A New Digital Paper Search Paradigm Based on FCA

Haibin Yu[1], Chongyang Shi[1], Bai Yu[2], Chunxia Zhang[1], Ryan Hearne[1]

[1] School of Computer Science, Beijing Institute of Technology, China,
[2] QCIS, University of Technology, Australia,
{yuhaibin, cy_shi}@bit.edu.cn, yu.bai-3@ student.uts.edu.au, cxzhang@bit.edu.cn, itsryanhearne@gmail.com

Abstract

This paper proposes a new digital paper search paradigm that controls the diversity of keyword-based search query topics based on Formal Concept Analysis (FCA). During pre-querying, papers are assigned to pre-specified, lattice-based context patterns built by a selected partial dataset, and query-independent lattice context scores are attached to papers with respect to the assigned lattice contexts. When a query is executed, the relevant lattice contexts are selected, a search is performed within the selected lattice contexts, the context scores of the papers are revised to become relevancy scores with respect to the query and the lattice context they are in, and the query outputs are ranked within each relevant lattice context. In this way, we (1) provide FCA with a path to deal with middling or larger amounts of documents, (2) minimize query output topic diversity and reduce query output size, (3) decrease the user's time spent scanning query results, and (4) increase query output ranking accuracy. Using China National Knowledge Infrastructure (CNKI) publications as the testbed, our experiments indicate that the proposed lattice context-based search approach produces search results with up to 50% higher precision, and reduces the query output size by up to 60% more than a CNKI search.

Keywords: Paper retrieval, Formal concept analysis, Concept lattice, Query context

1 Introduction

Currently, via the public (e.g. Google Scholar [1]), commercial (e.g. ACM [2]) or free (e.g. CiteSeer [3]) scholastic search engines, a myriad of papers are available to people in the research or study fields. Finding specific documents, such as papers, in the entirety of the scholastic data seas can be a challenge and is an important issue. Our goals are to achieve both efficient and effective searches.

Formal Concept Analysis (FCA) was proposed by Wille in 1982 [4], and, since the 90s, FCA has been integrated with basic Information Retrieval (IR) techniques to build more comprehensive systems for the information access field. A concept lattice has been used as a support structure for IR. A number of researchers have proposed lattice-based structures for IR [5-10], but most of whom have only conducted research projects or practices using a variety of small datasets showing the mathematical aspect of FCA. A search application using FCA aiming at middling or more amounts of documents could not be found.

In order to (1) provide FCA with a path to deal with middling or larger amounts of documents, (2) minimize the scope of query output for large numbers of papers, (3) provide controlled ways of eliminating query output topic diversity, and (4) present a new method that can effectively rank query output papers. We propose a new digital paper search paradigm, called the Formal Concept-Analysis-Based Search (FBS) approach, which is a context-based search model using a lattice, as follows:

(1) We perform two query-independent pre-processing steps before any query session starts, assigning papers to lattice based contexts; and compute the lattice (importance) scores for papers. Therefore, each lattice context contains two types of information: (a) the intrinsic lattice information for the paper set (object) and the term set (attribute) owned by the paper set, and (b) the context score for each paper.

(2) Then, at the time of the search, we perform the following steps:

(a) Automatically select the search lattice contexts.

(b) Perform the search within the selected lattice contexts.

(c) Within each lattice context, compute the relevancy scores for the located papers, re-rank the search results, and return the located papers.

With the FBS approach, (1) search input includes only papers residing in the selected lattice contexts as opposed to all papers, (2) search output is enhanced by a highly useful, context-based paper classification, (3) topic diffusion across search results is controlled, and (4) query output sizes are reduced to include only the search results in the lattice contexts of interest.

Since the FBS approach performs a search within the selected lattice contexts, some important results might be missing if they are not in the selected lattice

*Corresponding Author: Chongyang Shi; E-mail: cy_shi@bit.edu.cn
DOI: 10.3966/160792642018081904013

contexts. As an alternative, step 2 of the FBS approach can be modified to include all search results (FBSall) as follows:

(1) Automatically select the search lattice contexts.

(2) Perform the search across all papers to select the publications to be returned.

(3) For the returned papers that reside in the selected lattice contexts, compute the relevancy scores of these papers in each lattice context.

(4) Re-rank the search results and return the located papers.

FBS is a contextual IR model, and contextual IR models always faces many new challenges, such as context modeling, contextual document ranking and the system's effectiveness evaluation, and, in our work, we present several approaches to deal with them in Section.4.

Section 2 summarizes and compares our approach with the related work. Section 3 introduces a kind of concept lattice using the paper context. Section 4 is an overview of our lattice context-based search approach. Section 5 presents how to classify papers with scores for lattice contexts. In Section 6, we describe the methods used to select the search contexts for a given query term. Section 7 explains alternatives used to search and to rank the search results within contexts, and gives ways of merging results from multiple contexts. Section 8 presents the experiment setup and experiment results, respectively. Section 9 concludes this article.

2 Related Works

FCA can be used broadly in IR. In 1996, Carpineto [10] presented Galois, a system that automates and applies FCA with respect to IR via browsing. He also describes a prototype user interface for browsing using the concept lattice of a document-term relation. And, in 2005, Carpineto [9] further discussed the application of FCA in IR from different aspects. Qadi et al. [7] describes a mechanism based on FCA that determines semantical relations during the queries, and allows a reorganization, in the shape of a lattice of concepts, of the answers provided by a search engine. It proposes an incremental algorithm based on Galois lattice for the IR. This algorithm allows a formal clustering of the data sources, and the results that it retrieves are classified by order of relevance.

While in the process of IR preparation, documents can be organized by concept lattice, which has a fine view of document-document relations. Kovics and Baranyi [11] developed a document query system that clusters the documents into groups using a generated concept lattice. The users can start the query by entering a set of keywords. Then the system returns the concepts closest to the query vector. The users get a list of neighboring concepts too, and thus they can select the way to navigate to reach the result document set.

After fulfilling a search request, FCA can also be applied to clustering and ranking the query results. Zhang [12] proposes a method based on FCA to build a two-level hierarchy for the retrieved search results of a query. After the formal concepts are extracted using FCA, the proposed algorithm will extract the concepts most relevant to the query, and a two-level hierarchy is built and presented to the user. Cigarrn et al. [13] present the JBraindead Information Retrieval System, which combines a free-text search engine with online FCA to organize the results of a query. This paper focuses on the automatic selection of attributes and shows that conceptual lattices can be very useful for grouping relevant information in free-text search tasks. Tang et al. [14] discuss and compare Concept Lattice-based Ranking (CLR) and presents a combination CLR approach by measuring the similarity among the query, user profile and document according to the relation between the query and user interest, based on the concept lattice. The experiment shows that the documents retrieved by their combination CLR approach achieve a higher measure of precision than the traditional CLR approach.

Rough set theory can be employed in combination with Fuzzy Formal Concept Analysis (FFCA) to perform a semantic Web search to discover information in the Web. According to this proposal, the required data is not modeled by any formal concept, but the user can search for and discover information in the Web that is closer to his/her preferences by following a twofold approach [5]. De Maio et al. [6] present an ontology-based retrieval approach that supports data organization and visualization, and provides a friendly navigation model. It exploits the fuzzy extension of the FCA theory to elicit conceptualizations from datasets and generate a hierarchy-based representation of the extracted knowledge. An intuitive graphical interface provides a multi-faceted view of the built ontology. Through a transparent query-based retrieval, the end users navigate across concepts, relations and population.

Some FCA tools have been developed in IR, such as Search Sleuth [15], which is a program developed to experiment with the automated, local analysis of a Web search using FCA. Search Sleuth extends a standard search interface to include a conceptual neighborhood centered on a formal concept derived from the initial query. The neighborhood for the concept derived from the search terms is bordered with its upper and lower neighbors representing more general and special concepts respectively. The focus is on understanding the use and meaning of proximity and semantic distance in the context of IR using FCA.

Most of the above research and applications face the problem of a data sea in which we need to spend too much time to build a search concept lattice and so on. The first important issue is to make the FCA more practicable and usable in a real search environment. In

our work, a new means is presented to solve this issue.

3 Formal Concept by Papers' Context

In FCA, the use of object and attribute is indicative because in many applications it may be useful to choose object-like items as formal objects and then choose their features as formal attributes [6]. For instance, in our paper-searching work, papers could be considered to be object-like and terms considered to be attribute-like. The formal context (here "formal context" just means a table distinguishing the lattice context) is often represented as a cross table: the rows represent the papers and the columns are terms; the intersections represent the relations between them. In this paper, the papers and terms play the role of objects and attributes, respectively.

3.1 Formal Context of a Paper

Let U and A be two finite and nonempty sets. The U elements are papers (objects) from the dataset, and the A elements are terms (attributes) extracted from the papers. The relationships between papers and terms are described by binary relation I between U and A, which is a subset of the Cartesian product of $U \times A$. For a pair of elements $x \in U$ and $a \in A$, if $(x,a) \in I$, also written as xIa, we say that paper x has the term a, or the term a is possessed by paper x. Here, $(x,a) \in I$ is denoted by 1, and $(x,a) \notin I$ is denoted by 0. Thus, the formal context of a paper dataset can be represented by a table only containing 0 and 1.

Paper $x \in U$ has the set of terms:

$$xI = \{a \in V \mid xIa\} \in A \quad (1)$$

Term $a \in V$ is possessed by the set of papers:

$$Ia = \{x \in U \mid xIa\} \subset U \quad (2)$$

The triplet (U,A,I) is called a binary formal context. For simplicity, we only consider the binary formal context in the subsequent discussion.

3.2 Formal Concept Analysis of Papers' Formal Context

Remark 1. For formal context (U,A,I), for set $X \subseteq U$ of papers and set $B \subseteq A$ of terms, we define a set-theoretic operator *[16]:

$$X^* = \{a \in A \mid \forall x \in X, (x,a) \in I\} \quad (3)$$

It associates subset of terms X^* to the subset of papers X. Similarly, for any subset of terms $B \subseteq A$, we can associate a subset of papers B^*:

$$B^* = \{x \in U \mid \forall a \in B, (x,a) \in I\} \quad (4)$$

X^* is the set of all the terms shared by all the papers in X, and B^* is the set of all the papers that fulfil all the terms in B.

Remark 2. Let (U,A,I) be a paper's formal context. A pair (X,B) is called a formal concept deduced from the paper's context (for short, a concept of (U,A,I)), if and only if $X \subseteq U$, $B \subseteq A$, $X^* = B$ and $B^* = A$. X is called the extension and B is called the intension of (X,B). The set of all concepts in (U,A,I) is denoted by $L(U,A,I)$.

The operator * has the following terms: for all of X_1, X_2, $X \subseteq U$, and all of B_1, B_2, $B \subseteq A$,

$$X_1 \subseteq X_2 \Rightarrow X_2^* \subseteq X_1^*, B_1 \subseteq B_2 \Rightarrow B_2^* \subseteq B_1^* \quad (5)$$

$$X \subseteq X^{**}.B \subseteq B^{**} \quad (6)$$

$$X^* = X^{***}, B^* = B^{***} \quad (7)$$

$$X \subseteq B^* \Leftrightarrow B \subseteq X^* \quad (8)$$

$$(X_1 \cup X_2)^* = X_1^* \cap X_2^*, (X_1 \cup B_2)^* = B_1^* \cap B_2^* \quad (9)$$

$$(X_1 \cap X_2)^* \supseteq X_1^* \cup X_2^*, (B_1 \cap B_2)^* \supseteq B_1^* \cup B_2^* \quad (10)$$

Remark 3. Let (U,A,I) be a paper's formal context, and (X_1,B_1) and (X_2,B_2) be concepts of the context; the concepts of a formal context (U,A,I) are ordered by:

$$(X_1,B_1) \le (X_2,B_2) \Leftrightarrow X_1 \subseteq X_2 (\Leftrightarrow B_1 \supseteq B_2) \quad (11)$$

Where (X_1,B_1) and (X_2,B_2) are concepts, (X_1,B_1) is called a sub-concept of (X_2,B_2), and (X_2,B_2) is called a super-concept of (X_1,B_1). The notation $(X_1,B_1) \prec (X_2.B_2)$ denotes the fact that if $(X_1,B_1) \prec (X_2.B_2)$ and concept (Y,C) does not exist such that $(X_1,B_1) \prec (Y,C) \prec (X_2.B_2)$, then (X_1,B_1) is called a child-concept (immediate sub-concept) of (X_2,B_2) and (X_2,B_2) is called a parent-concept (immediate super-concept) of (X_1,B_1); this is denoted by $(X_1,B_1) \prec (X_2.B_2)$.

Remark 4. Let (U,A,I) be a formal context, then $L(U,A,I)$ is a complete lattice. The infimum and supremum are given by:

$$(X_1,B_1) \wedge (X_2,B_2) = (X_1 \cap X_2, (B_1 \cup B_2)^{**}) \quad (12)$$

$$(X_1,B_1) \vee (X_2,B_2) = ((X_1 \cup X_2)^{**}, B_1 \cap B_2) \quad (13)$$

Remark 5. Let $L(U,A_1,I_1)$ and $L(U,A_2,I_2)$ be two concept lattices. If, for any $(X,B) \in L(U,A_2,I_2)$,

$(X^{'},B^{'}) \in L(U,A_1,I_1)$ exists such that $X^{'}=X$, then $L(U,A_1,I_1)$ is said to be finer than $L(U,A_2,I_2)$, which is denoted by:

$$L(U,A_1,I_1) \le L(U,A_2,I_2) \quad \textbf{(14)}$$

If, in addition, we say that the two concept lattices are isomorphic, this is denoted by:

$$L(U,A_1,I_1) \cong L(U,A_2,I_2) \quad \textbf{(15)}$$

Example 1: A formal context (U,A,I) is given in Table 1, where $U=\{x_1.x_2,\ldots,x_n\}$ and $A=\{a_1,a_2,\ldots,a_m\}$:

Table 1. A small paper dataset formal context

	a	b	c	d	e
1	0	1	0	0	1
2	1	0	1	0	0
3	1	0	1	1	1
4	0	1	0	0	1
5	1	0	0	0	0
6	1	1	0	0	1

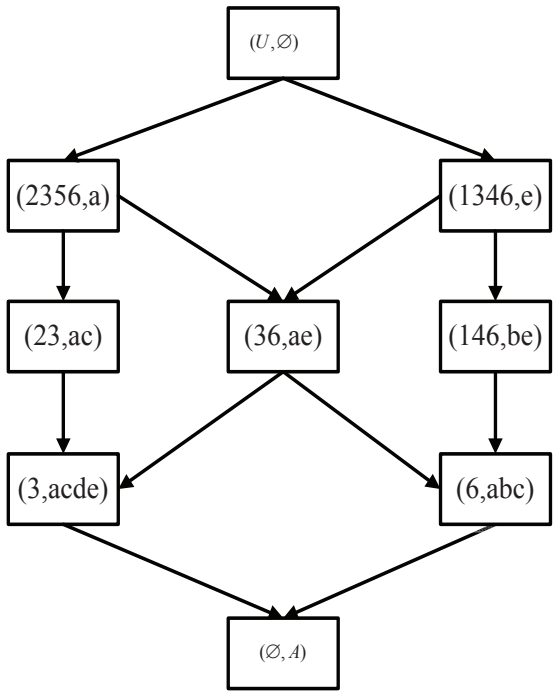


Figure 1. The paper concept lattice of (U,A,I) from Example 1

4 Approach Overview

Figure 2 summarizes all the processes for the FBS and FBSall methods. The following sequence of algorithms is used to describe our proposed lattice context-based search approach. The explanations for the algorithms will be detailed in the next few sections:

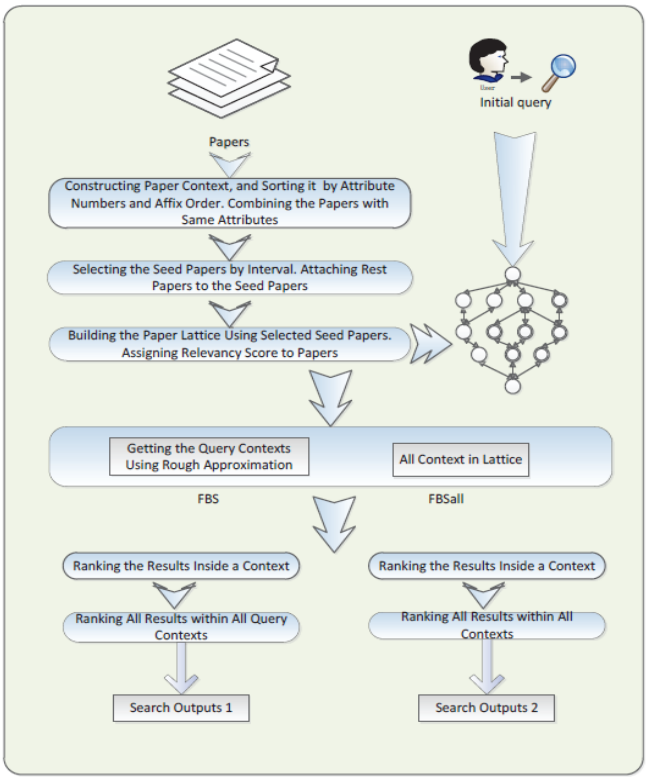


Figure 2. Whole Process for the FBS and FBSall Methods

(1) "Build_Contexts" and "Locate_Papers": Construct lattice context patterns from the part of the dataset selected using a manually chosen interval and use these patterns to locate the rest of the papers for the lattice context (a lattice context is a node of the concept lattice).

Note that "Build_Contexts" and "Locate_Papers" are pre-executed and not dependent on the queries.

1. **Algorithm Pattern_Based Paper_Lattice_Context Assignment**
2. **Input:** all the papers
3. **Output:** lattice context, which contains
4. the assigned papers
5. Construct lattice context patterns from the selected dataset
6. **for**each paper p **do**
7. **for**each lattice context c_i in patterns **do**
8. **if** c_i is p's neighbor **then**
9. add p to c_i;
10. **end for**
11. **end for**

This algorithm will be introduced in Section 5.

(2) Evaluate_Query

(2.1) Select_Query_Lattice_Contexts: Using the search term, we first select the lattice contexts to search for.

```
1. Algorithm Select_Query_Contexts
2. Input:
3.      q : query term (possibly multiple words)
4.      t : similarity threshold
5. Output: a set of query lattice contexts for query q
6.   foreach lattice context c_i do
7.     compute q's lower attribute approximations and
8.     corresponding approximated object set;
9.     compute Sim(q, c_i);
10.       if Sim(q, c_i) ≥ t
11.    then
12.       add c_i to query context set;
13.  end for
```

This algorithm will be introduced in Section 6.

(2.2) Perform_Search_per_Selected_Lattice_Context: The search within each lattice context is performed using a text-based measure of similarity between the given query term and the papers in the selected query context. The paper results are ranked using their relevancy (scores) to the query. The relevancy score of a paper in a lattice context is defined as a combination of the paper-to-query matching score and the pre-computed lattice context score for the paper.

```
1. Algorithm Search_FBS
2. Input:
3.    q : query term (possibly multiple words)
4.    query_contexts: set of selected query contexts
5. Output: a set of search results within selected query 6.
   contexts
7.  foreach paper p in each lattice context c in
8.    query_contexts do
9.   compute sim(p, q) as a text based similarity
10.       between p and q;
11.  compute relevancy_score(p, q, c) as a
12.    combination of sim(p, q) and the lattice
13.    context score of p in c;
14.  end for
15. Rank papers in each lattice context in descending
16.    order of their relevancy_scores and return;
```

This algorithm and related definitions of $sim(p,q)$, $relevancy\_score(p,q,c)$ and $sim(p,q)$ will be introduced in Section 6 and Section 7.

(2.3) Merge_Query_Results: When multiple lattice contexts are selected for a search, the results are displayed separately under different lattice contexts. In the case that the user wants to merge these results, a merging function that assigns only one aggregate score to each paper is presented. The new aggregate score of a paper is computed using (a) the relevancy score of the paper for the query in each lattice context, and (b) the similarity between each lattice context and the query.

```
1. Algorithm Merge_All_Results
2. Input:
3.      q : query term
4.      lattice_context_relevancy: list of similarity
5.      scores of each lattice context to the query term
6.      paper_context_relevancy: list of relevancy
7.      scores of each paper in each lattice context
8.      to the query term
9. Output: array of search results from all selected
10.    lattice context.
11.    foreach paper p in the output of q do
12.    forselected lattice context c_i where paper p
13.       resides do
14.    compute across_relevancy_score(p) as a
15.    combination of lattice_context_relevancy(c_i, q)
16.       and paper_context_relevancy(p,c_i,q);
17.      Add p and across_relevancy_score(p)
18.      to the merged_result;
19.    end for
20.  end for
21.  rank merged_results and return;
```

5 Classifying Papers with Scores as Lattice Contexts

This section presents some details for automatically locating the papers of lattice contexts and scores, assigning the algorithms shown in Section 4.

5.1 Lattice Context Pattern Building and the Papers' Lattice Context Assignment

Our approach constructs lattice context patterns from a partial data set and uses those patterns to locate the rest of the paper set of the lattice context. Lattice context pattern building is a process for actually building concept lattices and every pattern is a node (or a formal concept) of the concept lattice. We select the typical terms (in our experiments, we select the 40 most frequent terms from the key-word part of the papers) from these papers as the attributes of the papers' formal context to construct the paper' concept lattice. Traditionally, the concept lattice's building depends on the entire dataset. Due to having too many papers in the dataset, it needs a lot of time to construct the lattice in which there are enormous nodes from which we could not find the results easily. Then we just use part of the dataset to generate the patterns.

First, we sort the papers by attribute number and order (the attributes' affix order), and then we combine the papers with the same attributes. The papers are then selected using a manually appointed interval through which the scale of the data set is cut down, as is the time consumption and the number of lattice nodes. We name the selected papers as seeds, then the rest of the unselected papers will be assigned to the lattice context after the completion of lattice building only the seed papers. If we use the incremental lattice building

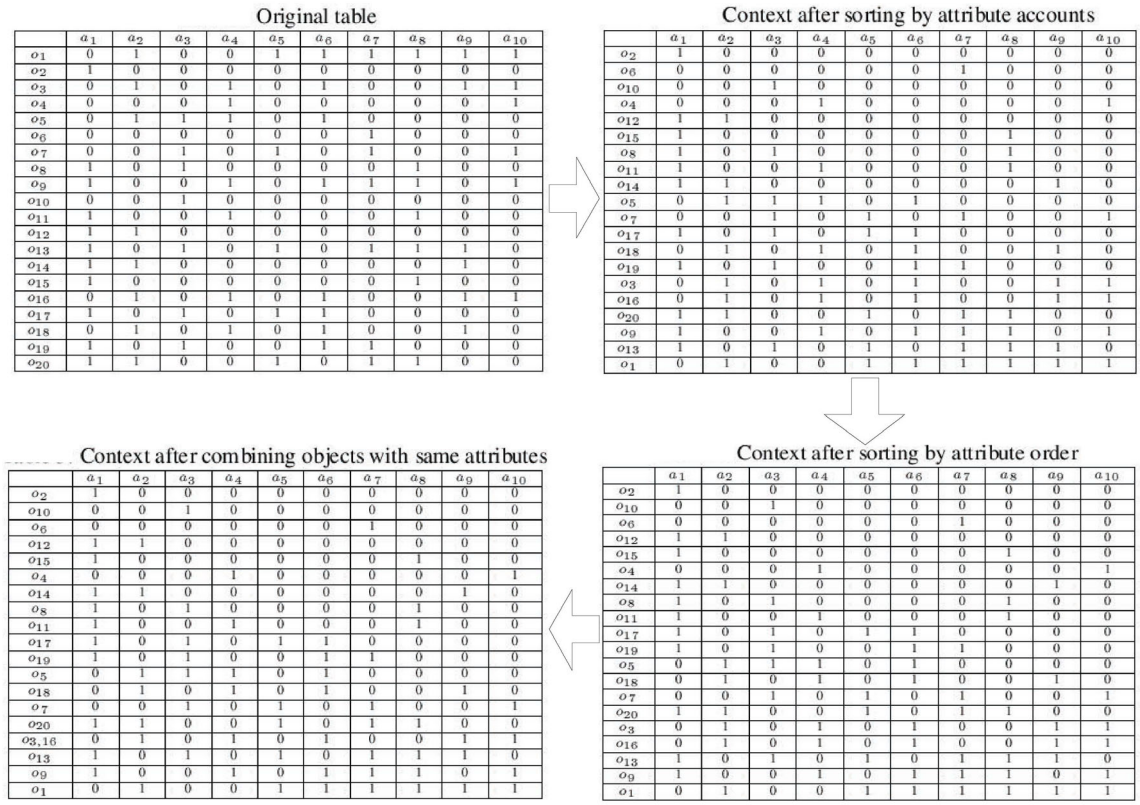


Figure 3. Sort the papers' formal context by attribute number and attribute affix order

algorithms [17], when new papers are added to the lattice patterns, according the incremental lattice building method, the patterns for the lattice contexts should be updated. To avoid this updating problem and ensure all the papers are included by the lattice contexts, the rest of the papers are assigned to the neighboring papers, such as the "4" and "4" in Figure 4, to the seed papers by which all lattice contexts could hold the line.

Example 2: In the top-left corner of Figure 3 is the original papers' formal context. The table of the papers' formal context is sorted by attribute number (top-right corner) and attribute affix order (bottom-right corner). Finally, the papers with the same attributes are merged into one line in bottom left corner.

After sorting and merging the papers, the formal contexts with intervals 3 and 4 are as shown in Figure 4 and Figure 5.

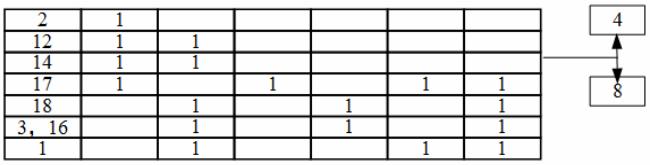


Figure 4. Interval 3

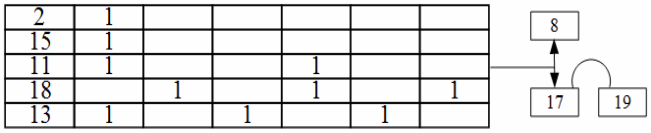


Figure 5. Interval 4

As shown in Figure 4 and Figure 5, the rest of the unselected papers are assigned to the neighboring selected papers. With the exception of the first and last papers, the previous $\left\lfloor \frac{s-1}{2} \right\rfloor$ paper and the following $\left\lceil \frac{s-1}{2} \right\rceil$ papers are distributed to one selected paper (where s is the interval).

Table 2 shows some numerical values for the papers' formal context by different intervals. Where N_o is the number of papers, N_{ln} is the number of lattice nodes, T_c is the time consumed to construct the lattice and AS is the average similarities between the attached papers and seed papers. We employ some strategies to choose the interval where (1) AS is more than 0.5 and (2) N_{ln} is less than 500,000. From the Table 2, we can see the interval 5 is eligible for the requirements.

Table 2. Some numerical values of paper formal context by different interval

Interval	N_0	N_{ln}	Tc (Minutes)	AS
1	10000	8455398	2853	1
3	3334	938745	1087	0.79
4	2501	549373	623	0.68
5	2001	458292	489	0.63
6	1667	283943	399	0.61
7	1429	194328	317	0.54
8	1251	83049	252	0.51
9	1112	31987	209	0.41
10	1001	24745	181	0.33

5.2 Assigning Lattice Context Scores to Papers

In [18], the lattice context score of a paper in each context is computed using text-based similarity measures based on the Term Frequency-Inverse Document Frequency (TFIDF) model [19]. In each lattice context c, a paper $p's$ lattice context score is defined as the text-based similarity score between the lattice context's centroid and p. The centroid of the lattice context is computed by averaging all of the papers scores in the lattice context. In other words, the papers in the lattice context that are highly similar to the centroid of c receive high lattice context scores. Based on the papers in a lattice context, one centroid is constructed. After counting the occurrences of a term in each paper, Shi et al. [18] approach is used to determine the average score for all of the occurrences, which is stored as an attachment within the lattice context. The average occurrences score for the centroid is defined as:

$$\text{Avg\_Occurrences\_Score}=\frac{\sum_M n_i}{N*M} \tag{16}$$

Where n_i is the number of times the lattice context term appears in paper i, and N is the length of a vector representing the paper. M is the number of papers in the lattice context.

$$Score(p)=sim(p_c,p) \tag{17}$$

Where p_c is the centroid of c, and $sim(p_c,p)$ is the text-based similarity between p and p_c.

However, the centroid of a lattice context needs to be frequently updated after adding each new paper. Calculating the scores of the papers can be tedious, so, in our case, a direct score that has been modified for a paper in the lattice context is obtained without getting the centroid and comparing the other papers with the centroid. Then the lattice context score of p in c is computed as:

$$Context\_Score=\frac{\sum_{k=1}^{N} weight_k}{\sum_{i=1}^{M}\sum_{j=1}^{N} weight_{ij}}*\frac{N}{S} \tag{18}$$

Where N is number of terms in the lattice context and M is the number of papers in the lattice context. S is the number of all terms in the paper p, not merely in the lattice context, $\sum_{k=1}^{N} weight_k$ is the weight of all the terms for paper p in the lattice context, and $\sum_{i=1}^{M}\sum_{j=1}^{N} weight_{ij}$ is the weight of all of the terms for all of the papers in the lattice context. Here the TFIDF model is also used for the weights of terms in papers.

Next, the attached papers' scores for seeds can be calculated using a text-based cosine value, which are computed between the papers' and the seed's vectors. According the seed's lattice context score, the attached paper's score is shown as:

$$Attach\_Score=S_{\text{score}}-S_{score}*\frac{1-SR_{score}}{1+SR_{score}} \tag{19}$$

Where S_{score} is the lattice context score for a seed, and SR_{score} is the cosine value between the attached paper and the seed.

6 Selecting Lattice Contexts for a Search Request

A context-based search query could be any set of keywords. In [20], we are introduced to a novel similarity evaluating model based on rough formal concept analysis and the information content similarity method. Given an arbitrary query term, it can therefore be viewed as an undefinable set of attributes in a formal concept. Following the theory of a rough set, such a set of attributes can be approximated by a definable set of attributes; namely, the intentions of formal concepts. Since a lower approximation is the greatest definable set for the concept, we build our similarity measure on the lower approximation. In this way, the particular contexts to query are selected.

Remark 6. For a query term q, its lower attribute approximations and corresponding approximated paper set are defined as:

$$q_{LA}=\text{int}\,ent(\wedge\{(X,Y)\in L|q\subset Y\}) \tag{20}$$

$$q_{LO}=extent(\wedge\{(X,Y)\in L|q\subset Y\}) \tag{21}$$

The similarity model between the term and the lattice context c, which is expressed by lattice information and (O,D) based on the lower attribute approximations is:

$$Sim(q,(O,D)) = \omega \frac{|q_{LO} \cap O|}{|(q_{LO} \cap O)| + (m_a - l_a)} + (1-\omega) \frac{|q_{LA} \cap D|}{|(q_{LA} \cap D)| + (n_a - r_a)} \quad (22)$$

You can find more details of the similarity model in [20].

Finally, given a query term q, and using the above similarity model, then those lattice contexts (formal concepts) with sufficiently high similarity (higher than a threshold t) are selected to be the query lattice contexts for q.

7 Search and Ranking Search Results

After mapping a given query to a set of query contexts, we perform the search and rank the search results within these lattice contexts. The search results returned from the lattice-based search are ranked using the irrelevancy scores with respect to the lattice context and the query term. The relevancy score of paper p to query q in lattice context c_i is computed as:

$$R(p,q,c_i) = w_{context}.Context\_Score(p,c_i) + w_{matching}.Text\_Matching\_Score(p,q) \quad (23)$$

Where $Context\_Score(p,c_i)$ is the lattice context score for p in lattice context c_i, text matching $Score(p,q)$ computes the similarity between p and q, and $w_{context}$ and $w_{matching}$ are the weights of the lattice context score and the text matching score, respectively. $w_{context} + w_{matching} = 1$. By default, we define $w_{matching} \succ w_{context}$. In this definition, the text-matching scores calculated between the query term and the papers are considered to be more important than the lattice context scores of the papers. However, the weights can be adjusted based on users' preferences. For example, if the user wants to increase the significance of the lattice contexts, $w_{matching}$ will be reduced while $w_{context}$ will be increased, and the search results within the lattice contexts will be ranked with respect to the new weights. In our work, we give an experimental adjustment to $w_{context}$ using this to get a higher precision in Subsection 8.3.

However, the users may want to view a single result set independent of the individually searched lattice contexts. To effectively rank search results for the latter case, the scores for a paper residing in multiple lattice contexts need to be merged into a final score. When appearing in multiple lattice contexts, paper p's overall relevancy score $R(p,q)$ for the query q is computed using (1) the relevancy score of p to q in each lattice context, and (2) the relevancy of each lattice context containing p to q, as follows:

$$R(p,q) = \frac{\sum_{i=1}^{np} (w_{Paper\,Relevancy} R_1(p,q,c_i) + w_{context} R_2(c_i,q))}{n_p} \quad (24)$$

Where $R_1(p,q,c_i)$ is the relevancy score of p to q in the lattice context c_i, $R_2(c_i,q)$ is the relevancy score of the lattice context c_i to the query q, np is the number of lattice contexts that contain p, $w_{Paper\,Relevancy}$ and $w_{context}$ are the weights of R_1 and R_2, respectively, and $w_{Paper\,Relevancy} + w_{context} = 1$. We define $w_{Paper\,Relevancy} \succ w_{context}$ (i.e. we used $w_{Paper\,Relevancy}$=0.6 and $w_{context}$=0.4 in the experiments).

8 Experimental Setup

We downloaded, parsed and populated our dataset with information from 10,000 full-text CNKI [21] papers. All selected papers came from the computer science area of IR. The 40 most frequent terms extracted from the keyword part of these papers were selected as the attributes to construct the papers' concept lattice, as shown in Table 3.

Table 3. Top 40 terms

Information Retrieval	Query	Data Retrieval	Searching
Information	Knowledge	Relevance	Web
Document	Term Weighting	Browsing	Pulling
Retrieval			
Filtering	Full Text	Stop Word	Stemming
Text Operation	Indexing	Logs	Clustering
Inverted	User Need	Query	Likelihood
File		Operation	
User	Human	Textual	Retrieval
Feedback	Computer Interaction	Images	Model
Evaluation	Visualization	Interface	Multimedia
Modeling	Parallel	Navigation	User Interface
Literature	Push	User Task	Scanning

8.1 Accuracy Evaluation

To evaluate the accuracy of the lattice context-based search approach, the recall and precision scores for the selected queries were used. Given a search term t as a query, its recall and precision are defined as:

$$Recall_t = \frac{|S_t \cap R_t|}{|R_t|} \quad (25)$$

$$Precision_t = \frac{|S_t \cap R_t|}{|S_t|} \quad (26)$$

Where S_t is the search result set for query term t, and R_t is the correct answer set for t. In addition to

recall and precision, we used the harmonic mean F1, which combines recall and precision. F1 is defined as:

$$F1_t = \frac{1}{\frac{1}{\mathrm{Re}call_t} + \frac{1}{\mathrm{Pr}ecision_t}} \quad (27)$$

8.2 AB-evaluating Set

To evaluate and compare the different approaches for keyword-based querying, clearly, the best approach is to obtain true answer sets for queries manually via domain expert judgments. However, such an approach is not always available and precludes using large numbers of queries to evaluate the overall methodology. Thus, we developed an approach to find the artificially built evaluating set (AB-evaluating set) of a query automatically. The AB-evaluating set was used to evaluate queries with no human judgments on their search results. Through domain expert evaluations of a small number of queries, we refined the AB-evaluating set creation process, and manually verified its correctness. Then the AB-evaluating set was used extensively in the experiments to evaluate the search query recall and precision scores.

AB-evaluating set construction. To construct an AB-evaluating set, we use an approach similar to the pearl-growing search strategy [22]. That is, we locate a highly relevant paper set for a given query and expand it iteratively through a highly compute-intensive expansion process. Given a query q, the database is queried for papers using a text-based similarity measure, and papers with similarity scores above threshold t are included in the initial answer set S1. By utilizing a high value of t, we ensure that the papers in S1 are highly relevant to the query term. After our initial construction, we expand S1 by using a citation-based approach.

This approach expands the AB-evaluating set with any citations for a paper in S1. Since a paper usually cites or is cited by other papers that are relevant to it, citations of a paper in S1 are potentially relevant to the query term. There are two approaches involving the citation-based expansion.

(1) **Text-based expansion:** This approach uses the text-based similarity measure to locate additional papers. Since a paper in S1 is highly relevant to the keyword query, papers that are cited in the paper are also potentially relevant to the query. Thus, papers cited with high similarity scores to p are added to the AB-evaluating set.

(2) **Citation-similarity-based expansion:** Citation similarity [23] is computed using co-citation [24] and bibliographic coupling [25]. Bibliographic coupling gives a high similarity score to a pair of papers $(p1, p2)$ with a large number of common citations. Co-citation gives a high similarity score to a pair of papers $(p1, p2)$ if the number of papers that cite both $p1$ and $p2$ is large. In this approach, papers in the database with high citation-similarity scores to a publication in S1 are added to the AB-evaluating set. Citation similarity [23] is computed as follows:

$$Sim_{Citation}(p1, p2) = BitWeight * Sim_{bib}(p1, p2) + (1 - BibWeight) * Sim_{coc}(p1, p2) \quad (28)$$

Where $p1$ and $p2$ are papers, $p1 \in S1, p2 \notin S1$, Sim_{bib} is the bibliographic coupling score, Sim_{coc} is the co-citation score, $BibWeight$ is the bibliographic coupling weight, $CocWeight = 1 - BibWeight$ is the co-citation weight, and $0 \le BibWeight \le 1$. Sim_{bib} is defined as:

$$sim_{bib}(p1, p2) = \frac{k_1}{M_b} \quad (29)$$

Where k_1 is the number of common citations between $p1$ and $p2$, and M_b is the maximum number of common citations between any pair of papers in the database. Sim_{coc} is defined as:

$$sim_{\mathrm{coc}}(p1, p2) = \frac{k_2}{M_{\mathrm{c}}} \quad (30)$$

Where k_2 is the number of papers that co-cite $p1$ and $p2$, and M_c is the maximum number of papers that co-cite any pair of papers in the database.

AB-evaluating set verification. We manually verified the AB-evaluating set accuracy in terms of precision. From all of the papers' attributes in the lattice context, we randomly chose ten search terms as a test set. Then the AB-evaluating set for each search term was constructed.

From our manual evaluation, the noise of the expansion (i.e. changes in the accuracy due to the expansion) depends on the choice of the initial set (S1). More specifically, when S1 includes only papers with high similarity scores to the query (i.e. the threshold for S1 is high), the papers retrieved after the expansion steps are at least 95% accurate. When relaxing the threshold for S1, the accuracy is reduced both in the initial set S1 and in the paper set from the expansion step. All of the expansion steps from the high-threshold S1 produce results with higher accuracy than S1 itself, which has a lower threshold.

8.3 Adjusting the $w_{context}$ in Relevancy Score

We use the precision as the criterion on which to adjust $w_{context}$ with different points. Adjusting $w_{context}$'s is an interactive training process between $w_{context}$ and precision. The precision values illustrated in Figure 6 are averages from varying the $w_{context}$ from 0.1 to 1.0

throughout the testing of the CNKI paper set with the resulting papers from the top 50 to 150 terms.

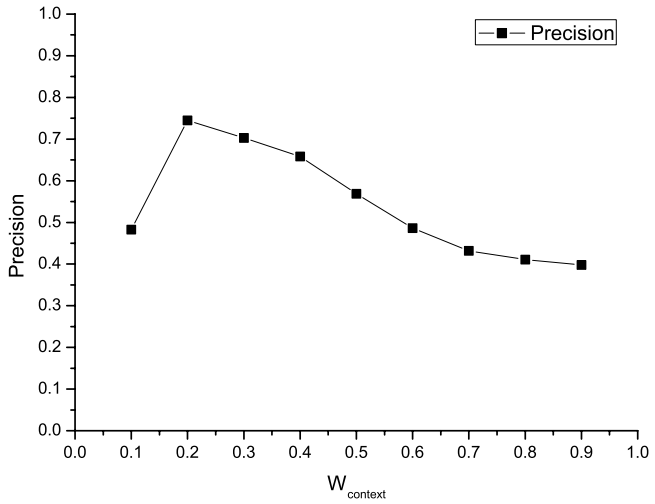


Figure 6. Adjusting the $w_{context}$ to achieve good precision

From Figure 6, we can see that when $w_{context}$ reaches approximately 0.2, the highest level of average precision is achieved. When $w_{context}$ decreases from 0.2 to 0.1 or increases from 0.2 to 0.9, the precision falls.

8.4 Experimental Results

In this section, we compare the recall, precision and harmonic mean of recall and precision (F1) when performing different search approaches.

Comparing lattice context-based results against the CNKI results. Here, we compare the recall and precision scores from the FBS and FBSall approaches to CNKI's general keyword-based search. Papers that are in the CNKI search results may include some papers that are not in our experiment dataset, which should be filtered out before our evaluations because our experiment dataset is only a part of CNKI's papers. While this experiment's FBS results include only papers with scores above a certain threshold, t, the CNKI search results include all papers since CNKI lacks a scoring function. Figure 7 shows the average number of papers from the CNKI, FBS and FBSall approaches.

We selected three sets of cut-off thresholds. The first set ($0 \le t \prec 0.15$ or low threshold values) contains a large number of results. With the first set, we expect high recall. The second set ($0.15 \le t \prec 0.35$ or moderate threshold values) contains a moderate number of search results. We expect a high F1 for the second set. The last set ($t \ge 0.35$) contains high-ranking results, and we expect high precision for this set.

Figure 8 illustrates the average F1 scores for the CNKI, FBS and FBSall approaches.

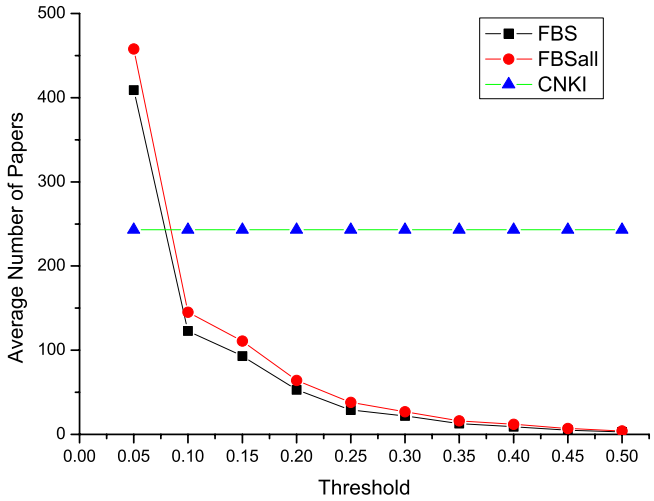


Figure 7. The average number of papers returned from the CNKI, FBS and FBSall approaches

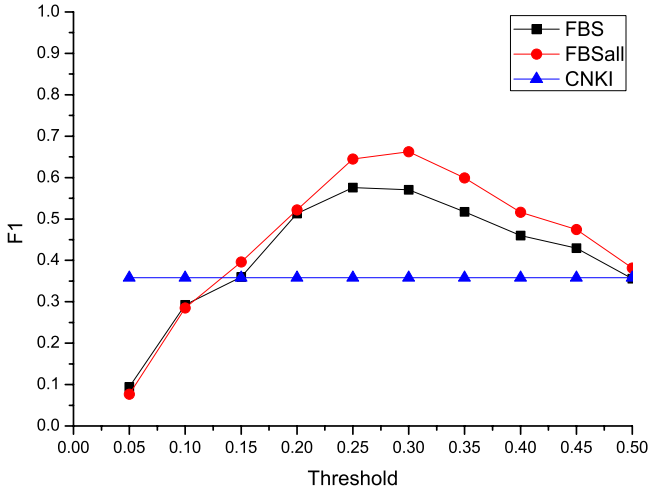


Figure 8. Average F1 scores for the CNKI, FBS, and FBSall approaches

The experiment's results show the following:

(1) At $t \succ 0.20$, CNKI's recall is higher than the lattice context-based (FBS and FBSall) recall. This is due to CNKI searching and returning more papers on average than the lattice context-based search approaches.

(2) The lattice context-based approaches produce approximately 50% higher precision at high thresholds and approximately 25% higher precision at moderate thresholds.

(3) At moderate thresholds, the lattice context-based approaches yield approximately 25% higher F1 scores than CNKI. Moreover, from Figure 7, there is a much smaller number of lattice context-based search results at moderate thresholds (approximately 10 times) than for the CNKI search results.

(4) The FBS approach reduces the query output size by up to 60% compared to the CNKI results.

(5) The FBSall approach has a similar or higher F1 compared to the FBS approach.

9 Conclusion

At the present time, a major problem in searching for scholastic papers within some search engines, such as digital libraries, is the lack of effective paper scoring and ranking systems. For a keyword-based scholastic search, the number of returned papers can be very large. Search results may also contain various topics, not all of which are of interest to the users. In order to solve the problems, we proposed a lattice context-based searching paradigm for paper finding.

In our approach, digital papers are classified as lattice contexts through a pre-processing step that uses lattice context pattern-extraction-based techniques. The lattice context scores are also assigned to papers within each lattice context, where high lattice context scores mean that papers are highly relevant to a given lattice context. After a user has specified a query term, we present to the user a set of lattice contexts that are relevant to the query. A search is performed within the selected lattice contexts, and the search results are ranked and returned to the user based on the strength of match to the query and their lattice context scores.

Our context-based approach improves on the various shortcomings of the present search methods, as follows:

(1) Since papers are classified as relevant contexts through a pre-processing step, the complete lattice paper context information is available before performing a search. After the users define the search queries, the queries are automatically matched against the lattice context information, and only the relevant lattice contexts are presented to the users. With this information, the users can define a scope for their contexts of interest before viewing the search results. Thus, the selected contexts are highly meaningful since they are (a) relevant to the queries and (b) interesting to the users. In contrast to many existing categorization techniques, context-based search results are grouped within only contexts that are of interest as opposed to a large number of all possible topics (contexts). This solves the topic-diffusion problem across search results.

(2) Using recall and precision analysis, we evaluated our two approaches and compared them with a traditional scholastic search engine. The experiment's results demonstrate that our approach produces comparable recall for the search results and higher precision for high-ranking papers. Moreover, the number of search results and contexts returned is much smaller in our approach. For any keyword query executed on a search engine, most users view only the top results. Therefore, high precision for high-ranking results is crucial.

(3) Our context-based approach is general and can be applied to other domains. A lattice context-based search engine allows for any set of keywords, as opposed to some systems that allow only specific types of keywords. Although we initially group the search results within the lattice contexts that they belong to, if the user likes the traditional approach and wants to view only a single ranked list of search results, we provide an approach to merge the relevancy scores from different contexts into one final score, and use the new scores to rank the search results. We present ways to generalize paper classification techniques to non-domain-specific methods that do not utilize any specific terms. Thus, our approach can be applied to any sets of papers.

Although we present a complete lattice context-based search framework, there are still possibilities for further improvement. First, in our experiments, only 40 keywords were selected, which will have resulted in losing some useful information. Second, although a simple, formal context selecting and cutting method is proposed in our work, there are other methods, such as a distributed sub-formal context, that could be explored. Finally, we just compared our work to the CNKI search engine from which our experiments' dataset was built. Our future work will be applied to more public and various other domains' datasets.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61502033, 61472034, 61772071, 61672098, 61272361), Basic Research Fund of Beijing Institute of Technology.

References

[1] Google Scholar, http://scholar.google.com/.

[2] ACM, Association for Computing Machinery Digital Library, http://dl.acm.org/.

[3] Citeseer, http://citeseerx.ist.psu.edu.

[4] R. Wille, Restructuring Lattice Theory: An Approach based on Hierarchies of Concepts, *7th International Conference on Formal Concept Analysis*, Darmstadt, Germany, 2009, pp. 314-339.

[5] A. Formica, Semantic Web Search Based on Rough Sets and Fuzzy Formal Concept Analysis, *Knowledge-Based Systems*, Vol. 26, pp. 40-47, February, 2012.

[6] C. De Maio, G. Fenza, V. Loia, S. Senatore, Hierarchical Web Resources Retrieval by Exploiting Fuzzy Formal Concept Analysis, *Information Processing & Management*, Vol. 48, No. 3, pp. 399-418, May, 2012.

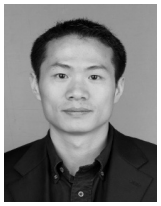
[7] A. El Qadi, D. Aboutajedine, Y. Ennouary, Formal Concept Analysis for Information Retrieval, *International Journal of Computer Science and Information Security*, Vol. 7, No. 2, pp. 119-125, February, 2010.

[8] M. Kim, P. Compton, A Hybrid Browsing Mechanism Using Conceptual Scales, *Pacific Rim Knowledge Acquisition Workshop*, Guilin, China, 2006, pp. 132-143.

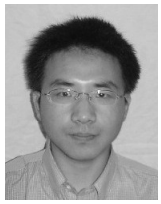
[9] C. Carpineto, G. Romano, Using Concept Lattices for Text Retrieval and Mining, in: B. Ganter, G. Stumme, R. Wille (Eds.), *Formal Concept Analysis, Lecture Notes in Computer Science*, Vol. 3626, Springer, Berlin, 2005, pp. 161-179.

[10] C. Carpineto, G. Romano, A Lattice Conceptual Clustering System and Its Application to Browsing Retrieval, *Machine Learning*, Vol. 24, No. 2, pp. 95-122, August, 1996.
[11] L. Kovics, P. Baranyi, Document Clustering based on Concept Lattice, *IEEE International Conference on Systems, Man and Cybernetics*, Hammamet, Tunisia, 2002, pp.1-6.
[12] Y. Zhang, B. Feng, Y. Xue, A New Search Results Clustering Algorithm based on Formal Concept Analysis, *Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, Shandong, China, 2008, pp. 356-360.
[13] J. M. Cigarran, J. Gonzalo, A. Penas, F. Verdejo, Browsing Search Results via Formal Concept Analysis: Automatic Selection of Attributes, *Second International Conference on Formal Concept Analysis*, Sydney, Australia, 2004, pp. 74-87.
[14] T. Jun, Y.-J. Du, J.-F. Shen, Research in Concept Lattice based Automatic Document Ranking, *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 2005, pp. 5560-5565.
[15] F. Dau, J. Ducrou, P. Eklund, Concept Similarity and Related Categories in Searchsleuth, *16th International Conference on Conceptual Structures*, Toulouse, France, 2008, p.255-268.
[16] B. Ganter, R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, 1999.
[17] R. Godin, R. Missaoui, H. Alaoui, Incremental Concept Formation Algorithms based on Galois (Concept) Lattices, *Computational Intelligence*, Vol. 11, No. 2, pp. 246-267, May, 1995.
[18] C. Shi, Z. Niu, X. Cheng, Lattice-context Based Digital Paper Search, *International Conference on Software Engineering & Knowledge Engineering (SEKE'2010)*, San Francisco Bay, CA, 2010, pp. 315-318.
[19] G. Salton, C. Buckley, Term-weighting Approaches in Automatic Text Retrieval, *Information Processing & Management*, Vol. 24, No. 5, pp. 513-523, October, 1988.
[20] C. Shi, Z. Niu, A Novel Similarity Evaluating Model based on Rfca and Ics, *IEEE International Conference on Digital Information Management*, Thunder Bay, Canada, 2010, pp. 114-119.
[21] CNKI, http://www.cnki.net/.
[22] D. T. Hawkins, R. Wagers, Online Bibliographic Search Strategy Development, *Online*, Vol. 6, No. 3, pp. 12-19, May, 1982.
[23] A. Al-Hamdani, *Querying Web Resources with Metadata in a Database*, Ph.D. Dissertation, Case Western Reserve University, Cleveland, OH, 2004.
[24] H. Small, Co-citation in the Scientific Literature: A New Measure of the Relationship between Two Documents, *Journal of the Association for Information Science and Technology*, Vol. 24, No. 4, pp. 265-269, July/August, 1973.
[25] M. M. Kessler, Bibliographic Coupling between Scientific Papers, *American Documentation*, Vol. 14, No. 1, pp. 10-25, January, 1963.

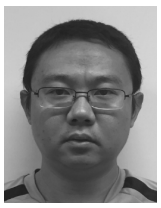
Biographies



Haibin Yu received his master's degree from school of computing, Beijing Institute of Technology, China, in 2006. He is currently an University Staff in Beijing Institute of Technology. His research interests include knowledge management, machine learning, etc.



Chongyang Shi is a lecturer in School of Computer Science, Beijing Institute of Technology. He obtained his Ph.D. degree from Beijing Institute of Technology in 2010, all in Computer Science. His research areas focus on Information Retrieval, Knowledge Engineering, Personalized Service, Sentiment Analysis etc.



Bai Yu received his B.E. and M.E. degree in computer science from Tianjin University, China. Currently, he is a senior research assistant at the Centre for Quantum Computation and Intelligent Systems (QCIS), University of Technology Sydney, Australia. His research focuses on data mining and machine learning.

Chunxia Zhang received her Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2005. She is currently an associate professor in School of Software, Beijing Institute of Technology. Her research interests include information extraction, knowledge management, machine learning, etc.

Ryan Hearne, a Computer Science Masters student at Beijing Institute of Technology. My area of research is data mining and sentiment analysis using Python. He graduated with a First Class Honours degree in Software Engineering from Waterford Institute of Technology in 2015. His research focuses on mobile and web development.