Refining the Results of Automatic e-Textbook Construction by Clustering

Jing Chen[1], Qing Li[1], and Ling Feng[2]

[1] Department of Computer Engineering and Information Technology,
City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong
{jerryjin, itqli}@cityu.edu.hk
[2] Department of Computer Science,
University of Twente, PO Box 217, 7500 Enschede, The Netherlands
ling@ cs.utwente.nl

Abstract. The abundance of knowledge-rich information on the World Wide Web makes compiling an online e-textbook both possible and necessary. The authors of [7] proposed an approach to automatically generate an e-textbook by mining the ranking lists of the search engine. However, the performance of the approach was degraded by Web pages that were relevant but not actually discussing the desired concept. In this paper, we extend the work in [7] by applying a clustering approach before the mining process. The clustering approach serves as a post-processing stage to the original results retrieved by the search engine, and aims to reach an optimum state in which all Web pages assigned to a concept are discussing that exact concept.

1 Introduction

The World Wide Web has evolved into one of the largest information repositories. It now becomes feasible for a learner to access both professional and amateurish information about any interested subject. Professional information often includes compiled online dictionaries and glossaries, course syllabus provided by teachers, tutorials of scientific software, overview of research areas by faculties from research institutes, etc. Discussion boards sometimes offer intuitive description of the interested subjects, beneficial for students or beginning learners. All these resources greatly enrich and supplement the existing printed learning material. The abundance of knowledge-rich information makes compiling an online e-textbook both possible and necessary.

The most common way of learning through the Web is by resorting to a search engine to find relevant information. However, search engines are designed to meet the most general requirements for a regular user of the Web information. Use Google ([1]) as an example. The relevance of a Web page is determined by a mixture of the popularity of the page and textual match between the query and the document ([2]). Despite its worldwide success, the combined ranking strategy still has to face several problems, such as ambiguous terms and spamming. In the case of learning, it becomes even harder for the search engine to satisfy the need of finding instructional

R.W.H. Lau et al. (Eds.): ICWL 2005, LNCS 3583, pp. 311 – 319, 2005.

information, since the ranking strategy cannot take into account the needs of a particular user group, such as the learners.

1.1 Background of Research

Recently, many approaches have been proposed to improve the appearance of Web search engine results. A popular solution is clustering, providing users with a more structured means to browse through the search engine results. Clustering mainly aims at solving the ambiguous search terms problem. When the search engine is not able to determine what the user's true intention is, it returns all Web pages that seem relevant to the query. The retrieved results could cover widely different topics. For example, a query 'kingdom' actually referring to biological categories could result in thousands of pages related to the United Kingdom. Clustering these results by whole pages or their snippets is the most commonly used approach to address this problem ([3][4][5]). However, the structure of the hierarchy presented is usually determined on-the-fly. Cluster names and their organized structure are selected according to the content of the retrieved Web pages and the distribution of different topics within the results. The challenge here is how to select meaningful names and organize them into a sensible hierarchy. Vivisimo [6] is an existing real-life demonstration of this attempt.

The clustering approach works well to meet the needs of a regular user. But when the application is narrowed down to an educational learning assistant, it is possible to provide the learners with more 'suitable' Web pages that satisfy their needs in the pursuit of knowledge. Users seeking for educational resources prefer Web pages with a higher quality of content. Such Web pages often satisfy the criteria of being self-contained, descriptive and authoritative [7]. Limited work has been done to distinguish higher quality data from the Web. An important one is [8], where the authors attempt to mine concept definitions on the Web. They rely on an interactive way for the user to choose a topic and the system automatically discovers related salient concepts and descriptive Web pages, which they call informative pages. They not only proposed a practical system that successfully identified informative pages, but also more importantly pointed out a novel task of compiling a book on the Web.

In [7], the authors proposed an approach to automatically construct an e-textbook on the Web. They extend Liu et al.'s work by adding a concept hierarchy that outlines the user-specified topic. In the concept hierarchy, also called a concept tree, each node corresponds to a concept and the ancestor relationship of nodes represents the containing relation of the concepts. The use of the concept tree is essential and benefits the learning experience to a great extent. The concept tree is used to gather Web pages that are more likely to be of learning importance. It also readily serves as a table-of-content for the final e-textbook. It is easier for the users to understand compared with the cluster hierarchy generated on-the-fly, thus saves time for browsing. The approach is described concisely in the following:

1. Dataset collection: The concepts in the concept tree are used to generate a query phrase for each node. The query terms indicate the relationship of concepts. Web pages that cover more concepts in the query are more likely to be ranked high in the list.

2. Mining process: "Suitable" pages from the retrieved list of each concept tree node are mined and re-ranked according to a combined ranking strategy.
3. Expansion: For some nodes that do not have sufficient "suitable" pages, an expansion process is activated.
4. Result presentation: Remaining Web pages are presented to users with the concept tree in the left area of the screen, serving as a navigation guide.

In the approach, the mining process is performed on the retrieved result of the search engine. However, the ranking strategy of the search engine cannot guarantee that the main theme of a highly ranked Web page is actually about the query. Often, a Web page describing an ancestor or offspring concept is ranked high in the list. For instance, for a query "infinite series", a Web page actually discussing a sub-topic "geometric series" is ranked high in the list. The phrase "infinite series" appears several times in the Web page, since "geometric series" is a sub-topic of the broader "infinite series". The search engine only notices to what extent this page is related to the search term, but cannot determine the main theme of the page. It should not be blamed for such a relevance measure, but in our scenario it is better that the page about "geometric series" is considered a candidate page for the node "geometric series" rather than for "infinite series". The algorithm proposed in [7] tries to stress on the search terms by giving higher priority to them, but is too simple and not sufficient to successfully identify a Web page's main theme. So the quality of the mining process is affected by these "noises" that could have been "hits" for other concept tree nodes.

1.2 Paper Contribution and Organization

In this paper, we add a clustering procedure before the mining process to adjust the distribution of the Web pages in the concept tree. The performance of the mining process is improved because Web pages are associated with the appropriate concept tree nodes in the adjusted Web page collection. In our approach, we treat the retrieved results of all nodes in the concept tree as the initial clustering condition, and perform a clustering procedure upon it to optimize the distribution of the documents in the collection. In order to make the clustering process suitable for such an application, we propose a new Web page representation model, which projects a Web page onto the concept tree. The projection is called an instance tree. The new Web page representation model can well describe the distribution and the relationship of the concepts appearing in a Web page, and consequently, characterize its main theme precisely. It also reduces the dimension of the representation and improves the efficiency of the clustering process. Then the corresponding tree distance measure is defined to evaluate the distance between two instance trees. When the clustering process terminates and the optimum status is reached, Web pages are assigned to the appropriate concept tree nodes that match with their main themes.

In the rest of the paper, we first define the new Web page representation model, along with the similarity metric used to measure the distance between two instances of the model. In Section 3, we discuss how the clustering algorithm is applied with Web page representation model. Section 4 gives a case study of our approach. And we conclude our paper in Section 5.

2 Web Page Representation Model

The most popular document representation model in modern information retrieval is the vector space model (VSM [9]). A document is considered as a set of terms and represented as a high-dimensional vector where a term stands for a dimension. A non-binary weight is assigned to each term in the term space. Based on the vector space model, the similarity of Web pages can be measured through computing the cosine distance between the two vectors.

But the vector space model is not very suitable in our scenario. Web pages associated with "close" concept tree nodes are sometimes similar with each other in their distribution of terms, even though they are not describing the exact same concepts. The previous example of "infinite series" and "geometric series" explains why their features can overlap. Thus the similarity between two "close" Web pages cannot be evaluated precisely. In our case, preciseness is required. We must identify the main theme a Web page is describing, at the presence of other "close" concepts in the concept tree. The Web page representation model must be able to record the information of the relationship of the concepts contained in the Web page along with the concept distribution.

2.1 Instance Trees

The central idea of our approach is about a user-specified concept hierarchy, which we call the concept tree. The concept tree should provide a hierarchical outline of the concerned topic, where nodes on the upper part of the tree represent more general concepts and those in lower positions stand for more specific topics. A concept tree is a labeled ordered rooted tree. A rooted tree is a tree with a vertex singled out as the 'root'. The root node of our concept tree represents the main topic the user is interested in. We consider the concept tree as ordered mainly for clarity in description. The concept tree is defined as follows:

Definition 1. Let CT denote a concept tree.

- |CT| represents the number of nodes in the concept tree CT;
- V_{CT} denotes the set of vertexes of CT;
- $V_{CT}[i]$ is the i-ism vertex of CT in a preorder walk of the tree;
- E[i, j] is the edge between two adjacent vertexes $V_{CT}[i]$ and $V_{CT}[j]$. The direction of the edge is ignored in our definition;
- C(x) stands for the corresponding concept to vertex x;
- EdgeDist (x,y) denotes the edge distance between vertex x and y, which is calculated by the minimum number of edges between x and y.

An example of a concept tree is displayed in Figure 1. The edge distance between vertex $V_{CT}[4]$ and vertex $V_{CT}[7]$ is 3 according to the definition of edge distance, where the corresponding edges are E[4, 3], E[3, 1], E[1, 7].

By mapping a Web page onto a concept tree, it is possible to analyze the relationship of the concepts appearing in a Web page, thus further determining the main theme of the document. In our approach, each Web page is represented as a tree structure identical to the concept tree, called an *instance tree*.

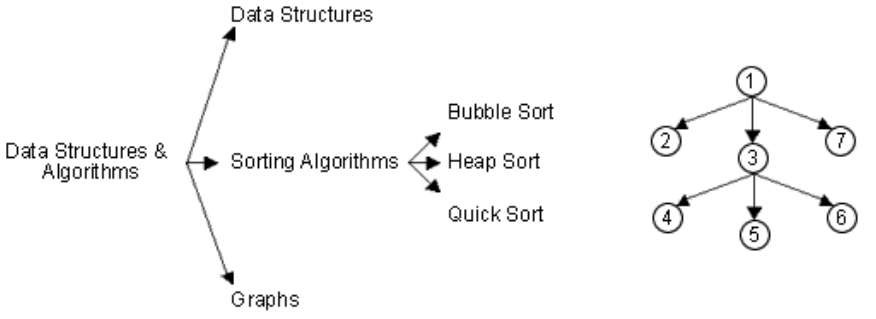

Fig. 1. Example of a concept tree

Definition 2. Let T_x be an instance tree from a Web page W_x to the concept tree CT. $|T_x|$ denotes the number of nodes in T_x. Let V_{Tx} be the set of vertexes in T_x , and $V_{Tx}[i]$ be the i-ism vertex in a preorder walk of T_x. Let C(x) be the corresponding concept to vertex x in T_x. Function $\phi: V_{Tx} \rightarrow V_{CT}$ is a projection from the vertexes in an instance tree T_x to vertexes of the concept tree CT such that the i-ism vertex in T_x is mapped to the i-ism vertex in CT in preorder. The value of a vertex $val(V_{Tx}[i])$ is denoted as the number of occurrences of the concept $C(V_{CT}[i])$ in Web page W_x.

The following conditions are held for instance tree and the concept tree:

- $|T_x|=|CT|$;
- $\phi(V_{Tx}[i]) = V_{CT}[i]$ for any $1 \leqq i \leqq |T_x|$;
- $C(V_{Tx}[i]) = C(V_{CT}[i])$ for any $1 \leqq i \leqq |T_x|$.

Figure 2 depicts three different instance trees corresponding to the concept tree in Figure 1. The numbers on the right side of each concept tree node stands for the value of that node.

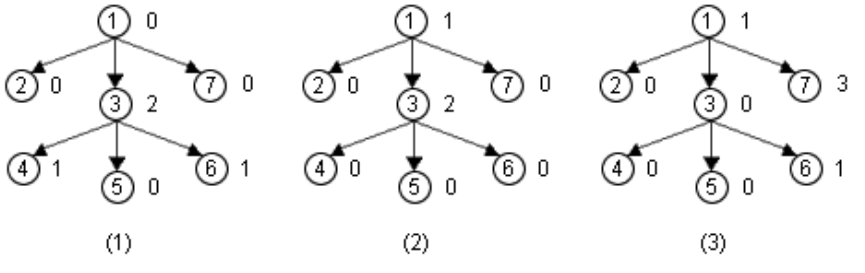

Fig. 2. Three different instance trees

2.2 Distance Measure

In k-means clustering and many other clustering approaches, it is necessary to calculate a "distance" between two objects or an object and a cluster centroid. In our approach, an object is an instance tree. A popular means to compare the difference

between two trees is the edit distance. This method tries to convert one tree into the other, and analyzes the distance by counting the number of steps needed for the transformation. [10] gives a general description about edit distance, and the measure is widely used in many tree comparing tasks ([11][12][14]). However, edit distance is mainly for evaluating structural similarity in two different tree structures, while the instance trees are all of the same structure. In addition, the instance tree not only reflects the distribution of the concepts in a Web page, but also records the relationship of the appearing concepts. The relational information is important and should not be ignored.

To take the relational information into account, given a vertex x in an instance tree T_1, we are first interested in its "closest" vertex y in T_2. The "closeness" is measured by the EdgeDist(ϕ(x), ϕ(y)) defined above. Such a distance is called the distance between the vertex x in T_1 and the instance tree T_2. We also define the following function σ which indicates whether a concept has occurred in a Web page:

$$\sigma(x) = \begin{cases} 1, & if\ val(x) > 0; \\ 0, & elsewise \end{cases}$$

Definition 3. Let x be a vertex in a tree distance T_1, the distance between x and instance tree T_2 is defined as:

$$dist(x, T_2) = \sigma(x) \times \left(min\left\{ EdgeDist(\phi(x), \phi(y)) \mid \sigma(y) = 1, y \in T_2 \right\} + 1 \right)$$

The σ(x) in the equation above guarantees that the distance makes sense only when the value of vertex x is not zero.

Given the distance of a vertex in T_1 and another instance tree T_2, the distance between two instance trees T_1 and T_2 can then be defined:

Definition 4. The distance between two instance trees T_1 and T_2 is:

$$treedist(T_1, T_2) = \frac{\sum_{i=1}^{|T_1|} | val(V_{T_1}[i]) - val(V_{T_2}[i]) | \times dist(V_{T_1}[i], T_2) + \sum_{j=1}^{|T_2|} | val(V_{T_2}[j]) - val(V_{T_1}[j]) | \times dist(V_{T_2}[j], T_1)}{\sum_{i=1}^{|T_1|} \sigma(V_{T_1}[i]) + \sum_{j=1}^{|T_2|} \sigma(V_{T_2}[j])}$$

It can be easily proved that for any two instance trees T_1 and T_2, the instance tree distance satisfies the following constraints:

- treedist(T_1, T_1)=0;
- treedist(T_1, T_2)≧0;
- treedist(T_1, T_2)= treedist(T_2, T_1).

However, the instance tree distance is not normalized. In the example of the three instance trees in Figure 2, the following can be easily calculated:

$$treedist(T_1, T_2) = \frac{((2-2)\times 1 + (1-1)\times 2 + (1-0)\times 2) + ((1-0)\times 2 + (2-2)\times 1)}{3+2} = \frac{6}{5}$$

$$treedist(T_1, T_3) = \frac{((2-0)\times 2 + (1-0)\times 3 + (1-1)\times 1) + ((1-0)\times 2 + (1-1)\times 1 + (3-0)\times 3)}{3+3} = \frac{5}{2}$$

$$treedist(T_2, T_3) = \frac{((1-1)\times 1 + (2-0)\times 2) + ((1-1)\times 1 + (3-0)\times 2 + (1-0)\times 2)}{2+3} = \frac{12}{5}$$

3 Clustering Process

K-means clustering is a well-known member of the family of clustering algorithms ([2]). The user first defines a preset number k of clusters. Initially, the objects can be arbitrarily divided into k clusters. Each cluster is represented as the centroid of the documents within it. Thereafter, an iterative process begins by assigning objects to the closest cluster. A detailed implementation of the k-means clustering algorithm can be found in [13]. This approach is especially useful when the k clusters are already formed by some other algorithm. For the ranking lists provided by the search engine, Web pages are naturally clustered to the concept tree node used to generate the queries. The k-means clustering algorithm can then be applied as a postprocessing stage to move the misplaced points to the appropriate cluster.

The centroid of a cluster of instance trees is represented by a tree structure similar as the concept tree and the instance trees. The value of its vertexes is defined as follows:

Definition 5. Let C_i denote the centroid of the cluster corresponding to the concept tree node $V_{CT}[i]$. N is the number of instance trees that belong to cluster C_i. The value of the i-ism vertex in C_i is calculated as:

$$Val(V_{C_i}[k]) = \frac{\sum_{T_j \in C_i} val(V_{Tj}[k])}{N}$$

A distortion metric is minimized during the clustering process. We choose to minimize the total distance between all objects and their centroids for simplicity. The minimal distortion and the instance tree distance together determine the shape of the optimum clusters.

4 Case Study

In this section, we provide a case study of how our algorithm works. The following concept tree (Figure 3) is used to generate queries and obtain the corresponding Web page set:

The following snippets in Figure 4 are provided by the search engine for the concept tree nodes “data mining” and “association rules” respectively. (a) is an abstract of a paper about mining association rules. It was mistakenly retrieved for the concept tree node “data mining” because the term “data mining” appeared several times in the Web page. (b) is someone’s publication list, mainly in the area of data mining. Besides his contribution in “association rules”, the author had many other publications in classification, clustering, etc.

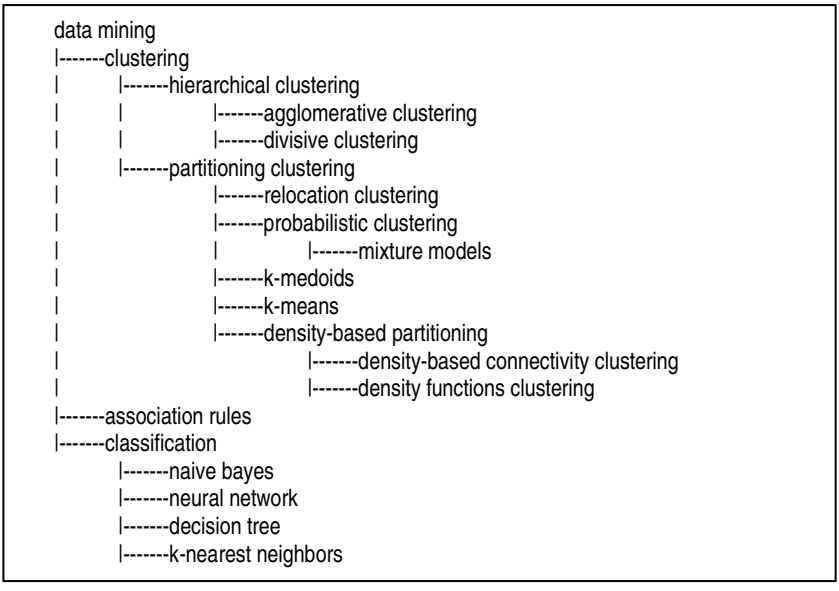

Fig. 3. Concept Tree for "data mining"

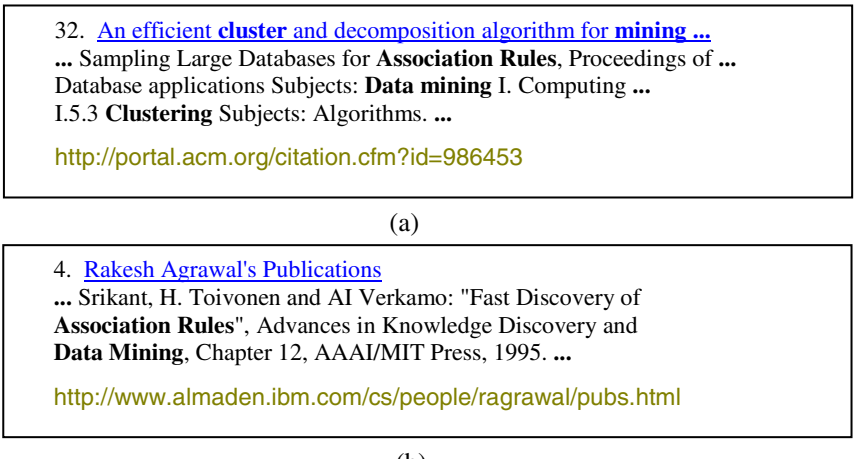

Fig. 4. Snippets from the SE results. (a) 32nd result for node "data mining"; (b) 4th result for node "association rules"

In our approach, (a) was moved to the concept tree node "association rules", while (b) was assigned to the correct ancestor concept tree node "data mining". The movement from a node to an ancestor or an offspring is the most common action taken by our algorithm.

5 Conclusion

In this paper, we target at improving the results of an automatically generated online e-textbook. We propose a new Web page representation model, which we call the instance tree, to highlight the relationship of concepts contained in the Web page as well as their numerical appearances. A clustering algorithm is introduced to cluster the instance trees, and obtain an optimum state where all Web pages are assigned to their appropriate concept tree node. In the future, we will carry on more extensive experiments to evaluate our proposed clustering algorithm thoroughly.

References

[1] S. Brin, L. Page, "The Anatomy of a Large-scale Hypertextual Web Search Engine", in Proceedings of International Conference on World Wide Web, 1998.
[2] S. Chakrabarti, Mining the Web: Discovering Knowledge from Hypertext Data, Morgan Kaufmann Publishers, 2002.
[3] Oren Zamir, Oren Etzioni: Grouper: A Dynamic Clustering Interface to Web Search Results. Computer Networks 31(11-16): 1361-1374, 1999.
[4] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, and Wei-Ying Ma. Learning To Cluster Web Search Results. In Proceedings of the 27th annual international conference on research and development in information retrieval (SIGIR'04), pp. 210-217, Sheffield, United Kingdom, July 2004.
[5] Paolo Ferragina, Antonio Gullí: The Anatomy of a Hierarchical Clustering Engine for Web-page, News and Book Snippets. ICDM 2004: 395-398, 2004.
[6] Vivisimo, http://vivisimo.com/html/index
[7] Jing Chen, Qing Li, Liping Wang, Weijia Jia: Automatically Generating an e-Textbook on the Web. ICWL 2004: 35-42, 2004.
[8] B. Liu, C-W. Chin, H-T. Ng, "Mining Topic-specific Concepts and Definitions on the Web", in Proceedings of International Conference on World Wide Web, 2003, pp. 251-260, 2003.
[9] Salton. G. and McGi11, MJ., Introduction to Modern Information Retrieval McGraw Hill, New York, 1983.
[10] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. SIAM Journal of Computing, 18(6):1245-1262, 1989.
[11] Yuan Wang, David J. DeWitt, Jin-yi Cai: X-Diff: An Effective Change Detection Algorithm for XML Documents. ICDE 2003: 519-530
[12] Andrew Nierman, H. V. Jagadish: Evaluating Structural Similarity in XML Documents. WebDB 2002: 61-66
[13] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, Angela Y. Wu: An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE Transaction on Pattern Analysis and Machine Intelligence, 24(7): 881-892, 2002.
[14] Davi de Castro Reis, Paulo Braz Golgher, Altigran Soares da Silva, Alberto H. F. Laender: Automatic web news extraction using tree edit distance. WWW 2004: 502-511