Information Processing and Management 46 (2010) 295–307

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Automatic index construction for multimedia digital libraries

San-Yih Hwang [a,*], Wan-Shiou Yang [b], Kang-Di Ting [a]

[a] Department of Information Management, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan
[b] Department of Information Management, National Changhua University of Education, Changhua 50007, Taiwan

ARTICLE INFO

Article history:
Received 27 August 2008
Received in revised form 1 October 2009
Accepted 6 October 2009

Keywords:
Cluster indexing
Multimedia database
Web usage mining
Information retrieval
Digital library

ABSTRACT

Indexing remains one of the most popular tools provided by digital libraries to help users identify and understand the characteristics of the information they need. Despite extensive studies of the problem of automatic index construction for text-based digital libraries, the construction of multimedia digital libraries continues to represent a challenge, because multimedia objects usually lack sufficient text information to ensure reliable index learning. This research attempts to tackle the problem of automatic index construction for multimedia objects by employing Web usage logs and limited keywords pertaining to multimedia objects. The tests of two proposed algorithms use two different data sets with different amounts of textual information. Web usage logs offer precious information for building indexes of multimedia digital libraries with limited textual information. The proposed methods generally yield better indexes, especially for the artwork data set.

1. Introduction

The rapid advances of information technologies have allowed for the inclusion of vast amounts of electronic information in digital libraries. This electronic information initially was primarily text-based, but it has expanded to include graphics, animation, audio, video, and interactive media (Tjondronegoro & Spink, 2008). Thus, the ability to help users easily, efficiently, and conveniently retrieve multimedia information from the vast array available presents both an opportunity and a challenge for modern digital libraries.

In traditional text-based digital libraries, indexing provides the main tool to help users seek information and understand the topics contained within documents of interest to them (Berry & Castellanos, 2007). Many researchers address the challenge of indexing text-based information by leveraging content features derived from titles, keywords, abstracts, or full texts and thereby determining similarities among objects (Berry & Castellanos, 2007; Boley et al., 1999; Zhao & Karypis, 2002). A clustering technique then develops a set of clusters, each of which receives a label, assigned manually or automatically (Yang & Pederson, 1997) that distinguishes the documents in that cluster from those in other clusters. The clusters then form an index for organizing text-based information.

Indexing multimedia information, however, is more challenging, because these data comprise opaque collections of bytes with limited textual information, such as short titles, the date of creation, or names of the artists (Mehtre, Kankanhalli, & Lee, 1997). Despite the existence of some techniques for automatic keyword extraction of multimedia objects in specific domains, the number of derived keywords and their accuracy remain limited (Tsai, McGarry, & Tait, 2006). Furthermore, with limited textual information for multimedia objects, traditional text-based clustering approach may not work well. Therefore, a pressing need emerges, namely, to integrate other sources of data to cluster objects in a multimedia digital library.

* Corresponding author. Tel.: +886 7 525 2000x4723; fax: +886 7 525 4799.
E-mail address: syhwang@mail.nsysu.edu.tw (S.-Y. Hwang).

0306-4573/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2009.10.006

Table 1
Sample Web usage log of NSYSU ETD.

Source IP	Access time	Method/URL/protocol	Referrer
218.165.248.55	01/Apr/2004:00:00:02 + 0800	GET/ETD-db/ETD-search-c/view_etd?URN = etd-0717101-163917 HTTP/1.1	http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search-c/search
218.160.2.43	01/Apr/2004:00:00:11 + 0800	GET/ETD-db/ETD-search-c/view_etd?URN = etd-0130101-140550 HTTP/1.1	http://www.google.com.tw/search?as_q=...
218.160.2.43	01/Apr/2004:00:00:11 + 0800	GET /ETD-db/images/logofull.gif HTTP/1.1	http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search-c/view_etd?URN = etd-0130101-140550
218.165.101.164	01/Apr/2004:00:00:12 + 0800	GET /ETD-db/ETD-search-c/getfile?URN = etd-0724101-130330&filename = etd-0724101-130330.pdf HTTP/1.1	http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search-c/view_etd?URN = etd-0724101-13033
220.228.131.188	01/Apr/2004:00:00:38 + 0800	GET /ETD-db/ETD-search-c/view_etd?URN = etd-0815102-010654 HTTP/1.1	http://www.google.com.tw/search?...
220.228.131.188	01/Apr/2004:00:00:41 + 0800	GET/ETD-db/images/logofull.gif HTTP/1.1	http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search-c/view_etd?URN = etd-0815102-010654
140.117.193.121	01/Apr/2004:00:00:46 + 0800	GET/ETD-db/ETD-search-c/search HTTP/1.1	http://www.lib.nsysu.edu.tw/eThesys/
140.117.193.121	01/Apr/2004:00:00:46 + 0800	GET/ETD-db/images/logofull.gif HTTP/1.1	http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search-c/search
140.117.193.121	01/Apr/2004:00:00:58 + 0800	GET/ETD-db/ETD-search-c/search?... HTTP/1.1	http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search-c/search

With the advent of the World Wide Web, an overwhelming number of digital libraries now provide interfaces that allow for ubiquitous information access. The usage data associated with Web-based digital libraries automatically get recorded in Web usage logs by Web servers. Therefore, each user click within a Web-based digital library results in one or more records in the Web usage log, such that each record represents the source IP, access time, access method/URL/protocol, referred URL, status, bytes transferred, browser type, and so forth. Table 1 displays a sample Web usage log for the electronic thesis and dissertation (ETD) system at National Sun Yat-Sen University. The first record in Table 1 shows that an entry with a universal resource number etd-0717101-163917 was accessed on 01/Apr/2004:00:00:02 by a user with the IP address 218.165.248.55. The second and third records in Table 1 indicate a user who has chosen to view an entry identified by etd-0130101-140550. Objects of the same category logically should have a higher chance of being accessed together compared with objects in different categories. Therefore, we propose to tackle the problem of indexing multimedia information by employing Web usage logs, in combination with limited keywords attached to multimedia information.

This article reports our endeavor to integrate textual data and usage data pertaining to multimedia objects and thus build the index. We develop two methods to construct an index for multimedia objects that employs both the (possibly limited) textual data associated with the objects and their usage data over a specified period of time, as recorded by Web servers. One method, called MCAT, applies both clustering and classification techniques, and the other, MCLU, uses only clustering techniques. We apply both methods, as well as methods that use only textual data, to two data sets derived from the World Art digital library from Airiti, Inc., in Taiwan and the ETD system at National Sun Yat-Sen University. The World Art digital library involves only a limited amount of textual information pertaining to images of artwork. The evaluation results using this data set show that an index constructed by considering both usage and content data better matches the predefined index than does an index that uses only one source of data. In addition, the resulting index effectively reduces users' efforts to find the information they require. The ETD system contains a profound amount of textual information, in addition to usage data, so we use it to investigate how our proposed methods perform even for a digital library with rich textual data. Compared with traditional text-based approaches, the indexes created by our proposed methods are only slightly inferior in terms of matching the predefined index. Nevertheless, our proposed indexes retain the advantages of enabling users to identify the information they need quickly. We thus conclude that the proposed methods offer promising improvements for building indexes for multimedia digital libraries.

The remainder of this article is organized as follows: In Section 2, we review related research efforts. In Section 3, we describe our methods for indexing multimedia information, using textual content information and Web usage logs. We report the results of our experiments in Section 4 and evaluate the various methods by applying real-world data collected from the World Art digital library and an ETD system. Finally, in Section 5, we summarize and point to some further research directions.

2. Literature review

Digital libraries attract tremendous interest, including several research projects that attempt to address the vast challenges in this field, such as the Alexandria Digital Library (ADL) project at the University of California at Santa Barbara (Manjunath & Ma, 1996), the DLI project at the University of Illinois (Chen et al., 1996), the Informedia project at Carnegie Mellon University (Wactlar, Kanade, Smith, & Stevens, 1996), the Variations2 project at Indiana University (Byrd & Isaacson, 2003), the Michigan Digitization Project (also known as MBooks) at the University of Michigan (Vaidhyanathan, 2006), the SAPIR project at IBM Haifa Research Laboratory (http://www.sapir.eu/), the DELOS project by DELOS Association for Digital Libraries in Europe (http://www.delos.info/), and the TelPlus project funded by the European Commission (http://www.theeuropeanlibrary.org/telplus/). Although some of these digital library projects address the problem of content-based search (e.g., SAPIR), most of them assign keywords to multimedia objects, whether manually or automatically, to enable high-level information inquiries. Automatic keyword assignments to multimedia objects usually begin by extracting low-level features, followed by the assignment of semantic concepts to the objects on the basis of their low-level features. Typically, color, shape, and texture features can be extracted to represent image or video data (Albuz, Kocalar, & Khokhar, 2001; Smeulders, Worring, Santini, Gupta, & Jain, 2000; Wactlar et al., 1996); pitch, loudness, duration, rhythm, and timbre provide the audio or music data (Mierswa & Morik, 2005; Morchen, Ultsch, Thies, & Lohken, 2006; Wold, Blum, Keislar, & Wheaton, 1996). Nevertheless, techniques for the automatic derivation of textual information from low-level features of other multimedia objects, such as images, video, or audio, usually have limited applicability and at best can generate small amounts of textual data (except for speech audio). For example, automatic image indexing typically uses classification techniques based on a few class labels, one for each semantic concept (Tsai et al., 2006). These techniques attempt to construct a model that associates semantic concepts with some extracted, low-level content features, such as color, contrast, or texture.

Generally, the problem with automatically indexing text-based digital objects, called *documents* hereafter, involves finding appropriate groups of documents that possess strong internal textual similarities. The resultant index can be used to understand the topics addressed by the documents, as well as facilitate information search (Berry & Castellanos, 2007). The formalization of the problem thus involves three separate issues. First, the most common approach to document representation depicts each document as a multidimensional vector, in which the element of each dimension corresponds to the weight of a distinct term extracted from the documents, such as in TF $\times$ IDF (Salton & McGill, 1983). Second, measuring the similarity/distance between documents (Korfhag, 1997; Salton & McGill, 1983) often relies on the inner product of two

vectors that represent the cosine of the angle between the vectors. Third, clustering algorithms are used to partition the documents automatically into a set of clusters (Han & Kamber, 2006).

The various techniques for deriving textual data also might apply to multimedia objects to enrich their textual data. Multimedia search engines, for example, index images, video, or audio objects linked by HTML pages or embedded within them, which serves the needs of search users interested in multimedia objects (Jansen, Goodrum, & Spink, 2000; Swain, 1999). These multimedia search engines use relevant text retrieved from the HTML pages that reference multimedia objects, header information (e.g., title, author, copyright information), and content information (e.g., color and shape for image/video; rhythm and timbre for music/audio), to index the multimedia objects. Li, Myaeng, and Kim (2007), for example, analyze users' opinions of music items, according to the groupings of music items, then integrate the content features of the music to make recommendations.

Another important data source for forming object clusters may pertain to Web usage logs. Most previous works that employ this source focus on Web pages; for example, Cooley, Mobasher, and Srivastava (1999) compute overlapping clusters of URL references on the basis of their co-occurrence patterns across user sessions. However, traditional clustering techniques, such as distance-based methods, cannot handle such clustering because of the inherently high dimensions, such that each Web page corresponds to one dimension. The association rule hypergraph partitioning (ARHP) technique (Han, Karypis, Kumar, & Mobasher, 1998) provides a good alternative, in that it identifies a set of frequent itemsets from Web usage logs, each of which contains a set of Web pages often accessed in the same session. These itemsets subsequently represent hyperedges of a hypergraph. Finally, the hypergraph can be partitioned into a collection of clusters (Karypis, Aggarwal, Kumar, & Shekhar, 1999).

3. Proposed methods

Most previous work employs textual information to construct document indexes, though more recent work also facilitates clustering with Web usage logs. We observe though that multimedia digital libraries often lack sufficient textural information and propose constructing an index of multimedia objects by employing both (textual) content and usage data. We define the content similarity between two multimedia objects according to their textual data. Specifically, every multimedia object can be converted into a vector, in which each dimension corresponds to a keyword identified from the set of objects. Traditional TF × IDF measures (Salton & McGill, 1983) determine the weight of each keyword for each object, where $w_{i,j}$ gets assigned to the jth keyword of the ith object:

$$w_{i,j} = f_{i,j} \times idf_j.$$

In turn, $f_{i,j} = \frac{freq_{i,j}}{\max freq_{i,j}}$ and $idf_j = \log \frac{N}{n_j}$, where $freq_{i,j}$ is the raw frequency of the jth keyword in the ith object, N is the total number of objects in the digital library, and n_j is the number of objects in which the jth keyword appears. Although this formula works well for ideal and small data sets, Salton and Buckley (1988) suggest a slight modification to reduce the effect of term frequency on the weight for more general cases:

$$w_{i,j} = (0.5 + 0.5 f_{i,j}) \times idf_j.$$

After this conversion, each multimedia object is represented by a vector. The content similarity $csim(a_1, a_2)$ of two objects a_1 and a_2 then is defined as the cosine of the angle between the vectors of a_1 and a_2.

As mentioned in Section 1, a Web-based digital library contains a Web usage log, and each record in the Web usage log represents a specific access to a file or object in the digital library. The raw Web usage log must be processed properly and converted into a set of user sessions for subsequent use by the proposed index construction approaches, which we describe subsequently. A user session includes a set of multimedia objects, consecutively accessed by a user to perform a task at hand. We follow the approach proposed by Srivastava, Cooley, Deshpande, and Tang (2000) to identify user sessions. To initiate the procedure, we remove unwanted Web usage log records (i.e., those accessing irrelevant pages or made by Internet robots), as well as any two records with different IP addresses, browser software, or operating systems, which we assume belong to two different user sessions. In addition, when the time interval between a usage log record and its preceding record (for the same user) exceeds 30 min, we assumed that a new user session had started.

In the past, several approaches have been proposed for using either content or usage data, but not both, to construct an index. We present two index construction methods that use both content and usage data.

3.1. Multimedia categorization-based approach (MCAT)

The *multimedia categorization-based approach*, or MCAT, combines clustering techniques based on the Web usage log with classification techniques based on textual data. It employs the Web usage log to cluster a (relatively) small number of multimedia objects, then classifies the remaining objects into existing clusters according to their textual data. Specifically, the two-step ARHP approach (Mobasher, Dai, Luo, & Nakagawa, 2002) serves to partition multimedia objects into a set of overlapping clusters based on the Web usage log by first applying the frequent itemset mining algorithm to the user sessions (identified from Web usage log) to find a set of frequent itemsets, each of which contains a set of multimedia objects often accessed during the same user sessions. The frequent itemsets, denoted $IS = \{I_1, I_2, \ldots, I_k\}$, then form a hypergraph $H = \langle V, E \rangle$,

Input: a desired number of clusters k, a multimedia object database D, a usage log $\Im$

Output: a partition $\{C_0, C_1, C_2, \ldots, C_k\}$ of D

1. $\Sigma \leftarrow$ Identify a set of user sessions from $\Im$;
2. $\wp \leftarrow$Discover a set of frequent itemsets from Σ;
3. $\{C_1, C_2, \ldots, C_k\} \leftarrow$Partition the hypergraph formed by $\wp$;
4. **for** each cluster C_i and each itemset I in $\wp$ **do**
5. **if** $|I \cap C_i| > 0.5 \cdot |I|$ **then** $C_i = C_i \cup I$;
6. **for** each cluster C_i **do** build a binary classifier CL_i using textual data;
7. $U \leftarrow D - (C_1 \cup C_2 \cup \ldots \cup C_k)$; $C_0 = \varnothing$;
8. **for** each object o in U **do**
9. *Clustered* = False;
10. **for** each cluster C_i **do**
11. **if** $CL_i(o)$ = TRUE **then**
12. $C_i = C_i \cup \{o\}$;
13. *Clustered* = True;
14. **if not** *Clustered* **then** $C_0 \leftarrow C_0 \cup \{o\}$;

Fig. 1. Pseudo-code of multimedia categorization-based approach (MCAT).

where $V = I_1 \cup I_1 \cup \cdots \cup I_k$ and $E = IS$. (A hypergraph is an extension of a graph in which each hyperedge can connect more than two vertices.) Following the approach proposed by Mobasher et al. (2002), we define the weight of a hyperedge $\{o_1, o_2, \ldots, o_k\}$ as $\frac{(o_1, o_2, \ldots, o_k)}{(o_1) \cdot (o_2) \cdots (o_k)}$, where (o_i) and $(o_1, o_2, \ldots, o_k)$ are the supports of $\{o_i\}$ and $\{o_1, o_2, \ldots, o_k\}$, respectively. A hypergraph partitioning algorithm then partitions the set of objects into disjointed clusters to minimize the total weight of hyperedges across clusters. Because an object may be classified into more than one category in practice, some objects get added back into a cluster, resulting in non-disjointed clusters. Specifically, for a given hyperedge I_i and a given cluster C_j, if the proportion of objects in I_i that appear in C_j is greater than a certain threshold (set to 50% for our experiments), all objects in I_i get added to C_j. In the resultant clustering, objects in the same cluster are more "similar," in the sense that users are more likely to access them in the same sessions.

Although clustering based on the Web usage log achieves some success in various application domains (e.g., recommendations), it suffers from a low coverage problem (Hwang & Chuang, 2004) because of its highly skewed distribution of access, such that the ratio of items that often appear with at least one other item is relatively small. Hwang, Hsiung, and Yang (2003) report that only about one-tenth of documents in a typical ETD database are accessed together with some other documents a sufficient number of times. To alleviate this problem, MCAT adds the multimedia objects that have not been clustered by the previous step to existing clusters on the basis of their (textual) content information. The assignment of these unclustered objects to existing clusters represents a classification problem, which perceives objects in the clusters as training data and labels as the identifiers of the clusters. Specifically, let the set of clusters obtained using ARHP be $\{C_1, C_2, \ldots, C_k\}$, where $T = C_1 \cup C_2 \cup \cdots \cup C_k$. A binary classifier CL_i can be built for each cluster C_i on the basis of a training data set in which the positive examples are C_i and the negative examples equal $T - C_i$. In our experiments, we use support vector machines (SVM; Joachims, 1998) to build a binary classifier for each cluster, according to the textual data. Each unclustered multimedia object o then can be added to C_i if it is predicted by CL_i to be positive. Finally, objects that are not assigned to any cluster are placed in a new cluster C_0. The resultant clustering $\{C_0, C_1, C_2, \ldots, C_k\}$ involves all objects, as depicted by the pseudo-code of MCAT in Fig. 1.

3.2. Multimedia clustering-based approach (MCLU)

The second method, the *multimedia clustering-based approach* (MCLU), directly clusters multimedia objects on the basis of both their textual and usage data. The basic idea behind MCLU is to construct a hypergraph that embodies both content and

Input: a desired number of clusters k, a multimedia object database D, a usage log $\Im$

Output: a partition $\{C_0, C_1, C_2, \ldots, C_k\}$ of D

1. $\Sigma \leftarrow$ Identify a set of user sessions from $\Im$;
2. $\wp \leftarrow$Discover a set of frequent itemsets (or patterns) from Σ;
3. $\Re \leftarrow$Discover a set of clique patterns using textual data in D;
4. $H \leftarrow$Generate a hypergraph by using patterns in $\wp$ and $\Re$ as hyperedges;
5. $\{C_1, C_2, \ldots, C_k\} \leftarrow$Partition the hypergraph H;
6. **for** each cluster C_i and each pattern I in $\wp$ or $\Re$ **do**
7. **if** $|I \cap C_i| > 0.5 \cdot |I|$ **then** $C_i = C_i \cup I$;
8. $C_0 \leftarrow D - (C_1 \cup C_2 \cup \ldots \cup C_k)$;

Fig. 2. Pseudo-code of multimedia clustering-based approach (MCLU).

usage similarities of multimedia objects, then partition that hypergraph into non-exclusive clusters of objects. It forms two kinds of hyperedges using textual content and the usage log of multimedia objects. The hyperedges derived from textual data represent cliques of multimedia objects that exhibit pairwise content similarities. Formally, a clique is an undirected complete graph $G = (V, E)$, in which V represents the set of multimedia objects, and $(u, v) \in E$ iff its content similarity $csim(u, v) \geqslant \tau$, where τ is a predefined threshold. The weight of a clique Q equals the average content similarity between any two objects in Q. The hyperedges derived from the Web usage log are frequent itemsets and take the weights assigned using the same approach as described previously for MCAT.

However, locating a clique of maximum size entails an NP-complete problem, which means it is not practical to enumerate all maximal cliques, especially when τ is small. Nonetheless, as Mobasher, Cooley, and Srivastava (1999) show, enumerating all maximal cliques of a size no larger than a constant K takes only polynomial time. Let V be the set of multimedia objects. For a given object $v \in V$, let $Ksim(v, k)$ denote the kth largest similarity values between v and any other object in V. If $\tau = \underset{v}{Max}(Ksim(v, K-1))$, the maximum clique of the graph $G = (V, E)$ must be no greater than K. Therefore, we set $\tau = \underset{v}{Max}(\hat{Ksim}(v, K-1))$ for a given constant K and use the approach described by Mobasher et al. (1999) to enumerate all maximal cliques. These cliques become the content-based hyperedges.

To prevent bias toward either usage or content data, we normalize the weights of the hyperedges, such that the maximum weight of the content-based hyperedges equals the maximum weight of the hyperedges generated by the Web usage log. The hypergraph finally can be partitioned into a set of non-disjointed clusters using the approach described in Section 3.1; we show the pseudo-code of MCLU in Fig. 2.

3.3. Index updating

The object collection of a multimedia digital library may change over time, and it is imperative to include new objects in the index to facilitate the search for these objects. Even if the set of multimedia objects stays the same, the Web usage logs, as required by our proposed methods, constantly increase at a very high speed. When the object access distribution changes with time, the index that was constructed using older usage data becomes obsolete. Therefore, there is a need to monitor the stream of Web usage logs continuously and rebuild the index whenever a significant change occurs in the multimedia object access patterns.

Both MCAT and MCLU identify frequent itemsets from user sessions to discover sets of multimedia objects that often are accessed together. However, new user sessions, as derived from new Web usage log records, arrive at a very fast rate, similar to a data stream. In the past decade, extensive studies have attempted to address the problem of mining frequent itemsets over a data stream, in which new transactions arrive at a high speed, and many of them propose algorithms based on various requirements. For example, some researchers focus on the entire history of object access; some consider the access history of more recent accesses, often called the sliding window approach; and still others downgrade the importance of older access history and prioritize only more recent access events. In addition, some algorithms attempt to find exact solutions for the frequent itemsets, at the expense of substantial time and space requirements, though most proposed algorithms intend to find only approximate, frequent itemsets with bounded error. These algorithms differ with regard to the data structures they use to maintain a set of itemsets that are currently frequent or have the potential to become frequent in the near future. A prefix tree or its variations are common solutions; when a batch of user sessions arrives or expires, some itemsets may be

Fig. 3. Homepage of Airiti World Art Digital Library.

inserted into or removed from the prefix tree (or other data structure). Chen, Ke, and Ng (2008) offer an excellent survey of this approach.

The stream-based frequent itemset mining algorithms also can maintain the current frequent itemsets of multimedia objects. If the set of frequent itemsets grows substantially different from that used to construct the current index, the index can be reconstructed using either MCAT or MCLU. Although algorithms exist for incremental clustering and incremental classification for streaming data (Gaber, Zavlavsky, & Krisnaswamy, 2005), we tend not to use them because the object access distribution will not be subject to constant change. Moreover, these incremental algorithms do not come without price—they often compromise clustering/classification accuracy.

4. Evaluation

4.1. Data sets

To evaluate our proposed methods, we collected data from two test beds: the World Art Digital Library from Airiti, Inc. (http://www.airiti.com/Arts), whose home page (in Chinese) is in Fig. 3, and the ETD System at National Sun Yat-Sen University (NSYSU) (http://www.lib.nsysu.edu.tw/eThesys/), whose English home page appears in Fig. 4. The World Art Digital Library contains a limited amount of textual information, whereas the NSYSU ETD System provides abundant textual content. We also obtained the Web usage logs for both systems; we refer to these data sets as the artwork data set and the ETD data set, respectively. We purposely choose two data sets with contrary textual data characteristics so that we can examine the effect of textual data on the performance of the constructed indexes.

The Airiti World Art Digital library contains digital images of more than 60,000 artworks, each of which is represented by a digitalized image and some text-based content information, such as the artist, title, size, and material. By August 2005, these artworks had been browsed more than 10,000,000 times, which means many Web usage log records were available.

The World Art Digital Library classifies each artwork using a scheme that, at the first level, includes five broad categories: Western, Chinese, Taiwanese, modern, and child art. We chose 2991 paintings classified as Western art for our experiment, which have been further classified into the following 16 categories:

1. Art of the Middle Ages
2. Renaissance

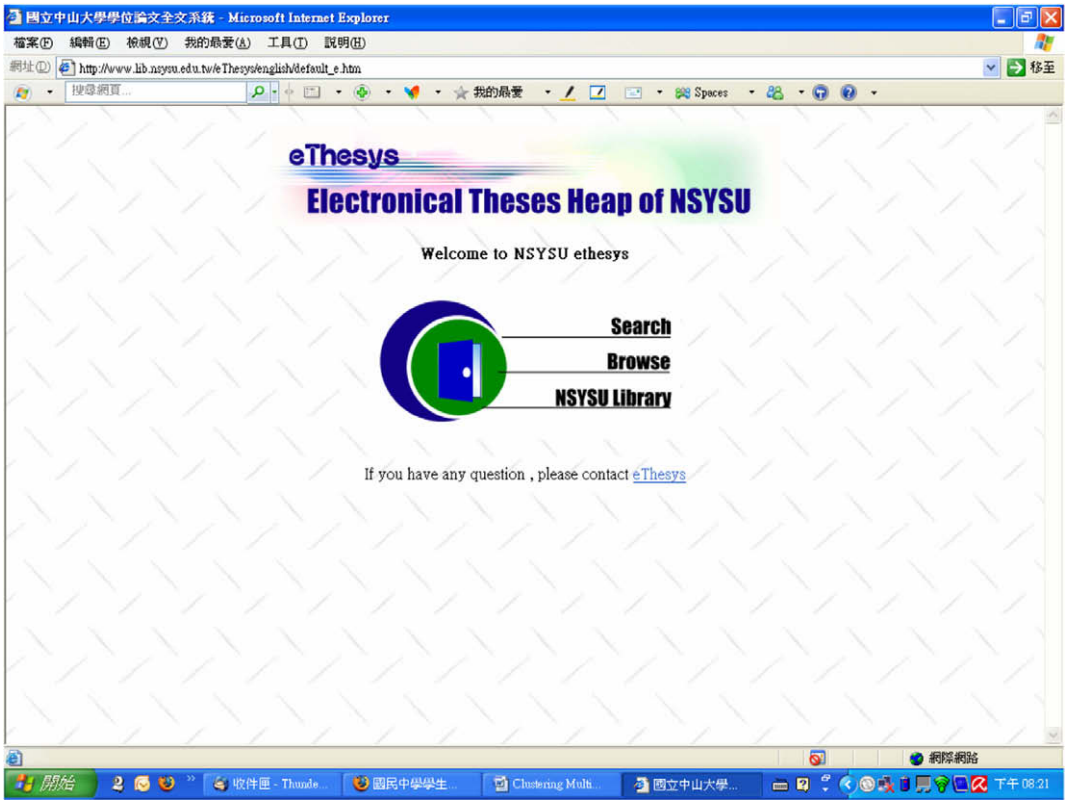

Fig. 4. Homepage of NSYSU ETD System (English version).

3. Baroque
4. Rococo
5. Neoclassicism
6. Romanticism
7. Realism
8. Impressionism
9. Post-Impressionism
10. Expressionism
11. Cubism
12. Surrealism
13. Abstract Expressionism
14. Symbolism
15. Pre-Raphaelite
16. Futurism

The Web usage log records dated between January 1, 2005, and December 31, 2005, provide input for our experiment. From the total of 701,273 log records, using the approach we described in Section 3, we extract 60,012 user sessions.

Since its initiation in May 2000, the NSYSU ETD System has collected more than 10,000 electronic theses. Each ETD thesis contains information in various metadata categories, including the author, title, abstract, keywords, table of contents, and bibliography, in addition to the full texts. To focus our study, we chose 408 computer science-related ETDs submitted before August 2003 from the departments of Computer System Engineering (CSE) and Management Information System (MIS). Two experts with doctoral degrees in computer science (CS) and information systems (IS) manually classified these ETDs into categories from the first-level ACM classification scheme (http://www.acm.org/class/1998/overview.html), as follows:

1. General literature
2. Hardware
3. Computer system organization
4. Software
5. Data

6. Theory of computation
7. Mathematics of computing
8. Information systems
9. Computing methodologies
10. Computer applications
11. Computing milieu

The experts could assign more than one category to each ETD. Three categories—general literature, mathematics of computing, and computing milieu—do not contain any ETDs, according to both experts, so our subsequent experiments involve only eight categories. To ensure accurate evaluations, we used 287 ETDs that both experts classified into the same categories.

The Web usage log records of the NSYSU ETD system, dated between October 31, 2003 and April 30, 2004, provide further information for our experiment. We followed the same approach as described previously and thereby identified 3183 user sessions involving ETDs in the data set.

4.2. Performance benchmarks and metric

To evaluate the performance of our proposed methods, we used two content-based clustering methods, namely, K-means (Han & Kamber, 2006) and ACHP (Hwang & Chuang, 2004), as our benchmarks. Pure usage-based methods, such as ARHP, do not appear in our experiments, because they can only cluster a few objects (e.g., one-tenth of all objects, as reported by Hwang et al. (2003)) and thus are of little use in practice. K-means is a traditional clustering method that creates a one-level partitioning of objects on the basis of their content similarity. The ACHP technique is similar to MCLU, as we described in Section 3.2, except that it uses only content-based hyperedges for the hypergraph partitioning. For our experiments, we convert each object in the ETD and artwork data sets into a content vector, following the approach described in Section 3. We then treat the set of vectors as the input for the K-means and ACHP methods.

Our proposed approaches for index construction involve online frequent itemset mining and offline hypergraph partitioning or classification. The efficiency of online frequent itemset mining is beyond the scope of this research, but extensive recent work addresses this topic (Chen et al., 2008). The hypergraph partitioning or classification instead takes place offline, so computation overhead is not a major concern. We focus on measuring the quality of the generated clusters, with *usage entropy* as our first performance metric; it measures the entropy of each user session generated from the Web usage log with respect to a set of clusters. For example, consider a clustering $C = \{C_1, C_2, \ldots, C_k\}$. Let a user session s of size n contain n_i objects of cluster C_i, $1 \leqslant i \leqslant k$. The entropy of s, denoted $E(s)$, then is given by

$$E(s) = -\sum_{1 \leqslant i \leqslant k} \frac{n_i}{n} \cdot \log \frac{n_i}{n}.$$

The usage entropy of C equals the average entropy of all user sessions. A clustering with a smaller usage entropy allows users to identify most of the needed objects from a small number of clusters, which offers greater convenience in users' information searches.

The second performance metric we adopt to measure the quality of the generated clusters is *content entropy*, which measures the homogeneity of a cluster with respect to the predefined categories to which objects in the cluster belong. In view of the overlapping nature between categories, we can divide the categories further into a set of disjointed segments. The entropy of a cluster then can be computed according to the set of disjointed segments. For example, suppose objects in a cluster involve three overlapping categories, a, b, and c. There are at most seven disjointed segments, $S = \{a, b, c, ab, ac, bc, abc\}$, where ab, for example, represents the set of objects categorized as both a and b. Without loss of generality, suppose there are m disjointed segments for a cluster C_j of size n, and n_i objects are in segment s_i, $1 \leqslant i \leqslant m$. The entropy of C_j, denoted $E(C_j)$, then can be defined as follows:

$$E(C_j) = -\sum_{1 \leqslant i \leqslant m} \frac{n_i}{n} \cdot \log \frac{n_i}{n}.$$

The content entropy of a clustering C equals the average of the entropies of clusters in C. A clustering with smaller content entropy is closer to the predefined categories, resulting in clusters that are more coherent in their subjects and therefore more accurate.

4.3. Evaluation results

This section reports our experimental results using both the artwork and ETD data sets. The two proposed methods, MCAT and MCLU, and two content-based clustering methods, K-means and ACHP, are evaluated according to their usage entropy and content entropy.

4.3.1. Artwork data set

The first experiment intends to measure the usage entropy of the artwork indexes constructed using the four methods. We tried various minimal support values for the four methods, ranging from 0.5% to 2% at 0.5% increments. In addition, for

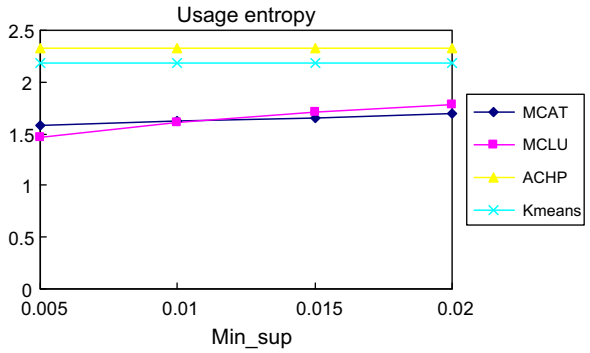

Fig. 5. Usage entropy of clustering resulting from various methods, artwork data set.

MCLU and ACHP, we must specify a threshold τ for content similarities. We set $K = 17$, which results in $\tau = 0.5004$ ($\tau = \underset{v}{Max}(Ksim(v, K-1))$). The same settings apply in the subsequent experiments as well, and we provide the results in Fig. 5.

As we show in Fig. 5, the usage entropies of the two content-based clustering methods, ACHP and K-means, do not change in response to Min_{Sup}. Their results appear for comparison purposes. As expected, our two proposed clustering methods incur lower (and thus better) usage entropy than their content-based counterparts, because both MCLU and MCAT employ the Web usage logs for their clustering. As Min_{sup} increases, MCAT tends to outperform MCLU, because the high Min_{Sup} generates smaller yet more usage-coherent itemsets for MCAT. These coherent itemsets provide the hyperedges of the hypergraph and are more likely to be preserved in the resultant clusters after partitioning the hypergraph. In contrast, the hyperedges generated by MCLU consider both content and usage. When Min_{Sup} is larger, there are fewer usage-based hyperedges compared with the content-based hyperedges, and the clusters contain fewer usage-based hyperedges as a result. Of the two content-based methods, K-means achieves better usage entropy than ACHP, because the artwork data set contains little parsable content information, which results in fewer dimensions, which then favors K-means.

In the second experiment, we measure the content entropy of the artwork indexes for all four methods. The results in Fig. 6 reveal that the content entropies of MCAT and MCLU increase with greater Min_{Sup}, because high Min_{Sup} tends to generate fewer frequent itemsets, which gives higher priority to the content features of the two methods. However, because the artwork data set contains very little content (textual) information, focusing more on the content features does not help reduce content entropy but rather achieves a content entropy similar to that of ACHP. Our two proposed methods incur lower (and thus better) content entropy than either of their content-based counterparts though. Therefore, when textual content information is limited, it is not wise to cluster multimedia objects using textual content only. Incorporating usage data in this case significantly improves the content entropy score.

The number of unclassified artworks for the different clustering schemes appears in Fig. 7. K-means has no unclassified artworks, because it can cluster every object, whereas ACHP offers the largest number of unclassified artworks. In our proposed methods, the number of unclassified artworks increases with an increase of MIN_{sup}, because higher MIN_{sup} induces fewer frequent itemsets, which in turn result in fewer artworks in each cluster. In particular, MCAT relies purely on usage data for clustering. With fewer positive examples in each cluster, the classifier employed by MCAT tends to assign more negative examples for the unclustered artworks, thereby causing even more unclassified objects in comparison with MCLU.

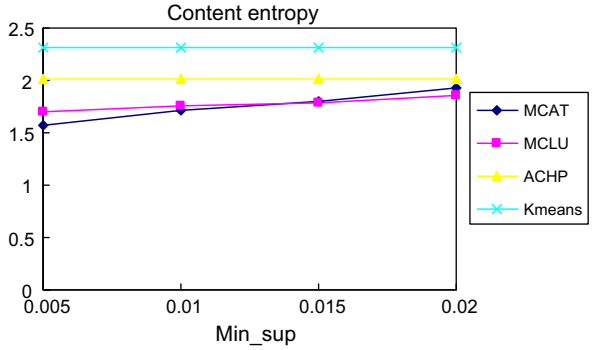

Fig. 6. Content entropy of clustering resulting from various methods, artwork data set.

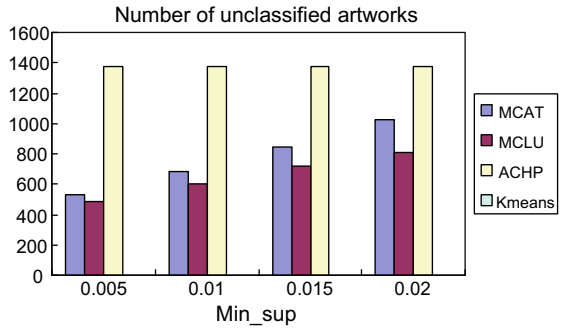

Fig. 7. Number of unclassified artworks remaining with various methods, artwork data set.

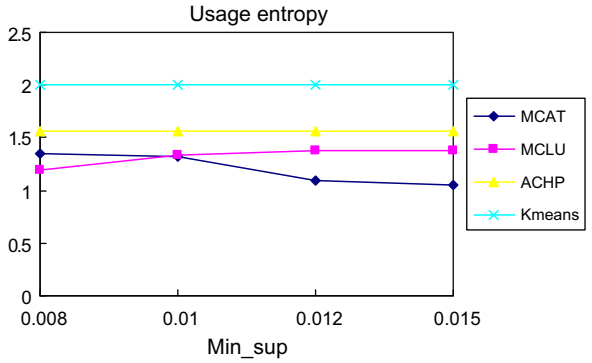

Fig. 8. Usage entropy of clustering resulting from various methods, ETD data set.

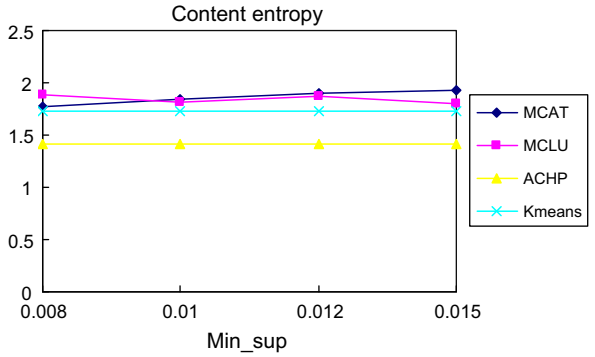

Fig. 9. Content entropy of clustering resulting from various methods, ETD data set.

4.3.2. ETD data set

The next experiment measures the usage entropy of the ETD indexes constructed using the four different methods. As we show in Fig. 8 and as expected, our two proposed clustering methods incur lower (and thus better) usage entropy than their content-based counterparts, because both MCLU and MCAT employ the Web usage logs for their clustering. Of the two content-based methods, K-means suffers the worst usage entropy, which coincides with observations in previous research (Han, Karypis, Kumar, & Mobasher, 1997) that K-means achieves poor performance when the number of dimensions increases. The relative performance difference between MCAT and MCLU follows the same trend as reported in Fig. 5 for experiments using the artwork data set.

In the last experiment, we measure the content entropy of ETD clusters generated by the four methods using the clusters identified by the experts as a benchmark. The results, as we show in Fig. 9, indicate that the content-based methods generally yield better (smaller) content entropies, and ACHP yields the best performance because of its ability to handle the

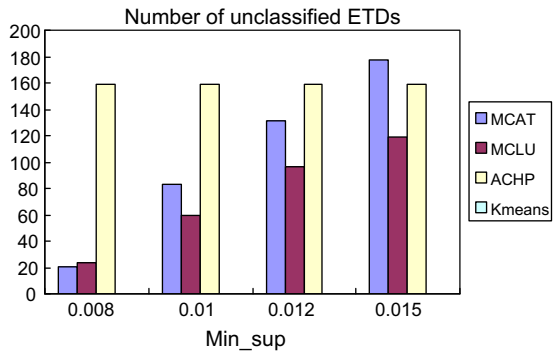

Fig. 10. Number of unclassified ETDs remaining with various methods, ETD data set.

high-dimensional data. Our two proposed methods generate similar (but higher) content entropies across various Min_{Sup} values. In particular, when Min_{Sup} increases, MCLU gradually outperforms MCAT, though with a very small difference. At larger Min_{Sup}, MCLU tends to give lower priority to usage clustering, because there are fewer usage-based hyperedges.

We report the numbers of unclassified ETDs in the different clustering schemes in Fig. 10. The trend is the same as that in Fig. 7 for experiments using the artwork data set. Therefore, ACHP exhibits the best content entropy, at the expense of more unclassified ETDs. In contrast, K-means comes second to ACHP in terms of content entropy but has the advantage of no unclassified ETDs.

4.3.3. Summary of results

When a digital library contains a large amount of textual content information, there is no clear winner among the various clustering methods: ACHP is the champion in terms of content entropy, K-means excels for the ratio of classified objects, and the two hybrid methods achieve the best usage entropy. However, if a digital library involves only a small amount of textual content information, our two proposed hybrid methods achieve the best usage and content entropies. For the best performance, we recommend a smaller Min_{sup} to reduce the number of unclassified multimedia objects. For example, as we show in Fig. 7, when we set Min_{sup} to 0.5%, only about one-sixth of artworks in the data set cannot be classified. Thus, MCAT achieves slightly better content entropy than MCLU, at the expense of slightly worse usage entropy, though the difference is small. We conclude that approaches that incorporate both textual content and usage data to construct indexes for a multimedia digital library yield better performance than those that use a single data source.

5. Conclusions

In this article, we address the problem of index construction for multimedia digital libraries by developing two index construction methods, MCAT and MCLU. These two methods employ primitive keywords and usage data to develop an index. The empirical experiments reveal that compared with traditional content-based clustering methods, our methods, when applied to digital libraries with limited textual data, generate indexes that exhibit better content and usage entropies. For digital libraries with rich textual information, our methods offer better usage entropy, though at the cost of slightly worse content entropy.

The performance of our proposed methods may depend on the quality of usage data. If the usage data contain mostly one-off searches or factoid searches, the index generated using our proposed methods may not retain good content entropy. To remedy this problem, some data cleaning strategies may be applied to the usage data to filter out subject-irrelevant sessions. In addition, the usefulness of our proposed methods has not been evaluated for digital libraries that collect other types of multimedia information, such as music and video. Finally, to improve efficiency, further research might integrate suitable data structures or techniques into the proposed methods.

References

Albuz, E., Kocalar, E., & Khokhar, A. (2001). Scalable color image indexing and retrieval using vector wavelets. *IEEE Transaction on Knowledge and Data Engineering, 13*(5), 851–861.

Berry, M. W., & Castellanos, M. (2007). *Survey of text mining. II: Clustering, classification, and retrieval*. London, UK: Springer.

Boley, D., Gini, M., Gross, R., Han, E.-H., Karypis, G., Kumar, V., et al (1999). Partitioning-based clustering for Web document categorization. *Decision Support Systems, 27*(3), 329–341.

Byrd, D., & Isaacson, E. (2003). A music representation requirement specification for academia. *Computer Music Journal, 27*(4), 43–57.

Chen, J., Ke, Y., & Ng, W. (2008). A survey on algorithms for mining frequent itemset over data streams. *Knowledge and Information Systems, 16*, 1–27.

Chen, H., Schatz, B. R., Ng, T. D., Martinez, J. P., Kirchhoff, A. J., & Lin, C. (1996). A parallel computing approach to creating engineering concept spaces for semantic retrieval: The Illinois digital library initiative project. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8), 771–782.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Creating adaptive Web sites through usage-based clustering of URLs. In *Proceedings of the workshop on knowledge and data engineering exchange* (pp. 19–25).

Gaber, M. M., Zavlavsky, A. Z., & Krisnaswamy, S. (2005). Mining data streams: A review. *ACM SIGMOD Record, 34*(2), 18–26.
Han, E. H., Karypis, G., Kumar, V., & Mobasher, B. (1997). Clustering based on association rule hypergraphs. In *Proceedings of SIGMOD workshop on research issues in data mining and knowledge discovery* (pp. 9–13).
Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques*. San Francisco, CA: Morgan Kaufmann.
Han, E. H., Karypis, G., Kumar, V., & Mobasher, B. (1998). Hypergraph based clustering in high dimensional data sets: A summary of results. *IEEE Bulletin of the Technical Committee on Data Engineering, 21*(1), 15–22.
Hwang, S.-Y., & Chuang, S.-M. (2004). Combining article content and Web usage for literature recommendation in digital libraries. *Online Information Review, 28*(4), 260–272.
Hwang, S.-Y., Hsiung, W.-C., & Yang, W.-S. (2003). A prototype WWW literature recommendation system for digital libraries. *Online Information Review, 27*(3), 169–182.
Jansen, B. J., Goodrum, A., & Spink, A. (2000). Searching for multimedia: Analysis of audio, video and image Web queries. *World Wide Web, 3*(4), 249–254.
Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of 10th European conference on machine learning* (pp. 137–142).
Karypis, G., Aggarwal, R., Kumar, V., & Shekhar, S. (1999). Multilevel hypergraph partitioning: Application in VLSI domain. *IEEE Transactions on Very Large Scale Integrated (VLSI) System, 7*(1), 69–79.
Korfhag, R. (1997). *Information storage and retrieval*. New York, NJ: John Wiley and Sons.
Li, Q., Myaeng, S. H., & Kim, B. M. (2007). A probabilistic music recommender considering user opinions and audio features. *Information Processing and Management, 43*(2), 473–487.
Manjunath, B. S., & Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8), 837–842.
Mehtre, B., Kankanhalli, M., & Lee, W. F. (1997). Shape measures for content based image retrieval: A comparison. *Information Processing and Management, 33*(3), 319–337.
Mierswa, I., & Morik, K. (2005). Automatic feature extraction for classifying audio data. *Machine Learning Journal, 58*(2–3), 127–149.
Mobasher, B., Cooley, R., & Srivastava, J. (1999). Creating adaptive Web sites through usage-based clustering of URLs. In *Proceedings of IEEE knowledge and data engineering exchange workshop* (pp. 19–25).
Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Discovery and evaluation of aggregate usage profiles for Web personalization. *Data Mining and Knowledge Discovery, 6*(1), 61–82.
Morchen, F., Ultsch, A., Thies, M., & Lohken, I. (2006). Modeling timbre distance with temporal statistics from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing, 14*(1), 81–90.
Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic retrieval. *Information Processing and Management, 24*(5), 513–523.
Salton, G., & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. New York, NJ: McGraw Hill Publishing Company.
Smeulders, A., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(12), 1349–1380.
Srivastava, J., Cooley, R., Deshpande, M., & Tang, P. (2000). Web usage mining: Discovery and applications of usage patterns from Web data. *SIGKDD Explorations, 1*(2), 12–23.
Swain, M. (1999). Searching for multimedia on the World Wide Web. In *Proceedings of the international conference on multimedia computing and systems* (pp. 32–37).
Tjondronegoro, D., & Spink, A. (2008). Web Search Engine Multimedia Functionality. *Information Processing and Management, 44*(1), 340–357.
Tsai, C.-F., McGarry, K., & Tait, J. (2006). CLAIRE: A modular support vector image indexing and classification system. *ACM Transactions on Information Systems, 24*(3), 353–379.
Vaidhyanathan, S. (2006). *Copyright Jungle. Columbia Journalism Review, 45*(5), 42.
Wactlar, H. D., Kanade, T., Smith, M. A., & Stevens, S. M. (1996). Intelligent access to digital video: Informedia project. *IEEE Computer, 29*(5), 46–52.
Wold, E., Blum, T., Keislar, D., & Wheaton, J. (1996). Content-based classification, search, and retrieval of audio. *IEEE Multimedia, 3*(3), 27–36.
Yang, Y., & Pederson, J. O. (1997). A comparative study on feature selection in text categorization. *Proceedings of the International Conference on Machine Learning, 412*, 420.
Zhao, Y., & Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings of the international conference on information and knowledge management* (pp. 515–524).