

# Content-based Navigation in Web-based Learning Applications

**Diana Purwitasari, Yasuhisa Okazaki, Kenzi Watanabe**

*Graduate School of Science and Engineering*

*Department of Information Science, Saga University, Saga, Japan*

{diana, okaz, watanabe}@ai.is.saga-u.ac.jp

**Abstract:** Rapid growth of the Internet make Web-based applications becomes a new means to learn. Authors of Web-based learning applications with multiple resources provide navigation to help users understanding structured idea of learning topics. Since the increasing of resources is such a growing field, navigation map of learning topics would be out of date if it is manually constructed. We define content-based navigation as a sequence list of topics and sub topics hierarchically structured from a collection of documents which is created without relying on human power. Content-based navigation not only provides subjects domain exists in the collection but also offers guidance of relevant items for users who want to learn particular subjects. In this paper, we present a framework for generating content-based navigation from resources applied in Web-based learning applications. We use data mining techniques to extract existing but hidden topics. We do hypergraph partitioning to derive the topics from hyperedges of words correlation which connecting vertices of topics corresponding words. Then we employ agglomerative clustering to merge any overlapping subjects of topics and produce a hierarchical topics list of navigation.

**Keywords:** content-based navigation, topic extraction, data mining, generating method

## 1. Introduction

Rapid growth of the Internet and availability of abundance information resources make Web-based learning applications become a new means to learn. Users can access learning materials of the applications without time and distance barriers via Web services despite lack of direct interaction between teachers and students. Moreover using existing Web pages as learning materials is pertinent idea to minimize authoring cost of learning applications development. However browsing through such amount could cause a problem of information overload. Some users prefer to read particular subjects from already focused and organized topics. Other users might not yet have enough structured idea of learning topics and can not decide which should be read first.

Web-based learning applications that reuse learning materials from multiple resources should take into account of creating navigation as an important task to help users understanding structured idea of learning topics. Each resource could have navigation at the ready which is usually designed following its own resource's point of view. Then again the authors of current application (with multiple resources) may not be able to directly reuse those designed navigations as well. It happens because there is a possibility of different point of view between current (multiple) resources and each of the original ones. Moreover since the increasing of resources is such a growing field, navigation map of learning topics would be out of date if it is manually constructed.

We define content-based navigation as a sequence list of topics and sub topics hierarchically structured from a collection of documents which is created without relying on human power. In this paper, we focus to analyze document contents and show how to bring generated navigation satisfying the information structure as guidance for the users through collection of learning materials.

## 2. Study Literatures

Many research efforts have been engaged to bring structured representation to a large collection of documents in the Web. Here are some of those works [1, 2, 3, 4, 5, 6, 7] which influence our framework to generate content-based navigation. Mendes, et al. [1] and Chen, et al. [2] with their defined needs in Web-based learning applications consent that tools are required to organize documents in order to determine the relevant ones for users. Prime function of the tools is to capture associated subjects domain exists in the collection of documents [1, 2, 3, 4, 5]. Subjects domain or semantic hierarchy or ontology signifies a description of the concepts or topics and their relationships for the purpose of enabling knowledge sharing. The intuition problem is to identify topics or subjects which are frequently discussed in the collection. Mendes, et al. represent subjects with fuzzy clustering of terms [1], Chen, et al. use multivariate data analysis method Principal Component Analysis (PCA) for mining concepts of principal or main terms [2], while Clifton, et al. cluster sets of common words or terms (frequent itemsets) using data mining techniques to identify topics [3]. With Internet booming, collection of information is distributed on the Web as well. The Web can be abstracted as a directed labeled graph consists of Web pages as nodes whose edges carry links information from the source node to the target node. Halkidi, et al. [4] and Zhu, et al. [5] analyze link structures and exploit semantics of the links in order to contextualize learning subjects.

Though there are various ways on how to represent subjects domain basically the representation is like a graph form [1, 2, 3, 4, 5]. Mendes, et al. label the nodes with clustering identifier [1]. Chen, et al. employ single principal term as a node [2]. It can be applied since their collection contains journal and conference papers thus each term or a keyword listed in one research article represents one essential concept. Zhu, et al. cluster Web pages by means of a distance similarity function regards to in-links and out-links then label the clusters from extracted terms surrounding link tags [5]. Zhu, et al. perform threshold for eliminating rare terms and use the rest of terms as node label. A little bit different with the others, Clifton, et al. [3] and Halkidi, et al. [4] reduce the number of concepts without losing information by using the ontology of WordNet [8]. However Clifton, et al. take terms of topics from all document contents whereas Halkidi, et al. only use terms surrounding link tags. In brief, [3, 4, 5] use set of terms as label of topics. Differ than previously mentioned works [1, 2, 3, 4, 5], Wissner-Gross [6] and Reinhold [7] prefer a directed path to show subjects domain. To be more exact their objectives are not derived from *how to organize documents in a domain of subjects* but *how to guide users through scenarios of topics for learning particular subjects*. Therefore Wissner-Gross and Reinhold apply title of Web pages for labeling.

Our framework of content-based navigation proposes a way to make the best use of both underlying models: domain of subjects and learning scenarios of topics. In Web-based learning applications [1, 2] with a need to perform navigation of learning materials from multi resources, we would like to not only provide a domain of subjects extracted from collection of documents [1, 2, 3, 4, 5] but also offers guidance for learning [6, 7]. Since the abstraction of the Web as graph contains information of contents (Web pages) [1, 2, 3] and links (hyperlinks) [5, 6, 7], we believe that both kind of information should be equally exploited.

We make adaptation of topic identification with data mining techniques [6] without any prior action like utilizing WordNet [3, 4] to help defining subjects domain as well. We demonstrate that the similarity function of document and cluster [3] is not only for merging topics. With some adjustments the function could be used for structuring and representing document into topics within navigation as well. We would like to show that when even typical documents may have membership in several domain subjects and fuzzy clustering could be suitable to partition that kind of collection [1], our framework is succeeded to apply a straightforward agglomerative clustering to handle it.

### 3. Content-based Navigation: A Generating Method

We use data mining techniques to extract existing but hidden topics which are frequently discussed in the collection. The issue is how to cluster words given that a certain set of corresponding words will identify a topic. We list frequent itemsets (sets of common words), construct hypergraph of topics with the vertices representing words and the edges representing strength relation between words, then partition the hypergraph to extract topics. After that is to cluster the extracted topics in order to hierarchically produce a sequence list of topics.

#### 3.1 Data Mining for Extracting Topics

Documents are mainly characterized by indexed terms. We use information retrieval (IR) techniques [9] (indexing, removing stop words, stemming, and weighting) to create a comprehensive inverted index of feature terms. Indexing is to list distinct terms of document contents in the collection. Some terms or words that typically modify other words but carry no inherent meaning themselves, such as adverbs, conjunctions, or prepositions are called stop words. Removing stop words is to clear those unimportant terms in order to reduce number of terms in the list. Terms with a common stem usually have similar meanings, i.e. learned, learning. Stemming, suffix stripping, is used to avoid similar terms being used as different feature terms. The inverted index also records some statistics to show that some terms have more weight or important than others, i.e. document frequency: number of documents which contain certain term. Our weighting refers to combination of term and document frequency (TF-IDF, [9]).

We begin those IR preprocessing techniques to retrieve important terms as candidates of frequent itemsets. Data mining approach generally list frequent itemsets that satisfying minimum support and minimum confidence. We introduce usage of other filters to set apart significant frequent 2-itemsets. First is indispensable feature value of TF-IDF term weight [9], and second is our own weighting schema of term entropy. All terms with term weight and term entropy value less than some thresholds should be ignored.

Entropy value measures uncertainty state. We think entropy value of a term reflects the effectiveness of the term in identifying certain document to others. This is derived from selection process of most useful attribute to test at each decision node in the decision tree algorithm for classifying [10]. For term entropy calculation we assume that there are only two classes as target classification: (i) whether the term is one of important terms in the currently observed document or (ii) in the contrary that the term is important for any other documents except the currently observed one. Attribute for classification of a term belongs to (i) or (ii) is test condition whether frequency of the term in currently observed document (term frequency) will be less or more than average value of term frequency from all documents in the collection.

Filtering of term weight and term entropy value makes sure only noteworthy terms should compose frequent 2-itemsets. Even then frequent 2-itemsets should at least

sufficiently enough on conditions of *support* and *confidence* [11]. *Support* ignores any frequent 2-itemsets that occur in few documents to ensure the ones that often occur are worthy of attention. *Confidence* makes further analysis to find whether the presence of the words pair is strongly significant to become potential frequent 2-itemsets. For identifying topics, frequent 2-itemsets that fall under these constraints should be eliminated [3]: (a) its support is less than average support and its confidence is less than average confidence; (b) its support is less than a summation of average and standard deviation of support; (c) its confidence is less than a summation of average and standard deviation of confidence.

After we have listed important frequent 2-itemsets we will construct hypergraph from those frequent 2-itemsets. A pair of vertices stands for a frequent 2-itemsets and its edge represents confidence. A hyperedge which can connect more than two vertices describes a complex relationship of some words corresponding to a topic. We do hypergraph partitioning to do segmentation of hyperedges for extracting the topics.

### 3.2 Agglomerative Clustering for Structuring Topics

Our problem demand is to produce a domain of subjects (i.e. graph form) extracted from collection of documents which offers guidance for learning (i.e. path form). In order to transform graph form into path form we change the angle of view such that graph will be restructured into tree. For that purpose agglomerative clustering would organize topics in nested groups of merged topics which can be displayed as a tree. Parent-child relationship among the nodes in the tree can be viewed as topics and sub topics in a subject hierarchy. By merging some too fine-grained topics we get clusters or groups of coarsely topics. The too fine-grained topics can be seen as sub topics of a new coarsely topic.

Though assigning document membership to certain topic or sets of topics is not part of clustering, however it has effects on process of merging [3]. We use extracted topics as cluster seeds in initialization. In the first iteration a cluster contains single topic but for the next iterations a cluster could become cluster of topics as a result of merging topics. Our assumptions in clustering initialization are that each document can only belong to exactly one cluster and any cluster without document members will be removed. Because of those assumptions there is possibility that some initial clusters do not have any document members and will be left out during tree construction.

Let a cluster consists of single topic or sets of topics is being treated like a document called topics-document,  $tpc_i$ . Note that all corresponding words of each too fine-grained topic will become the corresponding words of new coarsely topic. To know whether a document,  $d_i$ , is a member of a topics-document is the same with to know how similar currently observed document with topics-document (Eq 1). We calculate their similarity,  $sim(d_i, tpc_i)$ , based on their common words (Eq 1) [3].

Since the topics reside in clusters, thus a process to merge topics means to merge clusters. Document similarity value,  $sim(d_i, tpc_i)$ , is a weight value derived from any term frequencies of terms inside the document. Note only similar terms which exist in both document and topics-document are being counted. In this agglomerative clustering iteration, topics which will be merged are the ones with highest similarities (Eq 2) [3]. That is to say similarity between topics depends on similarity of documents which refer to the same topics. It will handle issue of documents that very likely have multi topics in a document.

Figure 1 demonstrates calculation examples of merging processes. From extracting topics processes we can see 9 significant terms and 3 topics that exist within collection of 5 documents. Note vertices relation between  $term_1$  and  $term_2$  represents a frequent 2-itemsets while  $term_1$ ,  $term_2$ ,  $term_3$ ,  $term_4$ , and  $term_7$  illustrate hyperedge of certain topic identified as

TopicA. By using Eq 1 and Eq 2, it is showed that TopicA is most likely going to be merged with TopicC.

$$sim(d_i, tpc_t) = \sum_{k \in t} \frac{tf_{ik} \times (\log \frac{N}{n_k})^2}{\sqrt{\sum_{j \in t} \log \frac{N}{n_j}} \times \sqrt{\sum_{j \in t} (tf_{ij})^2 (\log \frac{N}{n_j})^2}} \dots\dots\dots (1)$$

$$sim_{ab} = \frac{\sum_{i \in docs} sim(d_i, tpc_a) \times sim(d_i, tpc_b)}{N \times \sum_{k \in docs} sim(d_k, tpc_b)} \dots\dots\dots (2)$$

- $d_i$  is document  $i$ ;
- $tpc_t$  is topics cluster  $t$  consisting single topic or a set of topics extracted with data mining;
- $tf_{ik}$  is frequency of term  $k$  in  $d_i$ ;
- $N$  and  $n_k$  is number of documents in the collection and the ones that consisting term  $k$ ;
- Note  $\log \frac{N}{n_k}$  is the base 2 logarithmic of  $\frac{N}{n_k}$ ;
- $k \in t$  is all corresponding terms exist in topics cluster  $t$ .
- $sim_{ab}$  is similarity between topics cluster  $a, tpc_a$ , and topics cluster  $b, tpc_b$ ;
- $i \in docs$  is document  $d_i$  in the collection of documents.

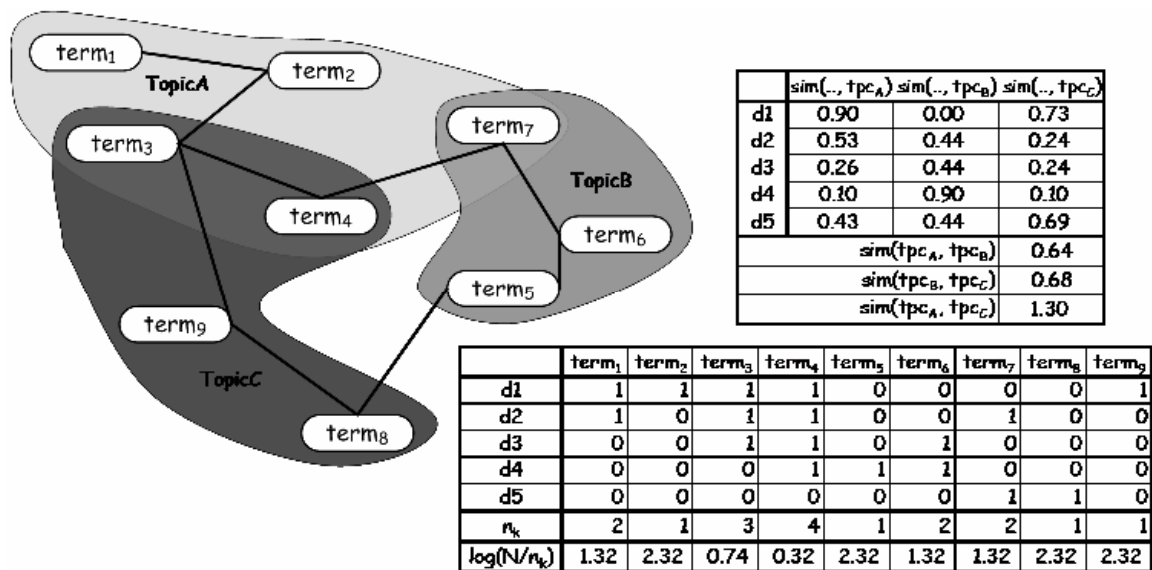


Figure 1. Illustration of merging processes

### 3.3 Representing Navigation

Representation of clusters and sub clusters is an ending issue for generating content-based navigation. Consider navigation looks like a table of contents, clusters in final clustering iteration become first-level headers or chapter titles while clusters in iteration before final clustering iteration become second-level or sections and so on. Consequently, cluster seeds are leaf nodes in the tree and have the most depth-level. The first problem here is how many levels should be considered to avoid over fitting of subjects domain and represent a moderate navigation but still specific enough for learning. For other problems let assume in second-level there are four clusters. The question is which cluster should be the first section. This is the second problem about how to sequence clusters in the same level. The third

problem is how to determine document representation of a cluster. Resulted clusters consist of single topic or a set of topics. Even in a case of single topic we can not easily assign document which has highest similarity with the topic as a representation because the document might have more specific subjects and make the document is more suitable as document representation to one of sub clusters.

Members of our clusters are topics. We do not know for sure whether to merge some topics is a correct decision or not because there is no information of class labels and class members for exact clusters. To evaluate how well the results of this agglomerative clustering without reference of external information is to make sure that within distances on members in the same cluster are minimized while between distances on different clusters are maximized. We compute variance ratio of between and within distance in clustering results. The smaller ratio value shows that the clustering results are better. We solve the first problem by choosing to list only the clusters retrieved after certain iteration when its variance ratio value tends to become smaller compare with previous iteration.

We employ link analysis to determine sequence of clusters in the same iteration level. Web pages with more *in-links* will receive higher rank in which its importance can be stated into a weight score [12]. In a previous sample case cluster which becomes the first section should imply having a set of document with the overall weight scores higher than other clusters. At the beginning, let topics-document consists terms from corresponding words of topics in a cluster to find the order of a cluster. After that we retrieve documents which show higher similarity to topics-document (Eq 1). We will average weight scores of some documents which having higher similarities. Because each cluster can only be represented by single document, we will only consider a number of documents with the same number of nodes exist in a sub tree for computing the average value. Notice that the sub tree has cluster of topics-document as its sub root. Cluster rank will be an average value of weight scores belong to those selected documents.

To find document representation of a cluster begins with listing of documents similar to topics-document of cluster arranged in a descending order (Eq 1). We search the representation from listing of documents by traversing the list in depth-first. We compute similarity between each traversed document and topics-document ignoring any document which has already become representation of another cluster. Each time a cluster is visited we will check that similarity (Eq 1). We also check similarity between document and sub clusters of currently visited cluster before assigning the document as cluster representation. This will avoid misrepresent because it is possible that the document is more suitable as sub cluster representation.

#### **4. Adaptive Content-based Navigation**

We emphasize the term of adaptive in educational fields via Web services as adaptive information filtering [13] which is to find items that are relevant to user interests in large collection of documents. Interests of the users can be derived from modeling user information like knowledge, goal and others. For this time being we interpret that content-based navigation should adapt when context of the user interests changes like in a time user clicks a topic to know more about certain topic.

An event when users click a topic means user would like to browse the topic. Consider browsing to any topic as a kind of searching for that topic, such event will evoke an inquiry using feature terms within the document representation of selected topic as query keywords. We intend to apply combination ranking factors of link analysis [12] and content analysis that still retaining spatial information of search keywords [14, 15]. Recording the positions of each term appeared in documents will be useful in keeping spatial information. Information of term positions make sure that terms equal to search keywords which occur

close together will be given higher weight. Afterward, inquiry results act as *dummy* collection with subjects contextualize to user interest in which we could produce new content-based navigation suitable to current context. However it is still in the early stages because of issue on structural matching between old content-based navigation and new content-based navigation. If the old one can be placed as sections and the new one as sub sections in the whole subjects domain then both navigation have structural matching. Regrettably some preliminary experiments show that the second navigation does not match structurally with the first one, yet.

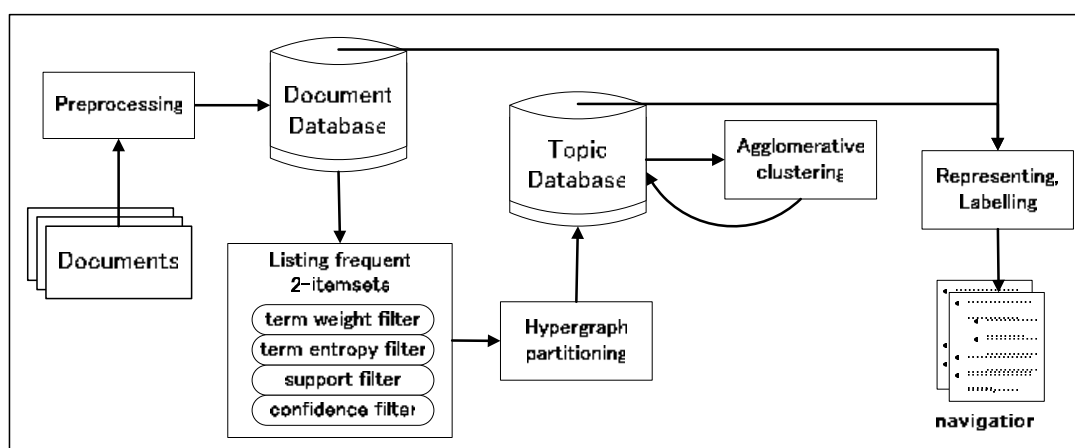


Figure 2. Methods to generate content-based navigation

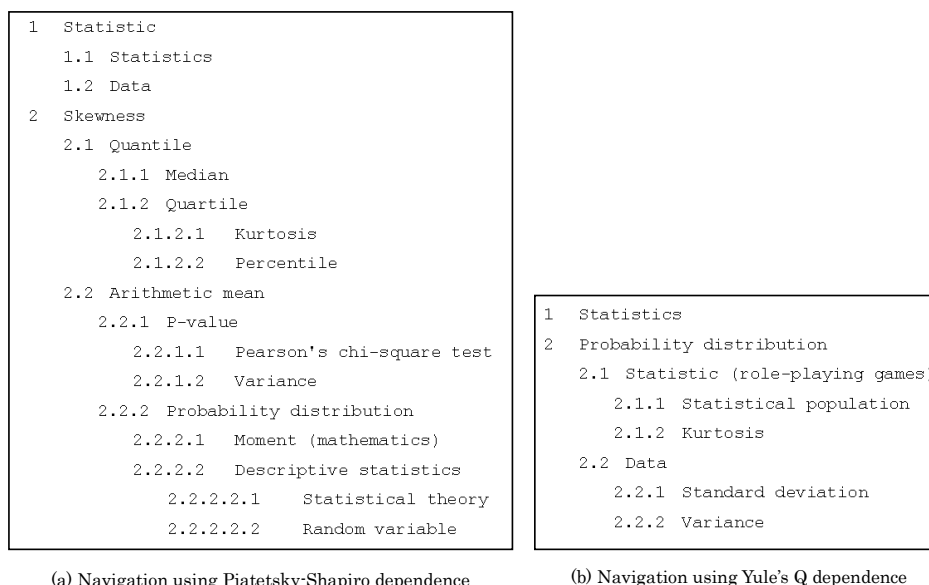


Figure 3. Samples of content-based navigation with different dependence functions

## 5. Prototype

We have implemented processes to generate content-based navigation (Figure 2) with small collection of 30 Wikipedia articles crawled from main page Statistic (URL: <http://en.wikipedia.org/wiki/Statistic>). We use Porter Stemmer algorithm in stemming process and calculate TFIDF (term frequency/inverse document frequency) in weighting process [9]. To list frequent 2-itemsets we consider a selection of some dependence measures [16]: Yule's Q, Yule's Y, Two-Way Support, Linear Correlation Coefficient, Piatetsky-Shapiro, and Information Gain. We choose those measures based on key

properties a good dependence measure should satisfy [16] which carefully selected in a way that the properties are appropriate to our problem domain. Then we execute shmetis library in hMetis tool for hypergraph partitioning [17] in the processes.

The results are shown in Figure 3. We take title of Web page representation as topic subject. From experiments, frequent 2-itemsets listed by Piatetsky-Shapiro dependence measure shows variance ratio of between and within distances with more stable tendency to smaller value compare to the others. It shows with Piatetsky-Shapiro dependence good clusters have been formed even 4 iterations before final iteration unlike clusters from Yule's Q measure. In fact Yule's Q results into almost two times number of frequent itemsets compares to Piatetsky-Shapiro which mining the least but giving more detail navigation.

## 6. Conclusions

We have described a framework for generating content-based navigation applied in Web-based learning applications. Our framework makes the best use of both models: how to organize documents in a domain of subjects and how to guide users through scenarios of topics for learning particular subjects. Our prototype implementation shows that generate methods are indeed able to construct content-based navigation among a test collection.

## References

- [1] Mendes, M. E. S., & Jarrett, W., & Prnjat, O., & Sacks, L. (2003). Flexible Searching and Browsing for Telecoms Learning Material. *In IST'2003: Proc. of the 2003 Intl. Symp. on Telecommunications.*
- [2] Chen, N.S., & Kinshuk, & Wei, C.W., & Chen., H.J. (2006). Mining e-Learning Domain Concept Map from Academic Articles. *In ICALT '06: Proc. of the Sixth IEEE Intl. Conf. on Advanced Learning Technologies.* 694-698.
- [3] Clifton, C., & Cooley, R., & Rennie, J. (2004). TopCat: Data Mining for Topic Identification in a Text Corpus. *IEEE Trans. on Knowledge and Data Engineering*, 16(8):949-964.
- [4] Halkidi, M., & Nguyen, B., & Varlamis, I., & Vazirgiannis, M. (2003). THESUS: Organizing Web Document Collections based on Link Semantics. *The VLDB Journal*, 12(4):320-332.
- [5] Zhu, J., & Hong, J., & Hughes, J.G. (2004). PageCluster: Mining Conceptual Link Hierarchies from Web Log Files for Adaptive Web site Navigation. *ACM Trans. Inter. Tech.*, 4(2):185-208.
- [6] Wissner-Gross, A.D. (2006). Preparation of Topical Reading Lists from the Link Structure of Wikipedia. *In ICALT '06: Proc. of the Sixth IEEE Intl. Conf. on Advanced Learning Technologies.* 825-829.
- [7] Reinhold, S. (2006). WikiTrails: Augmenting Wiki Structure for Collaborative, Interdisciplinary Learning. *In WikiSym '06: Proc. of the 2006 Intl. Symp. on Wikis*, 47-58.
- [8] Miller, G.A., & Beckwith, R., & Fellbaum, C., & Gross, D., & Miller, K.J. (1990). Introduction to WordNet: An On-Line Lexical Database. *Intl. J. Lexicography*, 3(4):235-244.
- [9] Yates, R.B., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. *Addison-Wesley.*
- [10] Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1:81-106.
- [11] Agrawal, R., & Imielinski, T., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. *In SIGMOD '93: Proc. of the 1993 ACM SIGMOD Intl. Conf. on Management of Data*, 207-216.
- [12] Page, L., & Brin, S., & Motwani, R., & Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. *Technical Report*, Stanford Digital Library Technologies Project.
- [13] Brusilovsky, P., & Peylo, C. (2002). Adaptive and intelligent Web-based Educational Systems. *Intl. J. of Artificial Intelligence in Education: Special Issue on Adaptive and Intelligent Web-based Educational Systems*, 13(2-4): 159-172.
- [14] Park, L. A., & Ramamohanarao, K., & Palaniswami, M. (2004). Fourier Domain Scoring: A novel document ranking method. *IEEE Trans. on Knowledge and Data Engineering*, 16(5): 529-539.
- [15] Purwitasari, D., & Okazaki, Y., & Watanabe, K. (2007). A Study on Web Resources' Navigation for e-Learning: Usage of Fourier Domain Scoring on Web Pages Ranking Method. *In ICICIC '07: Proc of the Second Intl. Conf. on Innovative Computing, Information and Control.*
- [16] Geng, L., & Hamilton, H.J. (2006). Interestingness Measures for Data Mining: A Survey. *ACM Comput. Surv.*, 38(3).
- [17] Karypis, G. (2002). Multilevel Hypergraph Partitioning. *Technical Report 02-25*, Comput. Sci. and Eng. Dept., Univ. Minnesota, Minneapolis.