An ontology-based method for integrating heterogeneous itembanks

Chun-Wei Tsai and Shih-Pang Tseng
Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung 80424, Taiwan
{cwtsai87, flutetsen}@gmail.com

Yu-Sheng Yang
Engineering Science
National Cheng Kung University
Tainan 70101, Taiwan
ysy1018@gmail.com

Ming-Chao Chiang
Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung 80424, Taiwan
mcchiang@cse.nsysu.edu.tw

Chu-Sing Yang
Electrical Engineering
National Cheng Kung University
Tainan 70101, Taiwan
csyang@ee.ncku.edu.tw

Abstract

In this paper, we present a simple but efficient algorithm for integrating a collection of heterogeneous itembanks, called Heterogeneous Itembanks Integrator (HIBI). This algorithm is motivated by the desire to integrate itembanks provided by publishers to the users of an e-Learning system, which generally use different content structures. The proposed algorithm starts off with one of the itembanks as a reference itembank. All the items on the other itembanks are integrated into the reference itembank to create the so-called meta-itembank. Moreover, by treating the meta-itembank thus created as the reference itembank, it can be easily extended by using exactly the same algorithm. We also use the concept of ontology to share itemsets with other systems. The experimental results showed that the proposed algorithm can provide an extremely high quality result in terms of both the relevance of items and the computation time in Chinese itemsets.

Keywords: *Itembanks, e-Learning System, and ontology.*

1 Introduction

In order to provide a consistent content structure to both instructors and students of an e-Learning system, the content manager usually needs to integrate heterogeneous but somehow correlated itembanks into a single unified itembank. However, several questions arise: (1) Why are heterogeneous itembanks correlated? (2) Why do we want to integrate heterogeneous itembanks that may not be correlated into a single itembank? (3) When will the integration take place? In theory, itembanks (also known as itemsets or datasets) are originated from course material or textbooks, and they may have nothing in common! In practice, however, most, if not all, of the itembanks provide items that are correlated with each other for the following two main reasons. (1) One course taught by many instructors: In this case, instructors of the same course may have different background or come from different departments or even different universities. (2) One course using several reference books: The other case is that most, if not all, of the instructors will base their teaching material on some reference books, or students can find the related information from other books or articles. These two observations exist often enough in both the traditional learning and e-Learning environments, and they eventually answer the first question we stated previously. That is, why are different itembanks correlated? Now, if we move one step further, we can eventually construct a much more flexible environment by integrating several related itembanks together, and as a consequence, the integrated and clustered itembank will increase the quantity of an itembank. It can then be used to enhance the learning performance of students. For this reason, the question becomes how do we combine these different but correlated itembanks in such a way that the combined itembank would increase the items or in the worst case would not decrease the items available from all the individual itembanks? The proposed algorithm uses two structures to accomplish this task. One is tree-structured; the other is flat in the sense that no tree structure is imposed. No matter which structure is used, it is implemented as three required and one optional modules. The required modules are data cleanup module, similarity computation module, and item integration module. The optional one is the data abstraction

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.193

module that is used when the structure of the meta-itembank is flat.

2 Related Work

In general, data mining technologies play a key role in finding out the important or hidden information about the learning behavior of students on an e-Learning system [5][16]. A different kind of technology, information retrieval, can be used to analyze the data or documents on an e-Learning system, such as course material and itemsets. Vector Space Model [3] is usually used for computing the similarity between documents or itemsets, and it can also be combined with other data mining techniques to give a complete analysis. Recently, many researches on e-Learning have focused their attention on improving the interactivity and flexibility of such a learning environment [8]. The web 2.0 technology encourages e-Learning systems to generate, modify, and control the contents by people and to share the final results. There are many success cases in web 2.0, such as Wikipedia, social networks, and blogs [23]. Other technologies that also affect the learning performance of students include Item Response Theorem (IRT) [18], critical criteria [15], the feedbacks of the behavior [4], and so on.

Another important research issue is ontology [9, 20, 1, 25]. Ontology is a concept for defining specific knowledge and sharing the information with others. The technologies of eXtensible Markup Language (XML), Resource Description Framework (RDF), and Online Writing Lab (OWL) can usually help researchers accomplish the goal of ontology and semantic web. In [1], the authors used two ontology's to realize two course units on e-Learning systems. In another research [10], Guangzuo integrated the grid computing, ontology and other related technologies to present a more flexible educational platform architecture. In general, the ontology technology plays the role of making the contents of an e-Learning system be easily shared and reused [21, 19, 22, 2]. For instance, the course contents on different e-Learning systems may have different structures. How to integrate them has become a very important issue because if we can improve the information coverage of a system by adding more contents to the system. As a consequence, the system we describe herein not only integrates different itembanks into a single itembank for the e-Learning system in question, but it also provides a ontology module to import and export the itemsets to other e-Learning systems.

3 The Proposed Algorithm

To simplify our discussion that follows, throughout the rest of this paper except where no confusion is possible, the following notations will be used.

I The set of input itembanks. In other words, $I = \{I_1, I_2, \ldots, I_m\}$, where m is the number of itembanks.

I_i The i-th input itembank (also known as itemset or dataset). That is, $I_i = \{\tau_{i,1}, \tau_{i,2}, \ldots, \tau_{i,N_i}\}$ where N_i is the number of items in itembank i.

T The meta-itembank, i.e., the itembank of itembanks. Also known as meta-item or content concept tree.

T_i The item or content concept tree corresponds to itembank I_i.

These notations can be paraphrased as follows: As far as this paper is concerned, meta-itembank (also known as meta-item or content concept tree) represents the itembank that is composed out of two or more itembanks. Itembank (also referred to as itemset or dataset) represents a set of items. Item (or document) indicates a set of terms. And term is the smallest unit referring to a word or phrase or the like in an item or document.

3.1 Design and Implementation of HIBI

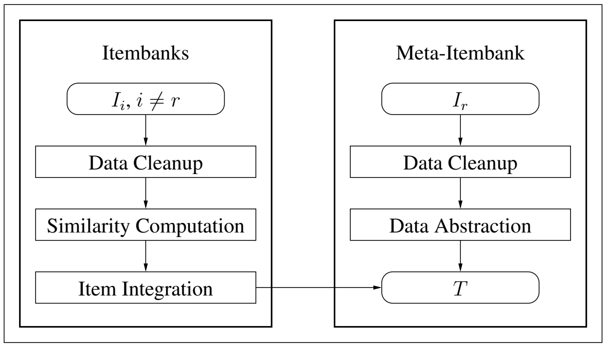

Figure 1. Framework of the proposed system.

Fig. fig:system shows the framework of HIBI, which is made up of three required and one optional modules, as we have previously described, in addition to the Chinese word segmentation system [6] for segmenting items and annotating all the terms segmented.

The data cleanup module is similar in spirit to Stemmer's algorithm, but it is responsible for removing all the stopwords in an item in Chinese in terms of the attributes of each term. For English, it is the stop word list that is used to remove all the useless words by Stemmer's algorithm [17]. The data abstraction module is used to compute an abstraction to represent the flat meta-itembank to speed up the computation of similarities in the similarity computation module. Once the abstraction is computed, the similarity computation module takes the responsibility of computing the similarities between new items and the abstraction representing the meta-itembank. Insofar as the HIBI is concerned, several approaches such as VSM [3] and the phrase based methods [24, 11] can be used to compute the similarity between items. In this paper, we use VSM. Then, the item integration module is responsible for putting all the new items into the meta-itembank T based on the similarity computed in the similarity computation module. Finally, the meta-itembank T, which plays a central role in a collaborative, interactive e-Learning system, is returned to the item management system of an e-Learning system, which would greatly enhance the learning contents available to the users of such an system.

3.2 The Proposed Algorithm HIBI

In this section, we present a novel clustering algorithm to integrate a collection of heterogeneous itembanks, called HIBI. This method is built on the notion of meta-itembank (also known as meta-item tree or content concept tree). In other words, it has been designed specifically for the classification of all the items in a set of itembanks each of which is itself stored as an item tree. For example, given a set of two itembanks I_1 and I_2, we can choose one of them as the initial meta-itembank and then apply the data cleanup and data abstraction modules to it. Without loss of generality, let us assume that I_1 is chosen. Otherwise, we can simply swap the role of I_1 and I_2. This will essentially clean up all the useless terms and compute an abstraction to speed up the similarity computation later on. Then, all the items of the other itembanks will be processed item by item using the data cleanup module to clean up all the useless terms, the similarity computation modules to compute the similarity between terms in I_2 and the abstraction of I_1, and the item integration module to put the new item into the right spot of I_1, i.e., into the most suitable node.

```
1  Procedure HIBI($I$)
2  {
3    Pick up one of the itembanks $I_r \in I$, $1 \le r \le m$, as the
         meta$-$itembank $T$.
4    For each itembank $I_i$, $1 \le i \ne r \le m$, do {
5      Let $\ell = 0$.
6      For each item $\tau_{i,j} \in I_i$, compute the similarity between
         $\tau_{i,j}$ and the abstraction of $T$.
7      Use the most similar node to determine the catalog of
         item $\tau_{i,j}$ at level $\ell$ and to reduce the search space.
8      If $\ell$ is less than the maximum level of $T$, then let
         $\ell = \ell + 1$ and goto step 7.
9      Else, add $\tau_{i,j}$ into the leaf of $T$.
10   }
11   Return $T$;
12 }
```

Figure 2. Outline of HIBI.

Fig. fig:proutline gives an outline of HIBI. The input to HIBI is a set of itembanks I, and the output is the meta-itembank T, as defined previously. Now, given the input I and the expected output T, the proposed clustering algorithm can be outlined as follows:

Insofar as an e-Learning system or a very large data analysis system are concerned, the response time and the accuracy rate are probably the most important issues to be addressed. Fig. fig:proutline shows how HIBI works to create a brand new meta-itembank or to update an existing one. Initially, on line 3, HIBI picks up one of the itembanks in I and uses it as the meta-itembank T. Then, the loop beginning on line 4 and ending at line 10 will take care of putting all the items in all the itembanks except I_r into T. First, on line 5, ℓ is initialized to 0, meaning that we are at the root of the meta-itembank T. Then, on line 6, the similarity between the item to be put into T and the abstraction of T is computed. Then, on line 7, the most suitable category at level ℓ is assigned to the item to be put into T. Then, the search will continue, but with a much smaller search space, i.e., the subtree of the node to which the item is assigned. Then, if the node is not a leaf, the search will be continued to the next level until we reach the leaf node, as lines 7 and 8 show. Finally, on line 11, the meta-itembank T is returned, to the item management subsystem of HGLS [13] as far as the testbed is concerned.

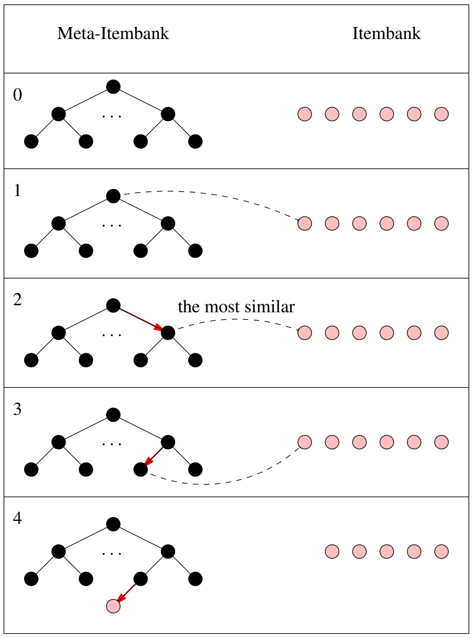

Figure 3. A simple example illustrating how HIBI works when a tree-structured meta-itembank is used. See the text for more detailed explanation.

3.3 Itemset Management

To illustrate how HIBI works, we present a simple example as shown in Fig. fig:example. Step 0 in Fig. fig:example shows that the meta-itembank T and items in an itembank, say, I_i. In steps 1–4, the most suitable cluster for, say, $\tau_{i,j}$, is found, beginning at level 0 of the item tree and heading all the way down to the leaf node to which the item is assigned. Level 0 may consist of courses or chapters. Steps 2 to 4 are essentially the same as step 1 except that they finds the most suitable child, grandchild, and so on. Finally, when the leaf at level $d - 1 = 4$ of the tree T is reached, HIBI adds the item $\tau_{i,j}$ into T. This explains how an item is eventually added to the meta-itembank and how the learning contents is increased.

In this paper, we also use the ontology technology to share these itemsets with other e-Learning systems. The proposed system generates the itemsets in OWL (Web Ontology Language) [7]. Or more precisely, we define the itemsets, contents, and relations in OWL to make the itemsets themselves more scalable and reusable on our e-Learning system.

4 Experimental Results

This section presents the experimental results of the proposed algorithm, HIBI. The empirical analysis was conducted on a 2.8GHz Intel Pentium4 machine with 512MB of memory running Ubuntu 8.04 with Linux kernel 2.6.24. Moreover, all the programs are written in C++ and compiled using g++ 4.2.3. In this paper, we use HGLS [13], an e-Learning system we built, as a testbed of the proposed algorithm.

4.1 Datasets and Parameter Settings

In what follows in this paper, we will use two different approaches to evaluating the itembanks from Kang Hsuan Publishing [14] and Han Lin Publishing [12]. These two books are edited following the same textbook standard; so the domains covered by these two books are eventually identical. But the structure of the two books is different. The itembank provided by Han Lin Publishing has 6 chapters, 13 sections, and 697 items while the itembank provided by Kang Hsuan Publishing has 3 chapters, 15 sections, and 464 items. Before we proceed, these two datasets need to be segmented into terms using the Chinese word segmentation system [6]. To get rid of the verbality, we do not use the stop word list to remove all the useless words. Instead, in this paper, we retain three kinds of terms that are, respectively, noun (N), adjective (A), verb (V) of Chinese keyword and remove the other kinds of terms.

There are total 11,705 terms in the test itembanks used in this paper after segmentation. Then, after removal of the replicated terms, 2,578 terms are left. Two search mechanisms are proposed in this paper. One is flat; the other is tree-structured. Using the flat mechanism, the similarity of an item is compared directly with all the items in the leaf of the meta-itembank. Using the tree-structured mechanism, it is the tree structure of the meta-itembank that is searched starting with the root and going all the way down to the leaf. The distinguishability is expected to be improved but the accuracy is not influenced by using tree-structured mechanism.

4.2 Experimental Results and Performance Analysis

Two methods are used to evaluate the performance of the proposed algorithm. One of them is based on the distinguishability while the other is based on the distribution of one of the two itembanks with respect to the other. The distinguishability between a given item τ and nodes in the meta-itembank is defined as follows:

- First, the two nodes in the meta-itembank, say, a and b, that are most similar to the given item τ are found and denoted, respectively, by $\mathrm{sim}(\tau, a)$ and $\mathrm{sim}(\tau, b)$.
- Then, the distinguishability δ is computed as

$$\delta = \frac{|\,\mathrm{sim}(\tau, a) - \mathrm{sim}(\tau, b)|}{\max(\mathrm{sim}(\tau, a), \mathrm{sim}(\tau, b))}$$

where δ ($0 \leq \delta \leq 1$) indicates the certainty of the categorization and the similarity between items x and y, denoted $\mathrm{sim}(x, y)$, is computed using VSM (or TFIDF). In this paper, we are assuming that the threshold for the certainty of categorization is 0.2. Thus, if $\delta < 0.2$, then the item τ is assumed to be indistinguishable. Otherwise, it is distinguishable. The threshold is, however, tunable. The results show that Kang Hsuan itembank is centralized in the middle part of Han Lin itembank while Han Lin itembank is distributed at the two ends of Kang Hsuan itembank. This indicates that different editors may have different views on the same materials. In other words, for a course, even if the same course standard is used, the structure of the textbook—thus the structure of the itembanks—provided by different editors may be highly correlated but quite different. Thus, if we can efficiently integrate all such itembanks, then the diversity of the itembanks can be easily enhanced.

To measure the accuracy rate of item categorization, we conducted two experiments. First, all the items in Han Lin itembank, which plays the role of itembank to be integrated, are selected. Then, the items selected are reinserted back into the Han Lin itembank, which now plays

Table 1. The distinguishability rate.

method	h2k	k2h
flat	78.77%	79.32%
tree-structured	87.37%	93.53%

Table 2. The accuracy rate.

method	h2h	k2k
flat	98.24%	98.21%
tree-structured	98.09%	97.33%

Table 3. The average running time.

method	h2k	k2h	h2h	k2k
flat	0.171	0.094	0.141	0.116
tree-structured	0.131	0.088	0.131	0.087
(Δ_T)	(-23.4%)	$(\; -6.4\%)$	$(\; -7.1\%)$	(-25.0%)

the role of meta-itembank. If the item is distinguishable but not inserted back into its original section, the insertion is counted as an error. Items, which are indistinguishable (i.e., $\delta < 0.2$) is not taken into account in the computation of the accuracy rate. The same process is repeated for Kang Hsuan itembank. The results are as shown in Table tab:notrec.

Table fig:acc shows the accuracy rate in the integration of itembanks. As Table fig:acc shows, the flat and tree-structured mechanisms are not that different in terms of the accuracy rate. Table tab:runtime shows the average running times for integrating both the flat and tree-structured meta-itembanks for 50 runs. In Table tab:runtime, Δ_T, which represents the speedup of the tree-structured meta-itembank with respect to the flat meta-itembank in percentage, is computed as follows:

$$\Delta_T = \frac{T_t - T_f}{T_f} \times 100\%$$

where T_t indicates the average running time of tree-structured meta-itembank and T_f the average running time of flat meta-itembank as given in Table tab:runtime. In brief, the flat meta-itembank takes $O(n)$ time whereas the tree-structured meta-itembank takes $O(\log_m n)$ time where n is the size of meta-itembank and the subscript m indicates the degree of the meta-itembank represented as an item tree.

5 Conclusion

This paper gives a detailed description of how HIBI can be used to integrate a collection of heterogeneous itembanks. All of our experimental results showed that the proposed algorithm can provide a very high accuracy rate in the integration of itembanks, which can be used to help existing e-Learning systems integrate itembanks from different systems or environments. Another contribution of this paper is HIBI using the concepts of ontology to manage the itemset information. For this reason, HIBI provide a sharable and reusable method for different e-Learning System. In this paper, we integrate two heterogeneous itembanks to increase the interactivity between instructor and instructor. Students using our e-Learning system can return their response such as learning performance, learning behavior, and so on to the system or the instructor. Instructors can use this information to change their teaching strategies. In the future, our primary goal is to generalize the proposed algorithm to any learning contents related areas such as Wikipedia, blogs, search engines, and so on to make our system even more interactive and flexible.

References

[1] M.-H. Abel, A. Benayache, D. Lenne, C. Moulin, C. Barry, and B. Chaput. Ontology-based organizational memory for e-learning. *Educational Technology & Society*, 7(4):98–111, 2004.

[2] G. Angelova, O. Kalaydjiev, and A. Strupchanska. Domain ontology as a resource providing adaptivity in elearning. In *OTM Workshops*, pages 700–712, 2004.

[3] R. Baeze-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[4] C.-H. Chen. The user interface design of networked computer assisted cooperative/collaborative learning. Master's thesis, National Chiao Tung University, 1995.

[5] C.-H. Chen, G. Horng, and C.-H. Hsu. A novel private information retrieval scheme with fair privacy in the user side and the server sidee. *International Journal of Innovative Computing, Information and Control*, 5(3):801–810, 2009.

[6] Chinese Document Segmentation, 2008. http://ckipsvr.iis.sinica.edu.tw/.

[7] Deborah L. Mcguinness and Frank van Harmelen, OWL Web Ontology Language Overview, 2004. http://www.w3.org/TR/owl-features/.

[8] K. Eguchi, S. Kurebayashi, H. Zhu, T. Inoue, and F. Ueno. A self-learning support system for pupils based on a fuzzy scheme. *International Journal of Innovative Computing, Information and Control*, 4(10):2441–2450, 2008.

[9] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[10] C. Guangzuo, C. Fei, C. Hu, and L. Shufang. Ontoedu: A case study of ontology-based education grid system for e-learning. *The Official Journal of Global Chinese Society FOR Computers in Education*, pages 59–72, 2004.

[11] K. M. Hammouda and M. S. Kamel. Efficient phrase-based document indexing for web document clustering. *IEEE Transactions on Knowledge and Data Engineering*, 16(10):1279–1296, 2004.

[12] Han Lin Publishing, 2002. http://www.hle.com.tw/.

[13] HGLS, 2002. http://hgls.tn.edu.tw/hgls/.

[14] Kang Hsuan Publishing, 2002. http://www.knsh.com.tw/.

[15] T.-M. Lin. The research of developing quality evaluation criteria of kids learning websites. Master's thesis, National University of Tainan, 2001.

[16] T. Miyoshi and H. Joichi. Comparison with fuzzy reasoning and modified tf-idf in page grouping for the result of web retrieval. *International Journal of Innovative Computing, Information and Control*, 3(2):307–317, 2007.

[17] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[18] H. S. Ronald K. Hambleton. *Item response theory :principles and applications*. Kluwer Academic Publisher, Boston, 1983.

[19] S. Staab and R. Studer, editors. *Handbook on Ontologies*. International Handbooks on Information Systems. Springer, 2004.

[20] L. Stojanovic and S. Staab. elearning based on the semantic web. *World Conference on the WWW and Internet*, 2001.

[21] L. Stojanovic, S. Staab, and R. Studer. E-Learning Based on the Semantic Web. In *Proceedings of the World Conference on the WWW and Internet*, pages 23–37, 2001.

[22] J. Tane, C. Schmitz, and G. Stumme. Semantic resource management for the web: an e-learning application. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 1–10, 2004.

[23] J. C. R. Tseng, G.-J. Hwang, P.-S. Tsai, and C.-C. Tsai. Meta-analyzer: A web-based learning environment for analyzing student information searching behaviors. *International Journal of Innovative Computing, Information and Control*, 5(3):567–579, 2009.

[24] O. Zamir and O. Etzioni. Grouper: A dynamic clustering interface to web search results. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11-16):1361–1374, 1999.

[25] A. Zouaq and R. Nkambou. Building domain ontologies from text for educational purposes. *IEEE Transactions on Learning Technologies*, 1(1):49–62, 2008.