. There are many duplicates in the data.
2. The meaning of data values (terms) is ambiguous since there may exist "synonymy" or "hyponymy" relations between terms.

In this paper, we present contributions for Web user profile mining in the information diagram. The difficult problem related to this research is to identify what kind of ontology can be automatically discovered from data sets to illustrate meaningful descriptions for user profiles. Usually, users themselves are easily acquainted with interesting Web pages while they read through contents of the Web pages. The rationale behind this is that the users implicitly use a concept model based on their knowledge about a specified topic; even though they do not know how to represent it.

It may be desirable to ask Web users to provide descriptions and narratives for their topics of interest while we try to represent user profiles. For example, Fig. 1 shows a specified topic used for the filtering track in 2002 TREC (Text REtrieval Conference, see http://trec.nist.gov/), where the description and the narrative of the topic were edited by linguists.

We can manually build a concept model for the topic illustrated in Fig. 1. Fig. 2 shows this model which consists of a set of subtopics and the relations between them, where a hollow arrow denotes an "is-a" relation between nodes, e.g., a *commercial espionage* is an *economic espionage*. In Fig. 2, there are four relevant subtopics: *commercial espionage*, *technical espionage*, *industrial espionage*, and *corporate espionage*; and two nonrelevant subtopics: *military espionage* and *political espionage*.

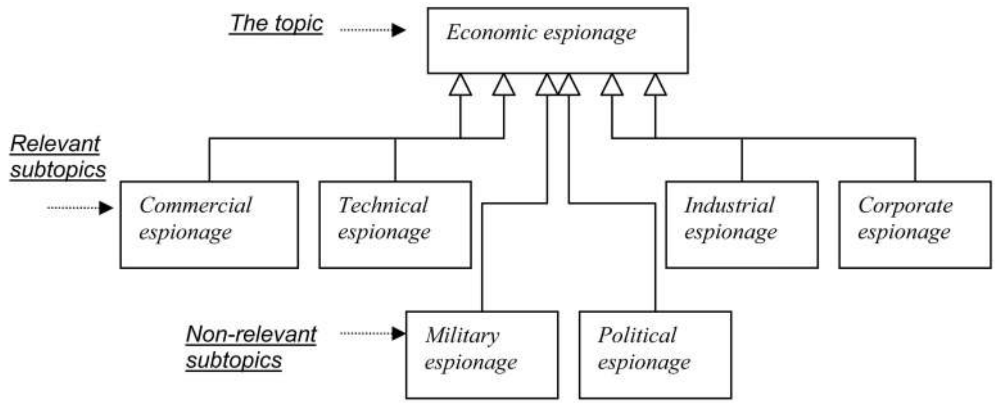

Fig. 2. An incomplete concept model.

It is difficult for general users to write adequate descriptions and narratives. Although linguists can provide tolerable descriptions and suitable narratives, the corresponding concept model is still incomplete. First, the linguists may ignore some important terms, for example, "spy" in this example. Dictionaries usually are not very useful for expanding the set of terms since we do not know authors' writing styles. Also, quite often the linguists and the dictionaries may ignore some relations between subtopics. For instance, we are not sure if there is any overlap between *technical espionage* and *industrial espionage* from Fig. 2.

In this research, we do not request users to provide descriptions and narratives; instead, we assume that the users can at least provide a set of positive documents for their topics of interest in the beginning. Table 1 depicts a set of positive documents for the specified topic in Fig. 1. The contents are extracted from the titles of original positive documents by using a basic text processing which includes case folding, stemming, and stop words and nonkeywords removal, where the set of keywords is {GERMAN, VW, US, ECONOM, SPY, BILL, ESPIONAG, MAN}.

The main objective of this research is to discover a required ontology automatically from a data set as shown in Table 1 for acquiring user profiles. We also discuss how to apply the discovered ontology to respond to what Web users want.

2.2 Definitions

Let $T = \{t_1, t_2, \ldots, t_k\}$ be a set of keywords (or terms) and D be a training set of documents, which consists of a set of positive documents, D^+; and a set of negative documents, D^-, where each document is a set of terms (may include duplicate terms). In the beginning, we let $D^- = \emptyset$.

TABLE 1
A Set of Positive Documents

Name	Content	*positive*
d_1	GERMAN VW	yes
d_2	US US ECONOM SPY	yes
d_3	US BILL ECONOM ESPIONAG	yes
d_4	US ECONOM ESPIONAG BILL	yes
d_5	GERMAN MAN VW ESPIONAG	yes
d_6	GERMAN GERMAN MAN VW SPY	yes

A set of terms is referred to as a *termset*. Given a document d (or a paragraph) and a term t, we define $tf(d,t)$ as the number of occurrences of t in d. A set of term frequency pairs, $P = \{(t,f) | t \in T, f = tf(t,d) > 0\}$, is referred to as a *pattern* in this paper. We also use *support(P)* to describe the extent to which the pattern is discussed in the training set: The greater the *support* is, the more important the pattern is.

Let $termset(P) = \{t | (t,f) \in P\}$ be the termset of P. In this paper, pattern P_1 is equal to pattern P_2 if and only if $termset(P_1) = termset(P_2)$. A pattern is uniquely determined by its termset. Two patterns should be composed if they have the same termset (or they are in a same category). In this paper, we use a composition operation, $\oplus$, to generate new patterns.

Let P_1 and P_2 be two patterns. We call $P_1 \oplus P_2$ the *composition* of P_1 and P_2 which satisfies:

$$\begin{aligned} & P_1 \oplus P_2 = \{(t, f_1 + f_2) | (t, f_1) \in P_1, (t, f_2) \in P_2\} \bigcup \\ & \{(t,f) | t \in (termset(P_1) \cup termset(P_2)) \\ & - (termset(P_1) \cap termset(P_2)), (t,f) \in P_1 \cup P_2\} \\ & support(P_1 \oplus P_2) = support(P_1) + support(P_2). \end{aligned} \tag{1}$$

We can verify that the operands of $\oplus$ are interchangeable according to the above definition.

Using the example in Table 1, six patterns can be discovered directly from the positive documents. Let Ω be the set of discovered patterns, we have $\Omega = \{P_1, P_2, P_3, P_4, P_5, P_6\}$ (see Table 2). Because

$$\begin{aligned} termset(P_3) &= termset(P_4) \\ &= \{\text{US}, \text{BILL}, \text{ECONOM}, \text{ESPIONAG}\}, \end{aligned}$$

we should compose them into a new pattern according to the above declarations. Table 2 illustrates these patterns, where $P_7 = P_3 \oplus P_4$, $\Omega = \{P_1, P_2, P_5, P_6, P_7\}$, and P_3 and P_4 can be removed since they are redundant patterns after finishing the composition operation.

Given a pattern $P = \{(t_1, f_1), (t_2, f_2), \ldots, (t_r, f_r)\}$, its *normal form* $\{(t_1, w_1), (t_2, w_2), \ldots, (t_r, w_r)\}$ can be determined using the following equations: $w_i = \frac{f_i}{\sum_1^r f_j}$ for all $i \le r$ and $i \ge 1$.

After normalization, we have $\sum_{(t,w) \in P} w = 1$. We call $(w_1, w_2, \ldots, w_r)$ the *weight distribution* of P.

TABLE 2
Pattern Examples

Name	Support	Pattern
P_1	1	{(GERMAN, 1), (VW, 1)}
P_2	1	{(US, 2), (ECONOM, 1), (SPY, 1)}
P_3	1	{(US, 1), (BILL, 1), (ECONOM, 1), (ESPIONAG, 1)}
P_4	1	{(US, 1), (ECONOM, 1), (ESPIONAG, 1), (BILL, 1)}
P_5	1	{(GERMAN, 1), (MAN, 1), (VW, 1), (ESPIONAG, 1)}
P_6	1	{(GERMAN, 2), (MAN, 1), (VW, 1), (SPY, 1)}
* P_7	2	{(US, 2), (BILL, 2), (ECONOM, 2), (ESPIONAG, 2)}

* where P_7 is generated using the composition.

TABLE 3
Patterns in the Base Backbone

Name	Support	Pattern
P_1	1	{(GERMAN, 1), (VW, 1)}
P_2	1	{(US, 2), (ECONOM, 1), (ESPIONAG, 1)}
* P_7	2	{(US, 2), (BILL, 2), (ECONOM, 2), (ESPIONAG, 2)}
* P_8	2	{(GERMAN, 3), (MAN, 2), (VW, 2), (ESPIONAG, 2)}

* where P_7 and P_8 are generated using the composition.

3 Ontology Mining Algorithm

Syntactically, we assume that topics interesting for the user are constructed from some primitive objects (e.g., terms). According to this assumption, an ontology consists of primitive classes and compound classes. The primitive classes are the smallest concepts that cannot be assembled from other classes; however, they may be inherited by some derived concepts or their children (e.g., subterms). The compound classes are the interesting topics, which can be constructed from a set of primitive classes.

The procedure of automatic ontology mining can be divided into two diagrams: the base backbone construction and the top backbone construction. The former is the construction of linkages between primitive classes and compound classes and the latter is the construction of linkages between compound classes.

Fig. 3 demonstrates a base backbone of the ontology, which organizes linkages between primitive classes and compound classes according to Table 2. The set of terms (primitive objects) now is *{VM, German, US, Econom, Espionag, Bill, Man}* because *Spy is Espionage*. We use an "is-a" link (the arrow in Fig. 3) to show the relation between the term "Espionag" and its subterm "*Spy*."

The compound objects are P_1, P_2, P_7, and P_8, where pattern

$$\begin{aligned} P_8 &= P_5 \oplus P_6 \\ &= \{(\mathrm{GERMAN}, 3), (\mathrm{MAN}, 2), (\mathrm{VW}, 2), (\mathrm{ESPIONAG}, 2)\} \end{aligned}$$

since P_5 and P_6 have the same *termset* now (note: *Spy is Espionage*). The "part-of" (the diamonds in Fig. 3) relation is used to illustrate the relation between compound objects and primitive objects. Table 3 lists the patterns in the base backbone, where $\Omega = \{P_1, P_2, P_7, P_8\}$.

Fig. 4 illustrates a top backbone of the ontology for this example, which organizes linkages between compound classes according to Table 2 and Table 3. This diagram includes eight patterns, two compositions, and two "is-a" relations since $termset(P_1) \subset termset(P_8)$ and $termset(P_2) \subset termset(P_7)$. Apart from the "is-a" relation and composition, there also exist overlaps between patterns. For example,

$$termset(P_7) \cap termset(P_8) = \{\mathrm{ESPIONAG}\}.$$

In this paper, we do not pay more attention to the construction of the base backbone (see [32] or [29] for some existing algorithms on the bottom up approach). We just assume that we can obtain a hierarchy (taxonomy) of all keywords in T, which consists of a set of clusters, Θ, where each cluster in Θ is represented as a term (note: the size of Θ is used to terminate the construction). For example, we have

$$\Theta = \{VM, German, US, Econom, Espionag, Bill, Man\}$$

for the above example. We called $\Theta \subseteq T$ the set of primitive objects.

The main task in this section is to build a model to represent the correlation in the top backbone of the ontology. Because all compound classes are constructed from some primitive ones, we use Θ as a common hypothesis space.

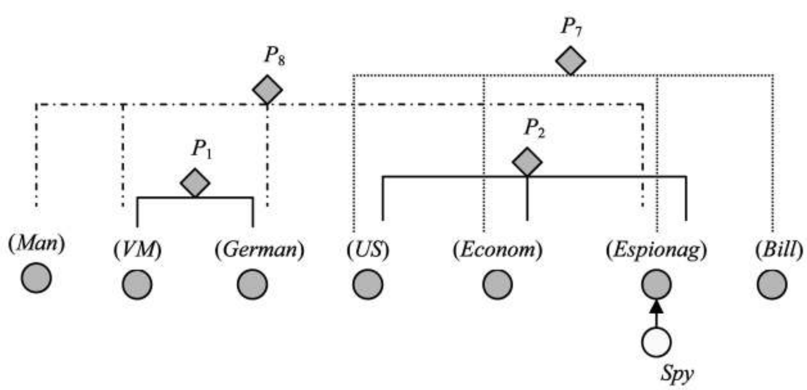

Fig. 3. The base backbone of the ontology.

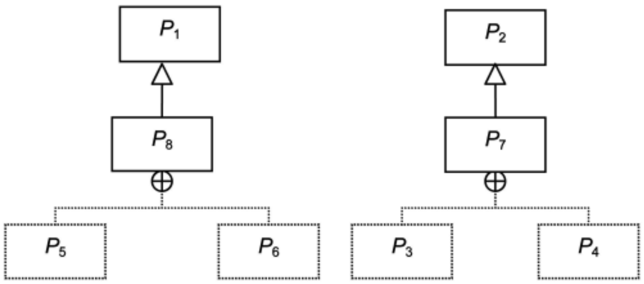

Fig. 4. The top backbone of the ontology.

We now consider how to describe the correlation between patterns over the common hypothesis space. We normalize *support* first, which satisfies:

$$support: \Omega \rightarrow [0,1], \text{such that}$$
$$support(P) = \frac{support(P)}{\sum_{P_j \in \Omega} support(P_j)} \quad (2)$$

for all $P \in \Omega$. After normalization, we have

$$\sum_{P \in \Omega} support(P) = 1.$$

We also use a mapping β to explicitly describe the relationship between patterns and the common hypothesis space, which satisfies:

$$\beta : \Omega \rightarrow 2^{\Theta \times [0,1]} - \{\emptyset\}, \text{such that}$$
$$\beta(P) = \{(t_1, w_1), (t_2, w_2), \ldots, (t_r, w_r)\} \subseteq \Theta \times [0,1],$$
$$\text{and } \beta(P) \text{ is } P\text{'s normal form.}$$

Therefore, the *correlation* can be described as a pair $< support, \beta >$, which is called an *association set* from Ω to Θ in this paper. The form of an association set looks like a random set (the concept of random sets can be found in [15]). However, a random set can only map a pattern to a termset. Different from random sets, an association set not only maps a pattern to a termset, it also provides a term weight distribution for the termset as well.

Table 4 shows an example for the representation of the correlation in the top backbone of the ontology in Fig. 4.

Algorithm $OntoMining(D^+.\Theta, \Omega, < support, \beta >)$
/* *Input parameter*: D^+; and *output parameters*: Θ, Ω, and $< support, \beta >$. */
i) // Extraction of the set of primitive objects Θ
  1) Execute basic text processing for all documents in D^+;
  2) Determine a set of terms (primitive objects);
ii) // Compound pattern generation
  1) let $\Omega = \emptyset$;
  2) for each $d \in D^+$ // discover patterns
    $\{P = \emptyset$;
     for each term $t \in d$ {
       let f be the occurrences of t in d;
       $P = P \cup \{(t, f)\}$; }
     $support(P) = 1, \Omega = \Omega \cup \{P\}$; }
  3) for each pair of patterns (P_i, P_j) // composition
    if $(termset(P_i) = termset(P_j))$
      $\Omega = (\Omega - \{P_i, P_j\}) \cup \{P_i \oplus P_j\}$;
iii) // representation of correlations
  Evaluate β and get an association set $< support, \beta >$;

TABLE 4
An Association Set $< support, \beta >$ from Ω to Θ, where $\Theta = \{VM, German, US, Econom, Espionag, Bill, Man\}$ and $\Omega = \{P_1, P_2, P_7, P_8\}$

Name	*Support*	β
P_1	1/6	{(GERMAN, 1/2), (VW, 1/2)}
P_2	1/6	{(US, 1/2), (ECONOM, 1/4), (ESPIONAG, 1/4)}
P_7	1/3	{(US, 1/4), (BILL, 1/4), (ECONOM, 1/4), (ESPIONAG, 1/4)}
P_8	1/3	{(GERMAN, 1/3), (MAN, 2/9), (VW, 2/9), (ESPIONAG, 2/9)}

Algorithm *OntoMining* describes the process of mining an ontology. In Step i), we first use the basic text processing for all positive documents, including case folding, stemming, and stop words and nonkeywords removal, where the *tf*idf* method is used to determine keywords. We also need to determine a set of terms using an existing hierarchical clustering algorithm (see [32] or [29]), which illustrates "synonymy" and "hyponymy" ("is-a") relations between keywords. The simplest case is that each keyword can be viewed as a term. In Step ii), the patterns are discovered directly from positive documents. We also compose those patterns with the same termset, remove them from the set of patterns, and insert the composition into the set of patterns. At last, the algorithm evaluates the correlation.

For example, using the above algorithm we can obtain a triple $(\Theta, \Omega, < support, \beta >)$ to represent the discovered ontology, where Θ, Ω, and $< support, \beta >$ are defined in Table 4. More formally, the discovered ontology is represented in an XML document (see Appendix A, the initial xml file, and Appendix B for its DTD file, which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm). In order to reuse defined elements in the DTD file, we use "**TermSet**" as a tag name. The meaning of **TermSet** in XML is different to the meaning of *termset* since we assign each term in **TermSet** a weight property to accumulate the results of relevance assessments on the ontology in Section 5.

Fig. 5 illustrates the document object model (DOM) of the XML document for the discovered ontology, $(\Theta, \Omega, < support, \beta >)$. The ontology about the **TOPIC** "Economic espionage" consists of a **TermSet** and a **Correlation**, where the boldfaces are elements in the corresponding XML document, "**P**" denotes "**Pattern**" element, and "w" denotes "weight" attribute.

4 Ontology Evolution

Indeed, not all discovered patterns are adequate for describing interesting topics because of noise in the training data [23]. The consequential result is that some irrelevant documents may be marked as relevant by the system. Increasing the size of the training set is useless if we do not remove the noise. In this section, we present an approach for tracing errors made by the system. The approach

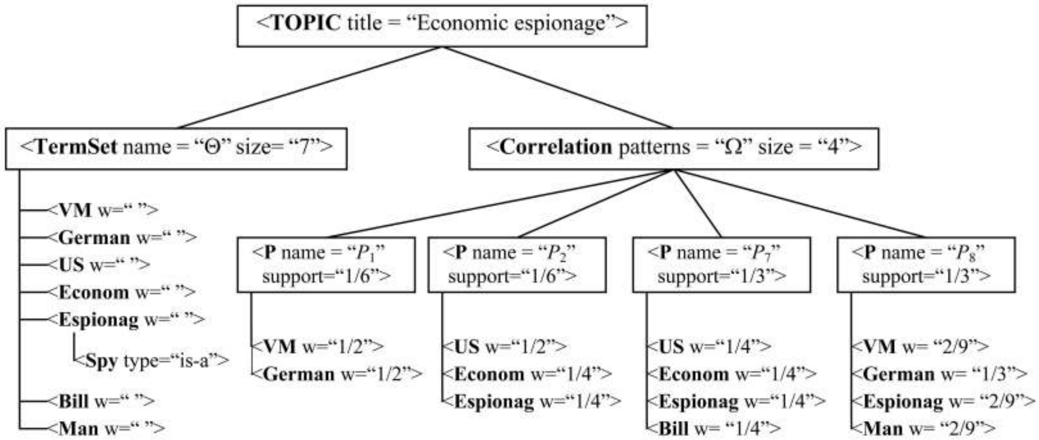

Fig. 5. DOM for the discovered ontology.

provides a novel technique to update the discovered knowledge.

A negative document is called an *interesting negative document* if it is marked as relevant by the system. The approach we use in this research is to trace the cause of the occurrences of interesting negative documents. For a given interesting negative one, nd, we check which patterns have been used to give rise to such error. We call these patterns *offenders* of nd. The set of offenders of nd can be determined by the following equation:

$$\Delta_{nd} = \{P \in \Omega | termset(P) \cap nd \neq \emptyset\}.$$

Fig. 6 demonstrates a paradigm about an interesting negative pattern, nd, and its offenders, where P_i, P_j, and P_k are three discovered patterns. Since $termset(P_i) \cap nd \neq \emptyset$ and $termset(P_j) \cap nd \neq \emptyset$ but $termset(P_k) \cap nd = \emptyset$, we have $\Delta_{nd} = \{P_i, P_j\}$.

There are two kinds of offenders: total conflict offenders whose $termsets$ are subsets of nd and partial conflict offenders whose $termsets$ are not subsets of nd but join with nd. For instance, P_i in Fig. 6 is a total conflict offender since $termset(P_i) \subseteq nd$, but P_j is a partial conflict offender since $termset(P_j) \not\subseteq nd$ but $termset(P_j) \cap nd \neq \emptyset$.

For example, given the following interesting negative document:

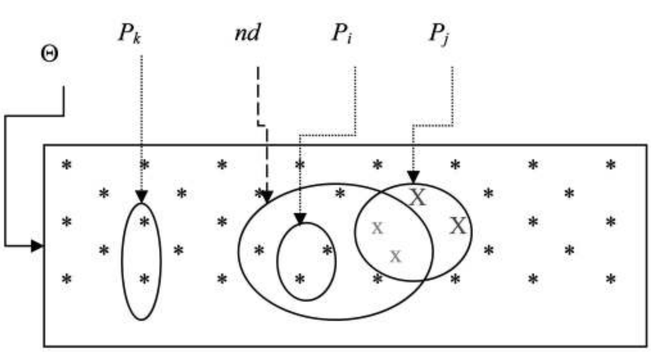

Fig. 6. An interesting negative document, nd, and its offenders.

$$nd_1 = \text{"GERMAN FOCUS VW unveils new Passat says sales,"}$$

we have $\Delta_{nd_1} = \{P_1, P_8\}$ for the patterns in Table 4 since

$$\begin{aligned} termset(P_1) \cap nd_1 &= termset(P_8) \cap nd_1 \\ &= \{\text{GERMAN}, \text{VW}\} \neq \emptyset, \end{aligned}$$

but $termset(P_2) \cap nd_1 = termset(P_7) \cap nd_1 = \emptyset$. Also, P_1 is a total conflict offender of nd_1 and P_8 is a partial conflict offender of nd_1 according to the above definitions.

Fig. 7 illustrates the relationship between discovered patterns and the interesting negative document. In this figure, we only show the important relation: the “is-a” relation. This figure also indicates that pattern P_1 may be a noninteresting pattern since it derives both positive patterns (e.g., P_8) and negative documents (e.g., nd_1).

The basic idea of updating the discovered ontology is explained as follows: We reduce the supports for all total conflict offenders (e.g., P_1 in Fig. 7). They may be removed if their supports are less than a minimum support, min_sup.

For partial conflict offenders, we reshuffle their weight distributions to evaporate the uncertainties contained in the patterns. Given an interesting negative document nd, its *reshuffle operation* first determines the *offering* of each offender from the joint part between nd and the offender (for convenience, we call this part the **lowercase part**) using the following equation:

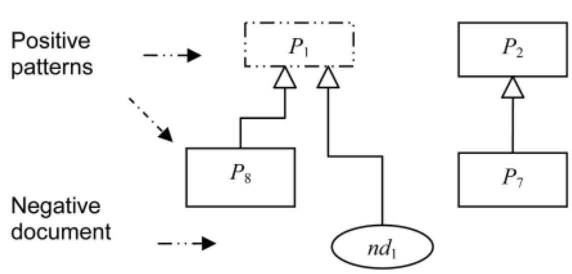

Fig. 7. The relationship between patterns and the interesting negative document.

TABLE 5
Example of the Reshuffle Operation Given nd_1 and P_8 (see Fig. 6 and Table 4), where $\mu = 2$

Definition	*Result*
$\beta(P_8)$, before the operation	{(GERMAN, 1/3), (MAN, 2/9), (VW, 2/9), (ESPIONAG, 2/9)}
Lower case part	{GERMAN, VW}
Upper case part	{MAN, ESPIONAG}
The *offering* from P_8	$\sum_{(t,w)\in\beta(P_8),t\in nd_1} w = 1/3 + 2/9 = 5/9$
$\beta(P_8)$, after the operation	{(GERMAN, 1/6), (MAN, 13/36), (VW, 1/9), (ESPIONAG, 13/36)}

$$offering : \Delta_{nd} \to [0,1],$$
$$\text{such that } offering(P) = \sum_{(t,w)\in\beta(P),t\in nd} w$$

for all partial conflict offenders $P \in \Delta_{nd}$. The reshuffle operation also shifts part of the offering to the remaining part of the offender (we call this part the **uppercase part**).

For example, given a partial conflict offender P_j in Fig. 6, the $\frac{\mu-1}{\mu}$ of offering obtained from its two lowercase x terms will be shifted to its two uppercase **X** terms, where μ is an experimental coefficient and $\mu > 1$. Table 5 interprets how the reshuffle operation works according to the example in Table 4, where P_8 is a partial conflict offender of nd_1.

Table 6 summarizes the result of the evolution given the interesting negative document nd_1. The result of evolution can also be recorded in the corresponding XML document (see **Correlation** element in Appendix C, which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm).

Algorithm *PatternEvolving* describes the above idea. In Step i), the algorithm finds the set of offenders. In Step ii), it uses a "for" loop to update every offender. There are two cases for offenders: total conflict ($itemset(P) \subseteq nd$) and partial conflict (*else*). For the total conflict offenders, the algorithm declines their supports or even removes them if their supports are less than *min_sup*; otherwise, it gets part of the offering, *s_offering*, from their lowercase part first. It also calculates the *base*, which is certainly not zero, and then updates the weight distributions using a "for" loop. For a given interesting negative document *nd*, the time complexity of the algorithm is $O(nm^2)$ if let $nd = nd \cap \Theta$, where $n = |\Omega|$, and $m = |\Theta|$.

Algorithm $PatternEvolving(nd, \Theta, \Omega, < support, \beta > .$
$\Omega, < support, \beta >)$
/* *Input parameters*: *nd*, Θ, Ω, and $< support, \beta >$;
Ω and $< support, \beta >$ are updated. */
i) // Find the set of offenders
$\Delta_{nd} = \{P \in \Omega | termset(P) \cap nd \neq \emptyset\}$;
ii) for each $P \in \Delta_{nd}$
if ($itemset(P) \subseteq nd$) { //for total conflict offenders
$support(P) = (\frac{1}{\mu}) \times support(P)$;
if ($support(P) < min\_sup$) //removal
$\Omega = \Omega - \{P\}$;}
else { //reshuffle operation
$s\_offering = (\frac{\mu-1}{\mu}) \times \sum_{(t,w)\in\beta(P),t\in nd} w$;
$base = \sum_{(t,w)\in\beta(P),t\notin nd} w$;
for each $(t,w) \in \beta(P)$
if ($t \in nd$) // for lower case part
$w = (\frac{1}{\mu}) \times w$;
else //for upper case part
$w = w + (s\_offering \times w) \div base$; }
iii) Normalize the *support* function using (2);

TABLE 6
Result of Pattern Evolving Given nd_1

Name	*Support*	β
P_1	1/11	{(GERMAN, 1/2), (VW, 1/2)}
P_2	2/11	{(US, 1/2), (ECONOM, 1/4), (ESPIONAG, 1/4)}
P_7	4/11	{(US, 1/4), (BILL, 1/4), (ECONOM, 1/4), (ESPIONAG, 1/4)}
P_8	4/11	{(GERMAN, 1/6), (MAN, 13/36), (VW, 1/9), (ESPIONAG, 13/36)}

As we assumed, discovered patterns are obtained directly from the set of positive documents, so, normally they describe more features of positive documents than negative ones. In our experiments, we call the algorithm for some topics in which the number of positive documents in the training set is greater than the number of interesting negative documents. We also tested different values for parameter μ: 2, 4, 8, and 16. They all performed well and $\mu = 8$ is the best one for considering both top 25 precision and breakeven point. In addition, min_sup is determined based on how many times a pattern is a total conflict offender. In Section 6.2, we provide the details of using this algorithm in our experiments. We also have the following theorem:

Theorem 1. *Let Ω be a set of discovered patterns, Θ be a set of terms, and $< support, \beta >$ be an association set from Ω to Θ. Given an interesting negative pattern nd, after calling Algorithm* PatternEvolving, *the pair $< support, \beta >$ is still an association set.*

Proof. From Algorithm *PatternEvolving*, it is obvious that, after finishing Step iii), *support* function satisfies $\sum_{P\in\omega} support(P) = 1$ if Ω does not equal empty. According to the algorithm, the weight distributions of total conflict offenders are not changed, although some of them may be removed. The algorithm only adjusts the weight distributions of partial conflict offenders.

For any partial conflict offender P_0, we denote it conveniently as P once finishing Step ii) in the following calculation. We have:

$$\sum_{(t,w)\in\beta(P)} w = \Bigg(\sum_{(t,w)\in\beta(P),t\in nd} w\Bigg) + \Bigg(\sum_{(t,w)\in\beta(P),t\notin nd} w\Bigg)$$
$$= \Bigg(\sum_{(t,w)\in\beta(P_0),t\in nd} \left(\frac{1}{\mu}\right) w\Bigg)$$
$$+ \Bigg(\sum_{(t,w)\in\beta(P_0),t\notin nd} (w + (w \times s\_offering \div base))\Bigg)$$
$$= \Bigg(\frac{1}{\mu}\sum_{(t,w)\in\beta(P_0),t\in nd} w\Bigg) + \Bigg(\sum_{(t,w)\in\beta(P_0),t\notin nd}$$
$$\Bigg(w + \Bigg(w\left(\frac{\mu-1}{\mu}\right)\sum_{(t,w)\in\beta(P_0),t\in nd} w\Bigg) \div \Bigg(\sum_{(t,w)\in\beta(P_0),t\notin nd} w\Bigg)\Bigg)\Bigg)$$
$$= \Bigg(\frac{1}{\mu}\sum_{(t,w)\in\beta(P_0),t\in nd} w\Bigg) + \Bigg(\sum_{(t,w)\in\beta(P_0),t\notin nd} w\Bigg)$$
$$+ \frac{(1-\frac{1}{\mu})\sum_{(t,w)\in\beta(P_0),t\in nd} w}{\sum_{(t,w)\in\beta(P_0),t\notin nd} w}\sum_{(t,w)\in\beta(P_0),t\notin nd} w$$
$$= \Bigg(\frac{1}{\mu}\sum_{(t,w)\in\beta(P_0),t\in nd} w\Bigg) + \Bigg(\sum_{(t,w)\in\beta(P_0),t\notin nd} w\Bigg)$$
$$+ \Bigg(\left(1-\frac{1}{\mu}\right)\sum_{(t,w)\in\beta(P_0),t\in nd} w\Bigg) = \Bigg(\sum_{(t,w)\in\beta(P_0),t\in nd} w\Bigg)$$
$$+ \Bigg(\sum_{(t,w)\in\beta(P_0),t\notin nd} w\Bigg) = \sum_{(t,w)\in\beta(P_0)} w = 1.$$

□

5 Relevance Assessments in Ontology

In general, the concept of relevance is subjective. We normally describe the relevance of a specified topic in two dimensions: specificity and exhaustivity. It is easy for human experts to subjectively assess objects using several scales. For example, we may use 0 to denote not specific, 1 to denote marginally specific, 2 to denote fairly specific, and 3 to denote highly specific (see [9]).

In this section, the main theme is the automatic extraction of relevance assessments directly from data for the discovered ontology. For this purpose, we present a formal definition for learning the two dimensions about reasoning on ontologies. The relevance of a topic in the ontology is assessed according to the following two dimensions:

- exhaustivity (*exh* for short), which describes the extent to which the pattern (or topic) discusses what users want and
- specificity (*spe* for short), which describes the extent to which the pattern (or topic) focuses on what users want.

A pattern in the ontology can be assessed as highly exhaustive relatively even though it is not specific to what users want. Similarly, a pattern can be assessed as highly specific relatively even though it discusses many or only a few aspects of what users want. However, a pattern that does not discuss what users want at all must have the lowest specificity.

TABLE 7
Examples of Intervals of *Specificity* and *Exhaustivity*

Name	*Support*	*termset*	*spe*	*exh*
P_1	1/11	{GERMAN, VW}	1/11	5/11
P_2	2/11	{US, ECONOM, ESPIONAG}	2/11	10/11
P_7	4/11	{US, BILL, ECONOM, ESPIONAG}	6/11	10/11
P_8	4/11	{GERMAN, MAN, VW, ESPIONAG}	5/11	1

In this paper, we define the following two numeral functions for measuring *exhaustivity* and *specificity*, respectively:

$$\begin{aligned}
&spe : 2^\Theta \rightarrow [0,1];\ \text{such that} \\
&\qquad spe(A) = \sum_{P\in\Omega, termset(P)\subseteq A} support(P), \\
&exh : 2^\Theta \rightarrow [0,1];\ \text{such that} \\
&\qquad exh(A) = \sum_{P\in\Omega, termset(P)\cap A\neq\emptyset} support(P),
\end{aligned} \tag{3}$$

for all $A \subseteq \Theta$.

In (3), the Dempster-Shafer theory (see [15]) is used for the concept of relevance to a specified topic, where spe is a belief function and exh is a plausibility function, respectively. The rationale of using Dempster-Shafer theory is discussed below.

Given a pattern P, its *exhaustivity* and *specificity* can be evaluated as $exh(termset(P))$ and $spe(termset(P))$, respectively. It is obvious that $0 \leq spe(termset(P)) \leq 1$, $0 \leq exh(termset(P)) \leq 1$, and

$$spe(termset(P)) \leq exh(termset(P))$$

for all $P \in \Omega$, where 0 means the pattern does not discuss or is not the theme of what users want and 1 means the pattern discusses most of or only the theme of what users want. According to (3), the specificity of pattern P is expressed by all its subpatterns (i.e., the belief in relation to it) and its exhaustivity is expressed by all patterns that overlap with it (i.e., the plausibility in relation to it).

Different from the method of using several predetermined scales (see [9]), here we use *relative* specificity and *relative* exhaustivity to denote that a small specificity value may be interpreted as fairly specific and a large exhaustivity value may be interpreted as marginally exhaustive.

Table 7 illustrates examples of patterns' *specificity* and *exhaustivity* intervals. For P_1, we have an interval [1/11, 5/11] which means that pattern P_1 is marginally specific relatively and marginally exhaustive relatively. We also say P_7 is highly specific relatively and highly exhaustive relatively given the interval [6/11, 10/11].

The above two dimension approach provides us with a basic framework to assess the relevance of patterns in the ontology. However, the two-dimension approach is only practical for human beings, not for computers, since an ambiguity may arise for determining the best interval (e.g., we are not confident which one is better given [6/11, 10/11] and [5/11, 1]). Also, the time complexity of calculating *specificity* and *exhaustivity* intervals is intolerable because of the prerequisite of determining many subsets.

Therefore, it is desirable to synthesize the two dimensions into one dimension. That requires a single function, *relevance*, for all documents. Function *relevance* is called *sound* if it satisfies:

TABLE 8
A Deduced Probability

Term in Θ	pr_β
US	0.1818182
GERMAN	0.1060606
BILL	0.09090909
ECONOM	0.1363636
ESPIONAG	0.2676768
MAN	0.1313131
VW	0.08585858

$$spe(d) \leq relevance(d) \leq exh(d)$$

for all documents d, where $spe(d)$ is the short form of $spe(d \cap \Theta)$ and $exh(d)$ is the short form of $exh(d \cap \Theta)$.

To obtain a relevance function, we first calculate a probability function pr_β from a given association set $< support, \beta >$, which satisfies:

$$pr_\beta(t) = \sum_{P \in \Omega, (t,w) \in \beta(P)} support(P) \times w \tag{4}$$

for all $t \in \Theta$. We call pr_β the probability function deduced by β. Table 8 illustrates an example of the deduced probability if we use the association set in Table 4. We can also accumulate the result into the corresponding XML document (see **TOPIC//TermSet** element in Appendix C, which can be found on the Computer Society Digital Library at http://www.computer.org/tkde/archives.htm).

Finally, a *relevance* function for Web documents can be defined as follows:

$$relevance(d) = \sum_{t \in \Theta} pr_\beta(t)\tau(t,d), \quad \text{where } \tau(t,d) = \begin{cases} 1 & if\ t \in d \\ 0 & otherwise. \end{cases} \tag{5}$$

In summary, we have the following theorem according to the above discussions.

Theorem 2. *Let* $< support, \beta >$ *be an association set from* Ω *to* Θ*. We have:*

1. pr_β *is a probability function on* Θ *and*
2. *function* relevance *is sound.*

Proof.

1. From (4), we have:

$$\sum_{t \in \Theta} pr_\beta(t) = \sum_{t \in \Theta} \sum_{P \in \Omega, (t,w) \in \beta(P)} support(P) \times w$$
$$= \sum_{P \in \Omega} \sum_{(t,w) \in \beta(P)} support(P) \times w$$
$$= \sum_{P \in \Omega} support(P) \sum_{(t,w) \in \beta(P)} w$$
$$= \sum_{P \in \Omega} support(P) \times 1 = 1.$$

2. Let $A = d \cap \Theta$. From (3), we have:

$$spe(A) = \sum_{P \in \Omega, termset(P) \subseteq A} support(P).$$

Also, from (5) and (4), we have:

$$relevance(d) = \sum_{t \in \Omega} pr_\beta(t)\tau(t,d)$$
$$= \sum_{t \in A} pr_\beta(t) = \sum_{t \in A} \sum_{P \in \Omega, (t,w) \in \beta(P)} support(P) \times w$$
$$= \sum_{P \in \Omega, (t,w) \in \beta(P), t \in A} support(P) \times w$$
$$= \left(\sum_{P \in \Omega, termset(P) \subseteq A} \sum_{(t,w) \in \beta(P)} support(P) \times w \right) +$$
$$\left(\sum_{P \in \Omega, termset(P) \not\subseteq A} \sum_{(t,w) \in \beta(P), t \in A} support(P) \times w \right)$$
$$\geq \sum_{P \in \Omega, termset(P) \subseteq A} \sum_{(t,w) \in \beta(P)} support(P) \times w$$
$$= \sum_{P \in \Omega, termset(P) \subseteq A} \left(support(P) \times \sum_{(t,w) \in \beta(P)} w \right)$$
$$= \sum_{P \in \Omega, termset(P) \subseteq A} support(P) = spe(A).$$

That is $spe(A) \leq \sum_{t \in A} pr_\beta(t)$.

Analogically, we can prove

$$\sum_{t \in A} pr_\beta(t) \leq exh(A)$$

□

Table 9 demonstrates an example for the comparison between relevance, specificity, and exhaustivity if we view each $termset$ as a document. It also indicates that pattern P_7 is the best one since it has the greatest relevance.

In order to evaluate relevance theoretically, we call a document d *logical relevance* if $\exists P \in \Omega$ such that $termset(P) \subseteq d$. We use $[P]$ to denote the *covering set* of P, which includes all documents d such that $termset(P) \subseteq d$. We call a method or an algorithm *complete* if it can retrieve all logically relevant documents $\bigcup_{P \in \Omega}[P]$.

Theorem 3. *Let* $< support, \beta >$ *be an association set from* Θ *to* Ω*. We have:*

$$relevance(d) \geq min_{P \in \Omega} \left\{ \sum_{(t,w) \in \beta(P)} pr_\beta(t) \right\}$$

for all $d \in \bigcup_{P \in \Omega}[P]$.

Proof. Assume $d \in \bigcup_{P \in \Omega}[P]$, that is, there exists a $P_0 \in \Omega$ such that $termset(P_0) \subseteq d$. From (5), we have $relevance(d) = \sum_{t \in d} pr_\beta(t)$. Since $pr_\beta(t) \geq 0$ for all $t \in \Omega$, we have:

$$\sum_{t \in d} pr_\beta(t) \geq \sum_{t \in termset(P_0)} pr_\beta(t) =$$
$$\sum_{(t,w) \in \beta(P_0)} pr_\beta(t) \geq min_{P \in \Omega} \left\{ \sum_{(t,w) \in \beta(P)} pr_\beta(t) \right\}.$$

□

According to the above theorem, we use the following formula to decide a threshold while we use the discovered ontology to assess the relevance of documents:

TABLE 9
Relevance, Exhaustivity, and Specificity

Name	Support	Termset	spe	relevance	exh
P_1	1/6	{GERMAN, VW}	1/11	0.1919198	5/11
P_2	1/6	{US, ECONOM, ESPIONAG}	2/11	0.5858586	10/11
P_7	1/3	{US, BILL, ECONOM, ESPIONAG}	6/11	0.67676769	10/11
P_8	1/3	{GERMAN, MAN, VW, ESPIONAG}	5/11	0.59090908	1

$$threshold = \min_{P \in \Omega} \left\{ \sum_{(t,w) \in \beta(P)} pr_\beta(t) \right\}. \tag{6}$$

The obvious advantage of using (6) is that we can retrieve all logically relevant documents.

Algorithm $RelAssess$ illustrates the details of making binary decisions. In this algorithm, a threshold is determined in Step i) according to both Ω (the set of discovered patterns) and the deduced probability. In Step ii), it estimates documents' relevance values and determines their relevance if their relevance values are greater than or equal to the $threshold$.

Algorithm $RelAssess(\Omega, \Theta, < support, \beta > . docs, rel)$
/* *Input parameters*: Ω, Θ, and $< support, \beta >$;
output parameters: *docs* and *rel* which accommodate document relevance pairs and relevant documents, respectively.
*/
i) $threshold = \min_{P \in \Omega}\{\sum_{(t,w)\in\beta(P)} pr_\beta(t)\}$;
ii) $rel = \emptyset$, $docs = \emptyset$; // let rel and $docs$ be empty.
for each d // for all documents in the testing set.
{
$relevance(d) = \sum_{t\in\Theta} pr_\beta(t)\tau(t,d)$;
$docs = docs \cup \{(d, relevance(d))\}$;
if $(relevance(d) \geq threshold)$
$rel = rel \cup \{d\}$;
}

The time complexity of the algorithm for evaluating pr_β is O(nm) (see [24]), where $n = |\Omega|$ and $m = |\Theta|$. The time complexity for determining a threshold in Step i) is also O(nm). Also, in Step ii), the time complexity of making a binary decision for each document is O$(s\ log^m)$, where s is the average size of documents. So, the time complexity of the algorithm is O$(mn + z\ s\ log^m)$, where z is the number of documents in the testing set.

It is obvious that Algorithm $RelAssess$ is complete. We also use experiments to evaluate the actual performance of the algorithm in next section.

6 Testing and Evaluations

In this section, we evaluate the proposed algorithms. We also use standard data collections, TREC2002 data collections for "filtering track" (Text REtrieval Conference, see http://trec.nist.gov/), which include Reuters Corpus articles from 1996-08-20 to 1997-08-19. In our experiments, we test two dozen topics: 101, 102, ..., and 124. We use about 10,000 XML documents in the experiments.

For each specified topic, its data collection is split into two sets: a training set and a testing set. In order to comprehensively test the proposed algorithms, we use two threads in our implementation: a *semisupervised* testing model (*Round* 1) and a *supervised* testing model (*Round* 2). The former estimates term weights based on the positive documents only and the latter considers both positive and some negative documents in the training set. We also use several other models for comparison. A common basic text processing is used for all models, which includes case folding, stemming, stop words removal, and term selection.

6.1 Popular Existing Techniques

The popular techniques that can be used for semisupervised estimation of term weights are the *Rocchio* classification [17] and Dempster-Shafer (DS) model [41], [24].

The technique used in *Rocchio* classification is called *tf*idf* (term frequency times inverse document frequency). Most models use this technique [33], [12], [40] for representing user profiles. For every term t in a given document d, its weight is determined by $w_d(t) = tf \times \log(N/df)$, where *tf* is the number occurrences of term t in document d, N is the total number of documents in the training set, and *df* is the number of documents in the training set that contain term t.

Given a topic, let D be the training set and D^+ be the set of positive documents. The *Rocchio* method can be simplified as follows, which uses $w_{Rocchio}(t) = \sum_{d\in D^+} w_d(t)$ to represent the centroid of positive documents. The cosine measure is also used to estimate the similarity between a new document d in the testing set and the topic.

The DS term weight technique first obtains a mass function on Θ, the set of terms, which satisfies

$$m : 2^\Theta \rightarrow [0, 1];$$
$$m(A) = \sum_{P\in\Omega, termset(P)=A} support(P).$$

It then transfers the mass function into a *pignistic* probability to estimate term weights:

$$w_{DS}(t) = \sum_{\emptyset \neq A \subseteq \Theta, t \in A} \frac{m(A)}{|A|}$$

for all $t \in \Theta$, where $|A|$ is the number of elements in A.

Probabilistic models are popular methods for supervised estimation of term weights. The individual term weight is estimated in the basic probabilistic model ($Prob1$) based on how often the term appears or does not appear in positive documents and negative documents, respectively. The rough set-based filtering model [20] has used such an idea to represent user profiles. $Prob1$ uses the equation:

$$w_{Prob1}(t) = \log\left(\frac{rdf}{R} \div \frac{df}{N}\right)$$

to estimate term weights, where rdf is the number of positive documents that contain the term t, R is the number of positive documents in the training set, and *df* and N are defined the same as in the *tf*idf* model, respectively.

To remove uncertainties involved in estimating term weights, the basic probabilistic model was revised into $Prob2$ which uses both the presence of search terms in documents and their absence from documents. The weight function appears as:

$$w_{Prob2}(t) = \log\left(\frac{rdf + 0.5}{(R - rdf) + 0.5} \div \frac{(df - rdf) + 0.5}{(N - df) - (R - rdf) + 0.5}\right),$$

where 0.5, an adequate experimental coefficient [8], is used to account for the uncertainty involved in estimating relevance.

6.2 Analysis of Computational Complexity

In the experiments, each model includes two phases: training and filtering. The procedure of filtering is similar to Algorithm $RelAssess$, where we do not require a threshold and only calculate relevance for documents in the testing set. We also need to sort the results. All models have the same time complexity in the filtering phase.

For *Round*1 in the training phase, we first use the positive documents in the training set and Algorithm $OntoConstruction$ to discover an ontology which consists of a set of discovered patterns Ω and an association set $< support, \beta >$ for each topic. Term weights are determined by pr_β, the probability function deduced by β. The basic procedure for estimating term weights (the probability function pr_β) in *Round*1 is described as follows:

for each $d \in D^+$ {
  Represent d as a pattern in normal form
    $\{(t_1, w_1), (t_2, w_2), \ldots, (t_r, w_r)\}$;
  Let Ω be the set of patterns;
  Let $support(P) = 1/|\Omega|$ for all $P \in \Omega$;
  For each $t \in \Theta$ //starting to estimate pr_β,
    $pr_\beta(t) = 0$;
  for each $P \in \Omega$
    for each $((t_i, w_i) \in P)$
      $pr_\beta(t_i) = pr_\beta(t_i) + support(P) \times w_i$; }

To improve the efficiency of *Round*1, composition operations are not used here, but it is easy to verify that the results are correct. This consideration may increase the time complexity of *Round*2 since Ω may accumulate more than one pattern with the same termset. However, we believe this consideration is better because we do not need to test every pair of patterns in Ω. The time complexity of *Round*1 in the training phase is $O(n_1 ms)$ since it only needs a traversal through only positive documents, where s is the average size of documents; $n_1 = |D^+|$ and $m = |\Theta|$. To compare this with other models, *Round*1 is the best one since other models may need a traversal through both positive documents and negative ones. For example, the time complexity of both probabilistic models is $O(n_2 ms) + O(n_1 ms)$, where $n_2 = |D^-|$.

In *Round*2, we use interesting negative documents in the training set to update Ω and $< support, \beta >$ by using Algorithm $PatternEvolving$, where $\mu = 8$ and min_sup is $\frac{1}{n_1 \mu^3}$. The basic procedure of evolution is described as follows:

Calculate a $threshold$ for each topic (see Algorithm $RelAssess$);
If (the number of interesting negative documents < the number of positive documents)
  Call Algorithm $PatternEvolving$;
Estimate term weights again using the method in *Round* 1;

According to the above procedure and Algorithm $PatternEvolving$, the time complexity of *Round*2 in the training phase is $O(n_3 n_1 m^2) + O(n_1 ms)$, where n_3 is the number of interesting negative documents.

In our implementation, we set $m = 150$. The average of n_1 in the first nine topics is 36 (see Table 15 in Section 6.3) and the percentage of negative documents that we use for training in *Round*2 is 38.6 percent, that is, $n_3 = n_2 \times 38.6\%$. Therefore, we have

$$\begin{aligned} &O(n_3 n_1 m^2) + O(n_1 ms) \\ &= O(n_3 m(150 \times 36)) + O(n_1 ms) \\ &= O(n_2 m(5,400 \times 38.6\%)) + O(n_1 ms) \\ &= O(n_2 m(2,084)) + O(n_1 ms) \\ &\cong O(n_2 ms) + O(n_1 ms), \end{aligned}$$

which means *Round* 2 is also efficient.

6.3 Analysis of Effectiveness

In order to clearly illustrate the results of these experiments, we first discuss all details for the first nine topics and then we use figures to demonstrate the results for all 24 topics. We use both precision and recall in the experiments, where the *precision* is the fraction of retrieved documents that are relevant to the topic and the *recall* is the fraction of relevant documents that have been retrieved.

We sort results first according to the relevance values of documents. We can obtain two arrays of floats for each method: xr array and yp array for the recall and precision, respectively, where the cut-off = 25. Instead of drawing many precision recall curves, we use both *top 25 precision* and *breakeven point*, which are two methods used in Web mining for testing effectiveness, where a breakeven point is a point in the precision and recall curve with the same x coordinate and y coordinate. The greater both the top 25 precision and the breakeven point, the more effective the method is.

We first compare our *Round*1 result with *Rocchio* classification and the *DS* model since they all use positive documents only for training. Table 10 and Table 11 illustrate results of both top 25 precision and breakeven points for this comparison. It is very clear that *Round*1 grants very good performance on both top 25 precision and breakeven points.

We also compare *Round*1 and *Round*2 with the probabilistic models. Table 12 and Table 13 show the results of this comparison. The performance of the proposed algorithms is no less impressive since both top 25 precision and breakeven points gain a significant increase.

Another advantage of our approach is that it can decrease the burden of online training since the proposed algorithm only requires a small amount of negative documents. It is significant for Web-based information gathering because of the huge amount of nonrelevant data. Table 14 shows the number of interesting negative documents and the number of negative documents in the

TABLE 10
Semisupervised Evaluation on Top 25 Precision

Topic	*Round*1	*Rocchio*	*DS*
101	100%	84.0%	92%
102	96%	92.0%	96%
103	52%	44.0%	40%
104	96%	100%	96%
105	72%	52.0%	68%
106	8%	32.0%	12%
107	52%	44.0%	36%
108	36%	40.0%	40%
109	40%	32.0%	44%
Avg	**61.3%**	57.8%	58.2%

TABLE 11
Semisupervised Evaluation on Breakeven Point

Topic	*Round*1	*Rocchio*	*DS*
101	0.798077	0.822805	0.760943
102	0.788814	0.791271	0.78741
103	0.397374	0.401274	0.374165
104	0.684025	0.634202	0.645915
105	0.666868	0.421053	0.678572
106	0.064516	0.27907	0.096774
107	0.384615	0.330508	0.336
108*			
109	0.294708	0.305168	0.306769
Avg	**0.510**	0.498	0.498

*The breakeven point does not exist.

training set. The percentage of negative documents that we use for training is 152/394 = 38.6 percent.

In addition, we also demonstrate the performance of the one dimension method for relevance assessment. Table 15 depicts the result. From this table, we can demonstrate that the thresholds are appropriate if the number of positive documents in the training set is big enough. When, for instance, the number of positive documents in the training set is greater than 7, the average recall is 81.48 percent (see shaded rows in Table 15).

For all 24 topics, on average, $Rocchio$ and DS have a similar performance and $Prob2$ performs better than $Prob1$. $Rocchio$ obtains a pair of (48.3 percent, 0.474) for its average top 25 precision and breakeven point; and $Prob2$ has a pair of (48.3 percent, 0.475). *Round*1 performs very well with a pair of (53 percent, 0.489) and *Round*2 achieves the best result with a pair of (54.3 percent, 0.498).

Fig. 8 and Fig. 9 show the difference between *Round*1 and $Rocchio$, and the difference between *Round*2 and $Prob2$ in breakeven point and top 25 precision for all topics, respectively. The positive values (the bars above the horizontal axis) mean our model performed better than others. The negative values (the bars below the horizontal axis) mean others performed better than our model.

TABLE 12
Supervised Evaluation on Top 25 Precision

Topic	*Round*1	*Round*2	*Prob*1	*Prob*2
101	100%	100.0%	100%	96.0%
102	96%	84.0%	96%	96.0%
103	52%	40.0%	40%	48.0%
104	96%	96.0%	96%	100.0%
105	72%	80.0%	80%	92.0%
106	8%	32.0%	20%	20.0%
107	52%	52.0%	20%	24.0%
108	36%	36.0%	36%	40.0%
109	40%	68.0%	40%	44.0%
Avg	61.3%	**65.3%**	58.7%	62.2%

TABLE 13
Supervised Evaluation on Breakeven Point

Topic	*Round*1	*Round*2	*Prob*1	*Prob*2
101	0.798077	0.798077	0.796742	0.787655
102	0.788814	0.704539	0.786335	0.8014
103	0.397374	0.4	0.397374	0.42623
104	0.684025	0.667196	0.655487	0.686318
105	0.666868	0.571429	0.64151	0.681657
106	0.064516	0.27907	0.230769	0.230769
107	0.384615	0.384615	0.168	0.189189
108*				
109	0.294708	0.471754	0.332278	0.386461
Avg	0.510	**0.535**	0.501	0.524

*The breakeven point does not exist.

The performance of our model is impressive since both top 25 precision and breakeven points gain an increase. Compared with $Rocchio$, *Round* 1 improves top 25 precision by $(53\% - 48.3\%) = 4.7\%$ on average and, compared with $Prob2$, *Round* 2 improves top 25 precision by $(54.3\% - 48.3\%) = 6.0\%$ on average.

As a result of these experiments, we believe that the proposed algorithms are significant since they can achieve the best result.

7 Related Work and Discussion

As mentioned in the introduction, currently Web mining can be classified into four categories: Web usage mining, Web structure mining, Web content mining, and Web user profile mining [28], [7], [43], [27]. The obvious difference between data mining and Web mining is that the former is based on databases and the latter is based on Web data, such as unstructured documents (e.g., HTML), semistructured documents (e.g., XML), Web log, services, and user profiles [16], [21].

Most researchers in the data mining community have focused their efforts on finding efficient algorithms for dealing with huge amounts of data. However, determining useful and interesting patterns is still an open problem [25]. The fundamental concern in this research is the automatic meaning discovery rather than pattern discovery. Some pilot work has been conducted on this question. If databases were represented as relational databases and the users

TABLE 14
Percentage of Interesting Negative Documents

Topic	Number of negative documents in the training set	Number of interesting negative documents in the training set
101	16	0
102	64	63
103	50	6
104	74	54
105	21	7
106	40	3
107	58	0
108	50	0
109	21	19
Total	394	152

TABLE 15
Evaluation of Threshold Setting

Topic	Number of positive documents in the training set	Number of positive documents in the testing set	Number of positive documents whose relevance ≥ the thresholds in the testing set	Recall
101	7	307	4	1.3%
102	135	159	156	98.1%
103	14	61	22	36.1%
104	120	94	77	81.9%
105	16	50	47	94.0%
106	4	31	2	6.5%
107	3	37	0	0%
108	3	15	0	0%
109	20	74	72	97.3%

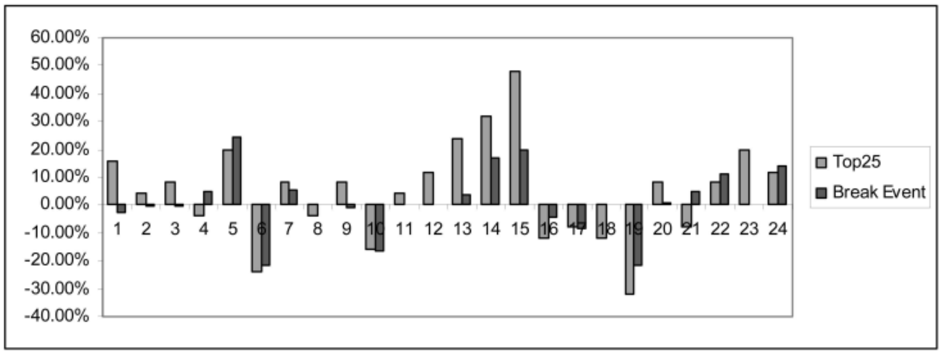

Fig. 8. Semisupervised evaluation for all topics: difference between *Round1* and $Rocchio$.

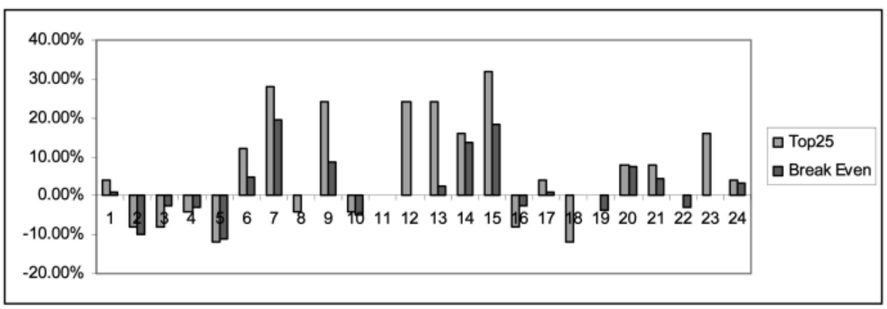

Fig. 9. Supervised evaluation for all topics: difference between *Round2* and $Prob2$.

could determine the premises and conclusions of rules, then rough sets [37], [38], [10] and random sets [18], [22] based decision rules could be used to show the meaning of discovered patterns. Also, we may research the similarity between discovered patterns [34], [45] to remove some redundant ones (e.g., nonclosed patterns [46]).

In implementation practices, data mining has been used in Web text mining, which refers to the process of searching through unstructured data on the Web and deriving meaning from it [5], [6], [11]. The main purpose of text mining was association discovery [2], where the association between a set of keywords and a predefined category (e.g., a term or a set of terms) was described as an association rule. Maximal patterns [5], [10], sequential patterns [42], and closed sequential patterns [47] were also used in text mining. However, the effectiveness of these approaches was even worse than Rocchio and probabilistic models when we tested them for mining user profiles using TREC data collections. The clustering-based mining technique was also applied to study semisupervised text classification, which uses a set of positive documents and the entire documents to build a classifier [17]. However, this method cannot be directly used for Web user profiles mining since it is impossible to obtain entire documents from the Web in advance for each topic.

The above methods provide some mechanisms to apply data mining techniques in text mining. However, the main drawback is that they cannot improve the effectiveness significantly since they failed to study the process of knowledge evolution. In this paper, we present an ontology mining technique to overcome the disadvantage.

The objective of ontology mining is fairly different to semiautomatic ontology engineering. The former concerns the automatic representation of discovered knowledge. The latter mainly develops tools to map, merge, and align existing ontologies [35], [44]. Some ontology mining algorithms have been mentioned in [29], [30], [50], [4], which currently are the discovery of the taxonomic backbone (e.g., hierarchical clustering [32]) and the nontaxonomic relation (e.g., association rules as mentioned before). The difficult problem here is how to formalize relationships between classes since most relationships (e.g., "part-of" relationship) are not well-known mathematical structures (e.g., lattices) [13].

In this paper, we use a new concept of association sets to formalize the relationships between classes. We also set up a reasoning model according to this kind of formalizations. In addition, we provide both theoretical and experimental evaluations for the model and the results demonstrate that our solution is achievable and promising.

8 Conclusions

There is no doubt that numerous discovered patterns can be found from the Web data using data mining techniques. However, it is ineffective to use the discovered patterns in Web user profile mining due to the ambiguities in the data values (terms). The consequent result is that we obtain some inappropriate discovered patterns and many discovered patterns include uncertainties. In this paper, we develop an ontology mining technique to provide a solution for this challenge. A discovered ontology in this research consists of two parts: the top backbone and the base backbone. The former illustrates the linkage between compound classes of the ontology. The latter illustrates the linkage between primitive classes and compound classes.

We set up a mathematical model to represent discovered knowledge on the ontology. We also present a novel method for capturing evolving patterns in order to refine the discovered ontology. In addition, we establish a formal method for learning how to assess relevance in the ontology in order to effectively use discovered knowledge. We have verified that the technique not only gains a better performance on both precision and recall, it also decreases the burden of online training.

The research is significant for WI since it makes a breakthrough by effectively synthesizing taxonomic relation and nontaxonomic relation in a mathematical model. It is also significant for data mining because it provides an approach for representation, application, and maintenance of discovered knowledge for solving real problems.

Acknowledgments

This paper was partially supported by Grant DP0556455 from the Australian Research Council. The authors also wish to thank the anonymous reviewers for their valuable comments.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval.* Addison Wesley, 1999.
[2] G. Chang, M.J. Healey, J.A.M. McHugh, and J.T.L. Wang, *Mining the World Wide Web: An Information Search Approach.* Kluwer Academic, 2001.
[3] M.S. Chen, J. Park, and P.S. Yu, "Data Mining For Path Traversal Patterns In A Web Environment," *Proc. 16th Int'l Conf. Distributed Computing Systems,* pp. 385-392, 1996.
[4] M. Craven, "Learning to Extract Symbolic Knowledge from the World Wide Web," *Artificial Intelligence,* vol. 118, pp. 69-113, 2000.
[5] R. Feldman et. al., "Maximal Association Rules: A New Tool for Mining for Keyword Co-Occurrences in Document Collection," *Proc. Third ACM SIGKDD Int'l Conf. Knowledge Discovery,* pp. 167-170, 1997.
[6] R. Feldman, I. Dagen, and H. Hirsh, "Mining Text Using Keywords Distributions," *J. Intelligent Information Systems,* vol. 10, no. 3, pp. 281-300, 1998.
[7] M.N. Garofalakis, R. Rastogi, S. Seshadri, and K. Shim, "Data Mining and the Web: Past, Present and Future," *Proc. ACM CIKM Int'l Workshop Web Information and Data Management,* pp. 43-47, 1999.
[8] D.A. Grossman and O. Frieder, *Information Retrieval Algorithms and Heuristics.* Kluwer Academic, 1998.
[9] N. Govert and G. Kazai, "Overview of the Initiative for Evaluation of XML Retrieval (INEX)," *Proc. First Workshop Initiative for the Evaluation of XML Retrieval,* pp. 1-17, 2002.
[10] J.W. Guan, D.A. Bell, and D.Y. Liu, "The Rough Set Approach to Association Rules," *Proc. Thrid IEEE Int'l Conf. Data Mining,* pp. 529-532, 2003.
[11] J.D. Holt and S.M. Chung, "Multipass Algorithms for Mining Association Rules in Text Databases," *Knowledge and Information Systems,* vol. 3, pp. 168-183, 2001.
[12] D.A. Hull and S. Roberston, "The TREC-8 Filtering Track Final Report," *Proc. Text Retrieval Conf. (TREC-8),* 1999.
[13] R. Janicki, "Towards a Mereological System for Direct Products and Relations," *Proc. Third Int'l Conf. Rough Sets and Current Trends in Computing,* pp. 113-122, 2002.
[14] K.S. Jones, "Information Retrieval and Artificial Intelligence," *Artificial Intelligence,* vol. 114, nos. 1-2, pp. 257-281, 1999.
[15] R. Kruse, E. Schwecke, and J. Heinsoln, *Uncertainty and Vagueness in Knowledge Based Systems.* Springer-Verlag, 1991.
[16] K.C. Lee, J.S. Kim, N.H. Chung, and S.J. Kwon, "Fussy Cognitive Map Approach to Web-Mining Inference Amplification," *Expert System with Applications,* vol. 22, pp. 197-211, 2002.
[17] X. Li and B. Liu, "Learning to Classify Texts Using Positive and Unlabeled Data," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 587-592, 2003.
[18] Y. Li, "Extended Random Sets for Knowledge Discovery in Information Systems," *Proc. Ninth Int'l Conf. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing,* pp. 524-532, 2003.
[19] Y. Li and Y.Y. Yao, "User Profile Model: A View from Artificial Intelligence," *Proc. Third Int'l Conf. Rough Sets and Current Trends in Computing,* pp. 493-496, 2002.
[20] Y. Li, C. Zhang, and J.R. Swan, "An Information Filtering Model on the Web and Its Application in *JobAgent*," *Knowledge-Based Systems,* vol. 13, no. 5, pp. 285-296, 2000.
[21] Y. Li and N. Zhong, "Ontology-Based Web Mining Model: Representations of User Profiles," *Proc. IEEE/WIC Int'l Conf. Web Intelligence,* pp. 96-103, 2003.
[22] Y. Li and N. Zhong, "Interpretations of Association Rules by Granular Computing," *Proc. Third IEEE Int'l Conf. Data Mining,* pp. 593-596, 2003.
[23] Y. Li and N. Zhong, "Capturing Evolving Patterns for Ontology-Based Web Mining," *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence,* pp. 256-263, 2004.
[24] Y. Li and N. Zhong, "Web Mining Model and Its Applications on Information Gathering," *Knowledge-Based Systems,* vol. 17, pp. 207-217, 2004.
[25] T.Y. Lin, "Database Mining on Derived Attributes," *Proc. Third Int'l Conf. Rough Sets and Current Trends in Computing,* pp. 14-32, 2002.
[26] B. Liu and K. Chang, *SIGKDD Explorations,* special issue on Web content mining, vol. 6, no. 2, pp. 1-4, 2004.
[27] Z.Y. Lu, Y.Y. Yao, and N. Zhong, "Web Log Mining," *Web Intelligence,* pp. 174-194, 2003.
[28] S.M. Madria, S.S. Bhowmick, W.K. Ng, and E.-P. Lim, "Research Issues in Web Data Mining," *Proc. First Int'l Conf. Data Warehousing and Knowledge Discovery,* pp. 303-312, 1999.
[29] A. Maedche, *Ontology Learning for the Semantic Web.* Kluwer Academic, 2003.
[30] A. Maedche, V. Pekar, and S. Staab, "Ontology Learning Part One—On Discovering Taxonomic Relations from the Web," *Web Intelligence,* pp. 302-319, 2003.
[31] H. Mannila and H. Toivonen, "Discovering Generalized Episodes Using Minimal Occurrences," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining,* pp. 146-151, 1996.

[32] C. Manning and H. Schuetze, *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

[33] J. Mostafa, W. Lam, and M. Palakal, "A Multilevel Approach to Intelligent Information Filtering: Model, System, and Evaluation," *ACM Trans. Information Systems,* vol. 15, no. 4, pp. 368-399, 1997.

[34] H. Motoda, "Active Mining," *Frontiers in Artificial Intelligence and Application,* no. 79, IOS Press, 2002.

[35] N.F. Noy and M.A. Musen, "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment," *Proc. 15th Nat'l Conf. Artificial Intelligence,* 2000.

[36] S.K. Pal and V. Talwar, "Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions," *IEEE Trans. Neural Networks,* vol. 13, no. 5, pp. 1163-1177, 2002.

[37] Z. Pawlak, "Rough Sets and Intelligent Data Analysis," *Int'l J. Information and Sciences,* vol. 147, pp. 1-12, 2002.

[38] Z. Pawlak, "In Pursuit of Patterns in Data Reasoning from Data—The Rough Set Way," *Proc. Third Int'l Conf. Rough Sets and Current Trends in Computing,* pp. 1-9, 2002.

[39] M. Perkowitz and O. Etzioni, "Adaptive Web Sites," *Comm. ACM,* vol. 43, no. 8, pp. 152-158, 2002.

[40] S. Robertson and D.A. Hull, "The TREC-9 Filtering Track Final Report," *Proc. Text Retrieval Conf. (TREC-9),* 2000.

[41] S. Schocken and R.A. Hummel, "On the Use of the Dempster Shafer Model in Information Indexing and Retrieval Applications," *Int'l J. Man-Machine Studies,* vol. 39, pp. 843-879, 1993.

[42] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys,* vol. 34, no. 1, pp. 1-47, 2002.

[43] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan, "Web Usage Mining: Discovery and Applications of Usage Pattern from Web Data," *SIGKDD Explorations,* vol. 1, no. 2, pp. 12-23, 2002.

[44] S. Staab and R. Studer, *Handbook on Ontologies.* Springer, 2004.

[45] S. Tsumoto and S. Hirano, "Visualization of Rule's Similarity Using Multidimensional Scaling," *Proc. Third IEEE Int'l Conf. Data Mining,* pp. 339-346, 2003.

[46] P. Tzvetkov, X. Yan, and J. Han, "TSP: Mining Top-K Closed Sequential Patterns," *Proc. Third IEEE Int'l Conf. Data Mining,* pp. 347-354, 2003.

[47] S.-T. Wu, Y. Li, Y. Xu, B. Pham, and P. Chen, "Automatic Pattern Taxonomy Exatraction for Web Mining," *Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence,* pp. 242-248, 2004.

[48] T.Y. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal, "From User Access Patterns to Dynamic Hypertext Linking," *Proc. Fifth Int'l World Wide Web Conf.,* 1996.

[49] Q. Yang, H. Zhang, I. Tian, and Y. Li, "Mining Web Logs for Prediction Models in WWW Caching and Prefetching," *Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 473-478, 2001.

[50] N. Zhong, "Representation and Construction of Ontologies for Web Intelligence," *Int'l J. Foundation of Computer Science,* vol. 13, no. 4, pp. 555-570, 2002.

[51] N. Zhong, J. Liu, and Y.Y. Yao, "In Search of the Wisdom Web," *Computer,* vol. 35, no. 11, pp. 27-31, Nov. 2002.

Yuefeng Li received the BSc degree in mathematics from Jilin University, China, the MSc degree in computer science from Jilin University, China, and the PhD degree in computer science from Deakin University, Australia. He is a senior lecturer at Queensland University of Technology, Brisbane, Australia. He has published more that 50 refereed papers and two books. He is an associate editor of the *International Journal of Pattern Recognition* and *Artificial Intelligence* and an associate editor of the *IEEE Intelligent Informatics Bulletin*. His current research interests include ontology and Web mining, Web intelligence, data mining, and multiagent systems.

Ning Zhong received the PhD degree in the interdisciplinary course on advanced science and technology from the University of Tokyo. He is currently head of the Knowledge Information Systems Laboratory and is a professor in the Department of Systems and Information Engineering at Maebashi Institute of Technology, Japan. He is also an adjunct professor in the International WIC Institute, Beijing University of Technology. He has conducted research in the areas of knowledge discovery and data mining, rough sets and granular-soft computing, Web intelligence, intelligent agents, brain-informatics, and knowledge information systems, with more than 150 journal and conference publications and 10 books. He is the editor-in-chief of the *Web Intelligence and Agent Systems* (IOS Press) and the *Annual Review of Intelligent Informatics* (World Scientific), an associate editor of the *IEEE Transactions on Knowledge and Data Engineering*, and *Knowledge and Information Systems* (Springer), a member of the editorial board of *Transactions on Rough Sets* (Springer), and of the Advanced Information and Knowledge Processing (AI&KP) book series (Springer), the Frontiers in AI and Applications book series (IOS Press), and editor (the area of intelligent systems) of the *Encyclopaedia of Computer Science and Engineering* (Wiley). He is the cochair of the Web Intelligence Consortium (WIC), the vice chair of the IEEE Computer Society Technical Committee on Intelligent Informatics, a member of the steering committee of IEEE International Conferences on Data Mining (ICDM), the advisory board of the International Rough Set Society. He has served or is currently serving on the program committees of more than 80 international conferences and workshops, including IEEE ICDM '02 (conference chair), IEEE ICDM '06 (program chair), IEEE/WIC WI-IAT '03 (conference chair), IEEE/WIC/ACM WI-IAT '04 (program chair), and IJCAI '03 (advisory committee member). He is a senior member of the IEEE and a member of the IPSJ, the JSAI, the IEEE Computer Society, the IEEE-SMC, the ACM, the AAAI, and the IRSS.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.