5-2002

Personalized information management for web intelligence

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Personalized Information Management for Web Intelligence

Ah-Hwee Tan

Kent Ridge Digital Labs

21 Heng Mui Keng Terrace, Singapore 119613

Email: ahhwee@krdl.org.sg

Abstract— **Web intelligence can be defined as the process of scanning and tracking information on the World Wide Web so as to gain competitive advantages. This paper describes a system known as Flexible Organizer for Competitive Intelligence (FOCI) that transforms raw URLs returned by internet search engines into personalized information portfolios. FOCI builds information portfolios by gathering and organizing on-line information according to a user's needs and preferences. Through a novel method called User-Configurable Clustering, a user can personalize his/her portfolios in terms of the content as well as the information structure. The personalized portfolios can then be used to track new information and organize them into appropriate folders accordingly. We show a sample session of FOCI which illustrates how a user may create and personalize an information portfolio according to his/her preferences and how the system discovers novel information groupings while organizing familiar information according to the user-defined themes.**

I. Introduction

Competitive intelligence (CI) refers to the process of scanning, tracking, and analyzing a company's external environment so as to gain competitive advantages. It includes information of competitors, suppliers, consumers, and other factors that may help a manager in strategic decision making. In the knowledge-based era, it is widely acknowledged that it is increasingly risky to do business without competitive intelligence. As a result, the software industry has invested increasing R&D resources to produce business or competitive intelligence products. Survey.com has estimated that sales of business intelligence and competitive intelligence software will reach $148 billion by 2003.

Web intelligence, in our context, refers to the process of scanning, tracking, and analyzing competitive information on the World Wide Web. According to Fuld & Company [3], an intelligence cycle consists of 1) Planning and Direction, 2) Published Information Collection, 3) Human Intelligence Collection, 4) Analysis and Production, and 5) Report and Inform. CI professionals have referred to published information collection as secondary sources and human intelligence collection as primary sources. We however believe that the importance of published information collection increases each day in the digital age as an explosive amount of new information continues to become available online. The challenge upon us is how to tap and organize such a massive amount of diverse information into actionable knowledge and reports.

Popular internet search engines, such as Yahoo!, Excite, AltaVista, and Lycos, retrieve documents upon users' search queries but do not organize the search results. More sophisticated tools such as Copernics, BullsEye, and NorthernLight organize search results into automatically generated folders to facilitate navigation and browsing. However, as in typical clustering systems [1], [5], [6], users have very little control on how the information are organized and the information clusters generated may not match the users' requirement. As a consequence, internet search tools are used mainly for gathering purpose only. Serious users, such as intelligence scouts, still have to manually compile the materials according to their needs and preferences. This can be a painstaking process, especially when the information needs to be updated frequently.

This paper introduces a system called FOCI (for Flexible Organizer for Competitive Intelligence) to assist knowledge workers to perform competitive intelligence on the web. FOCI bridges the gap between raw search results and personalized competitive information portfolios by providing an integrated platform that supports the key activities in a competitive intelligence cycle. FOCI constructs information portfolios by gathering and organizing on-line information into automatically generated folders. A user, upon inspecting the information groupings, can then modify the structure according to his/her requirement and preferences through a suite of cluster manipulation functions. It is an interactive process of clustering, personalization, and discovery through which a user turns an automatically generated cluster structure into his/her preferred organization. In FOCI, personalization is achieved through a method known as User-Configurable Clustering [9] that incorporates users' preferences in an information clustering system. The personalized portfolios can be constantly updated by tracking and organizing new information automatically. The portfolios thus function as "living reports" that can be published and shared by other users. In all, the system provides an environment for gathering, organizing, tracking, and publishing of competitive information on the web.

The rest of this article is organized as follows. Section 2 presents the FOCI architecture and a brief description of it's main components. Section 3 presents User-Configurable Clustering, a key enabling technology of FOCI. Section 4 illustrates how FOCI can be used in creating, organizing, and tracking an exemplary information portfolio. The final section summarizes and concludes.
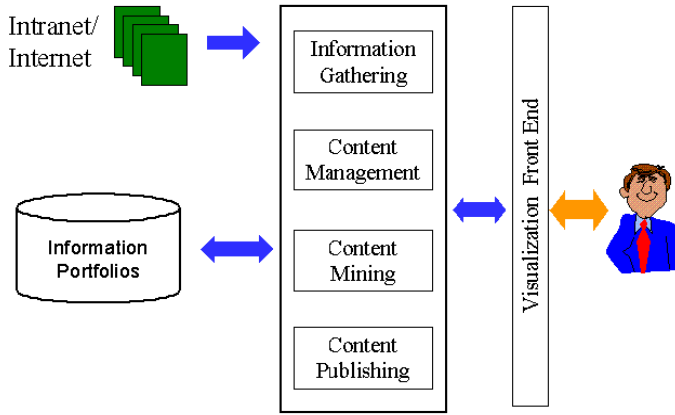
Fig. 1. The FOCI system architecture.

Fig. 2. A screen shot of FOCI.

II. FOCI System Architecture

Referring to Figure 1, FOCI comprises an information gathering module for retrieving and integrating online information from diverse sources, a content management module for organizing and personalizing portfolios, a content mining module for analyzing information portfolios, a content publishing module for sharing of portfolios, and a user interface module for graphical visualization and user interaction. We briefly describe the key functions supporting a competitive intelligence cycle below.

1. **Information gathering**: The information gathering module allows a user to build an information portfolio by sending search queries to major internet search engines and searchable news sites; and integrating the search results returned. The user can also insert his/her own links or documents not found in the search results into the portfolio directly. An automatic *tracking* function monitors a selected set of online sources and updates the portfolio with new content periodically.

2. **Content management**: The content management module provides a host of utilities for a user to organize and manage his/her competitive information in the preferred manner. Domain-specific template is also provided for organizing information into predefined section. In the domain of information technology for example, a recommended template organizes competitive information into *news*, *market*, *company*, *resources*, and *events*. Coupled with an automatic clustering engine, FOCI allows a set of personalization functions such as labelling, adding, deleting, grouping, and splitting of clusters. Additional functions include annotation and deletion of documents, clusters, and portfolios.

3. **Content mining**: The content mining module extracts key attributes from raw information content and transforms them into intuitive format for information discovery. Key analysis functions include topic detection/tracking, trend analysis[4], and link association. Due to the space constraint, content mining is outside the scope of this paper.

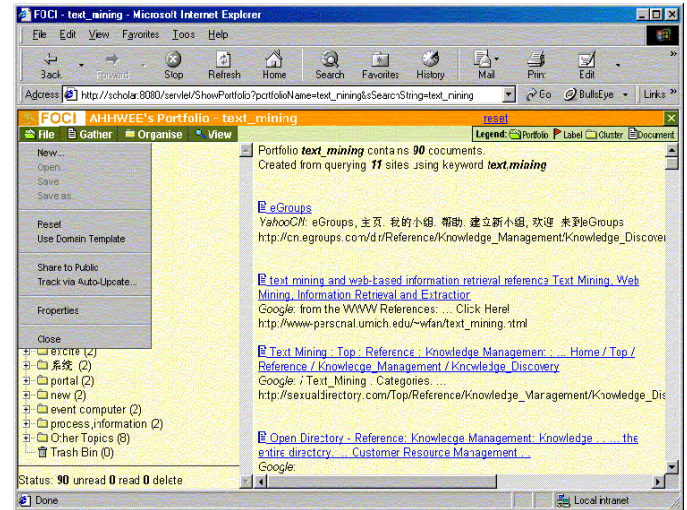4. **Content publishing**: The content publishing module handles the permission control of individual information portfolios so that a user can elect to release his/her portfolios for public access. Various views are also supported for presenting portfolios in different levels of details.

The FOCI server is running on UNIX SOLARIS workstations. A pre-alpha version of the system is available at http://textmining.krdl.org.sg/FOCI. As the user interface is based on Servlets and dynamic HTML, the application is accessible through Internet Explorer (IE) version 5.5 and above only. Figure 2 provides a screen shot of FOCI in action.

III. User-Configurable Clustering

A User-Configurable Clustering system (Figure 3) comprises an information clustering engine for clustering information based on similarities, a user interface module for displaying information groupings and obtaining user preferences, a personalization module for defining, labelling, and modifying cluster structure, and a knowledge base for storing user-defined cluster structure.

The personalization module works in conjunction with the information clustering engine to incorporate user preferences to modify the automatically generated cluster structure. By adopting a vector space model, we assume that each unit of information can be encoded by an information vector and that a user preference, indicating a preferred grouping of the information, can be encoded by a preference vector. Through the user interface module and the personalization module, a computer user is able to influence the organization of the information vectors (in the form of information groupings or clusters) by indicating his/her own preferences as preference vectors. Specifically, a user can perform a wide range of cluster personalization functions, including labelling, adding, deleting, merging, and splitting of information clusters. The customized cluster structure can be stored in the cluster structure knowledge base and retrieved at a later stage for processing new information. Based on the personalized cluster structure, new information can be organized according to the user's preferences captured over the previous sessions.
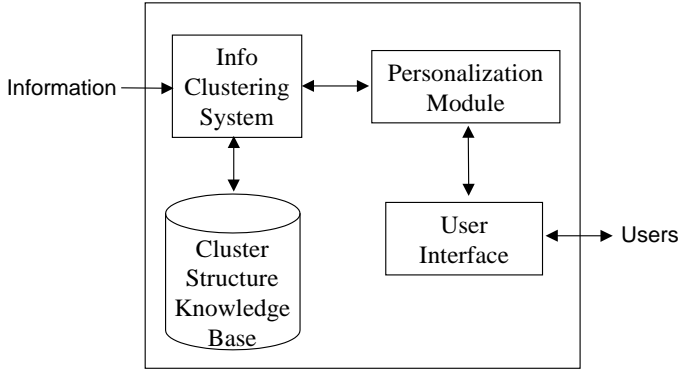
Fig. 3. A User-Configurable Clustering system.

Fig. 4. The Adaptive Resonance Associative Map architecture.

A. Clustering Engine

The information clustering engine presented in this paper belongs to a class of predictive self-organizing networks known as Adaptive Resonance Associative Map (ARAM) [7] that learns information groupings or clusters dynamically on-the-fly to encode pairs of information and preference vectors. ARAM is a natural extension of a family of unsupervised learning systems, namely Adaptive Resonance Theory (ART) networks, to incorporate supervisory preference signals. In addition, compared with traditional clustering engines, such as k-means and SOM, ARAM has the advantage of online incremental clustering, a critical requirement for supporting the interactive personalization and discovery process.

An ARAM system can be visualized as two overlapping Adaptive Resonance Theory (ART) [1] modules consisting of two input fields F_1^a and F_1^b with an F_2 category field (Figure 4). The ART modules used in ARAM can be ART 1 [1], which categorizes binary patterns, or analog ART modules such as ART 2, ART 2-A, and fuzzy ART [2] which categorize both binary and analog patterns. Fuzzy ARAM [7], [8] that is based on fuzzy ART is used in FOCI.

For User-Configurable Clustering, the F_1^a field contains the activities of the information vectors and the F_1^b field contains the activities of the preference vectors. Information clusters (represented at F_2) are created during *learning* through the synchronized clustering of the information and preference vectors. Specifically, each recognition node or cluster j learns to encode a pair of template information vector $\mathbf{w}_j^a$ and template preference vector $\mathbf{w}_j^b$.

B. Encoding Information Vectors

We have adopted a bag-of-word approach for representing text-based documents. To perform real-time content aggregation and clustering, we estimate the content of the pages based on the information provided on the search result pages returned by the search engines (instead of loading the original documents). In addition to the keywords contained in the titles and descriptions of links, we also make use of the URL addresses which provide meta information of the web pages.
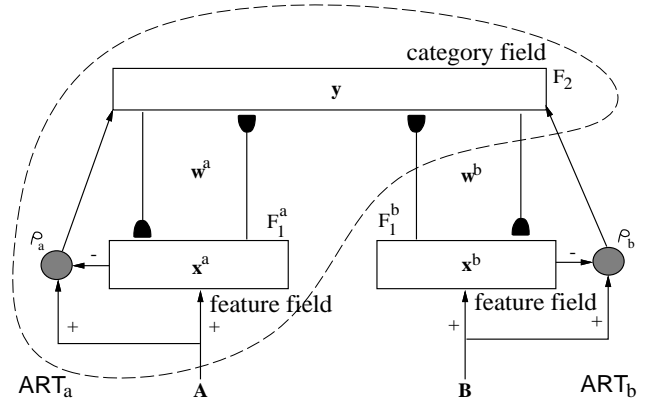
For a document d, we derive its information vector

$$\mathbf{A} = (a_1, a_2, \ldots, a_M) \tag{1}$$

such that

$$a_i = tf(w_i) * r(w_i)(1 - r(w_i)) \tag{2}$$

where M is the total number of keyword features selected from a document set after filtering stopwords and function words; the term frequency $tf(w_i)$ is the number of times the keyword w_i appears in document d; and the document ratio $r(w_i)$ is computed by

$$r(w_i) = \frac{df(w_i)}{D} \tag{3}$$

where the document frequency $df(w_i)$ denotes the number of documents that w_i appears in; and D is the number of the documents in the collection. The above term weighting scheme gives an advantage to keywords appearing in approximately half of the documents in the collection so as to encourage a more compact cluster structure.

The information vector is then normalized such that

$$a_i = \frac{a_i}{a_m} \qquad \text{where } a_m \geq a_i \text{ for all i.} \tag{4}$$

For user-defined terms (specified through adding clusters), the feature values are furthered enhanced by

$$a_i = \begin{cases} 1 & \text{if } a_i > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

C. Encoding Preference Vectors

User preferences, in this context, are in the form of themes or labels assigned by a user to individual documents or clusters. All labels specified by the users are stored in a *Label Table*. For a label l, we encode a preference vector $\mathbf{B} = (b_1, b_2, \ldots, b_N)$ such that

$$b_i = \begin{cases} 1 & \text{if } w_i = l \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

where w_i is the ith entry and N is the number of labels in the table. If a user-specified label cannot be found in the table, the label is added to the table and the dimension of the preference vectors (N) is updated accordingly.

Given a set of information items,
if a predefined cluster structure exists, load ARAM network $\mathcal{N}$;
else initialize ARAM network $\mathcal{N}$.
Loop
   For each information item $\mathcal{I}$,
    1. Derive an information vector $\mathbf{A}$ based on $\mathcal{I}$.
    2. Derive a null preference vector $\mathbf{E}$.
    3. Present $(\mathbf{A}, \mathbf{E})$ to $\mathcal{N}$ for learning.
    4. Record index of cluster J encoding $\mathcal{I}$.
until $\mathcal{N}$ is stable.

D. Clustering

The algorithm for information clustering using ARAM is summarized in Table I. If a predefined cluster structure exists, the system loads the ARAM network before clustering. Otherwise, a new network is created which contains zero cluster. During clustering, for each unit of information ($\mathcal{I}$), a pair of vectors $(\mathbf{A}, \mathbf{E})$ is presented to the ARAM network, where $\mathbf{A}$ is the information vector of $\mathcal{I}$ and $\mathbf{E}$ is a null vector such that $E_i = 0$ for $i = 1, \ldots, N$.

Given an information vector $\mathbf{A}$, the system first searches for a F_2 cluster J encoding a template information vector $\mathbf{w}_J^a$ that is closest to the information vector $\mathbf{A}$ according to a choice function. It then checks if the associated F_2 template information vector $\mathbf{w}_J^a$ of the selected category matches with the information vector according to a *match* criterion. If so, the template information vector of the F_2 cluster J is modified to encode the input information. Otherwise, the cluster is reset and the system repeats to select another cluster until a match is found or a new cluster is created.

Clustering is completed when ARAM is stable, in the sense that for a given set of information vectors, no new cluster is created and the changes in template vectors are below a specific threshold. With a predefined cluster structure, fuzzy ARAM organizes the information according to the cluster structure. Without predefined network structure, ARAM reduces to a pure clustering system that self-organizes the information based on the similarities among the information vectors only. The coarseness of the information groupings is controlled by the ART_a vigilance parameter (ρ_a).

E. Personalization

ARAM can also operate in an *insertion* mode whereby a pair of information and preference vectors can be inserted directly into an ARAM network. Whereas the *learning* mode is used for clustering and obtaining the cluster assignments of information vectors, the *insertion* mode enables a computer user to influence the clusters created by ARAM through indicating his/her own preferences in the forms of preference vectors. During *insertion*, the vigilance parameters ρ_a and ρ_b are each set to 1 to ensure that user preferences are explicitly encoded in the cluster structure. Under most cases, ARAM will create a new F_2 cluster node

to encode the input information and preference vectors inserted. In the event when the input information vector is identical to the template information vector of an existing cluster and there is a mismatch between the template preference vector and the input preference vector, we force the template preference vector to equal the input preference vector. This is appropriate for the purpose of personalization to favor preferences given directly by the user. We present the algorithms for performing the various cluster personalization functions below.

E.1 Labelling Information Clusters

Associating clusters with labels or themes allows a user to "mark" specific information groupings that are of interest to the user so that the information can be found readily in the future. More importantly, new information can be organized in the future according to such information groupings. To associate a cluster J with a label L, we simply insert $(\mathbf{w}_J^a, \mathbf{B})$ into ARAM, where $\mathbf{B}$ is a preference vector representing L. Labels reflect the user's interpretation of the groupings. They are useful landmarks to the user in navigating and locating old as well as new information.

E.2 Adding Information Clusters

A user can define and insert his/her own clusters into an ARAM network so that the information can be organized according to such information groupings. The inserted clusters reflect the user's preferred way of grouping information and are used as the default slots of organizing information.

To insert a new cluster, a pair of information and preference vectors $(\mathbf{A}, \mathbf{B})$ are first derived based on the key attributes of the information to be in the new cluster and the cluster label. After insertion, ARAM re-generates the cluster structures by clustering all the information vectors again. With the addition of user-defined clusters, new clusters may be generated during the re-clustering process.

E.3 Deleting Information Clusters

A user can delete a cluster by associating it with a *Trash* label. Information in a deleted cluster can then be handled separately and hidden from the user. In addition, a *Trash* cluster serves as a filter for removing unwanted information that is similar in content in the future.

To delete a cluster J, a pair of template information and preference vectors $(\mathbf{w}_J^a, \mathbf{T})$ is inserted into the cluster structure, where $\mathbf{w}_J^a$ is the template information vector of the cluster and $\mathbf{T}$ is derived based on the *Trash* label.

E.4 Merging Information Clusters.

Merging of clusters allows a user to combine two or more information groupings generated by clustering into a common theme. To merge clusters $J_1, \ldots, J_n$, the algorithm first derives a preference vector $\mathbf{B}$ encoding the user-specified label L. The vector pairs $(\mathbf{w}_{J_1}^a, \mathbf{B}), \ldots, (\mathbf{w}_{J_n}^a, \mathbf{B})$ are then inserted into ARAM one at a time so that the

template preference vectors of the clusters are modified to encode the common theme.

E.5 Splitting Information Clusters

Splitting of clusters allows a user to reorganize an information group, that he/she deems containing diverse content, into smaller clusters of specific themes. To split a cluster J, a user needs to select a number of information items $\mathcal{I}_1, \ldots, \mathcal{I}_n$ from the cluster as the pivots. The algorithm then derives information and preference vector pairs, namely $(\mathbf{A}_1, \mathbf{B}_1), \ldots, (\mathbf{A}_n, \mathbf{B}_n)$, where $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are the information vectors of $\mathcal{I}_1, \ldots, \mathcal{I}_n$ respectively; and $\mathbf{B}_1, \ldots, \mathbf{B}_n$ are the preference vectors derived from the cluster labels $L_1, \ldots, L_n$ respectively. After inserting these vector pairs into the ARAM network, each individual information vector, originally in the cluster J, will be re-organized into one of the smaller clusters depending on its similarities to $\mathbf{A}_1, \ldots, \mathbf{A}_n$.

IV. Experiments

In this section, we illustrate how FOCI can be used to create, organize, and track specific topics of interests indicated by a user. First, we show how the personalization functions can be used to create a personalized information portfolio. We then demonstrate how the personalized portfolio can serve as a template for tracking and organizing new information.

A. Clustering

An information portfolio on "text mining" was created by integrating search results of four internet search engines. For illustration purpose, we constrained the size of the portfolio by selecting only top 25 hits from each search engine. After removing duplicated links, there were 71 hits. Figure 5 depicts the clustering results based on a combination of URL-based and content-based keyword features. There are 17 clusters, each characterized by one to three keywords (read from its corresponding template information vector) listed in decreasing order of importance. Three clusters, namely *fortune*, *data*, and *information* [1] are the most prominent ones with 20, 17, and 7 documents respectively.

B. Personalization

Based on the raw cluster structure generated, this section illustrates how a user may use the various cluster manipulation functions, namely labelling, inserting, merging, and splitting, to personalize his/her portfolios.

Figure 6 shows a partially personalized portfolio. The *fortune* cluster containing news articles from the Fortune news site has been labelled under the theme of *Fortune News*. In addition, a number of user-defined cluster have been created under the theme of *Technology*. The documents in these user-defined clusters are mainly from the original *data*, *information*, and *knowledge* clusters in figure 5. In addition, a user-defined cluster with a keyword

[1] For convenience, we refer to a cluster by the first keyword in its keyword list.
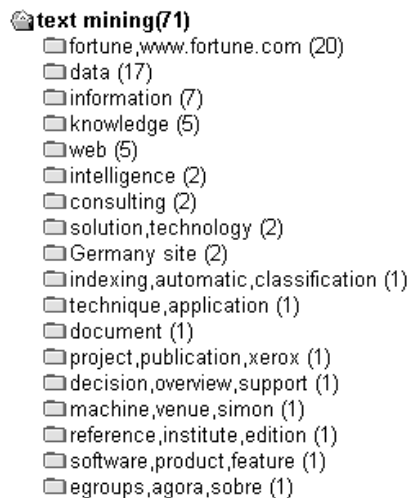

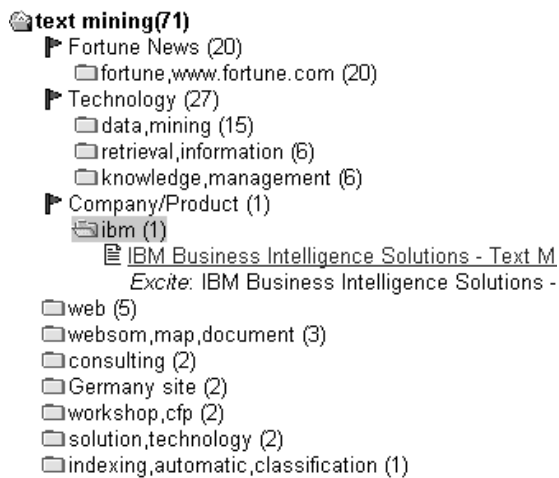Fig. 5. Clusters created by FOCI based on the 71 documents collected through the four internet search engines.

Fig. 6. A partially personalized portfolio on text mining.

IBM under *Company/Product* manages to pull out a link to a IBM Business Intelligence/Text Mining page which was buried somewhere previously. With these user-defined clusters, new clusters have emerged. A (previously hidden) grouping discovered is the *websom* cluster containing three links to websom related information.

Figure 7 shows an exemplary fully personalized portfolio. A number of split and merge clusters operation have been performed to organize the clusters into five themes. This portfolio has used a combination of organizing schemes. While much of the information is grouped according to their sources and the nature of the content (such as News, Company/Products, Research, and Events), there is a horizontal grouping on *Technology* that organizes information according to the various subfields and related topics in text mining, such as knowledge management, data mining, and information retrieval.

C. Tracking

In this section, we show the benefits of tracking and clustering new information using the personalized portfolio. A
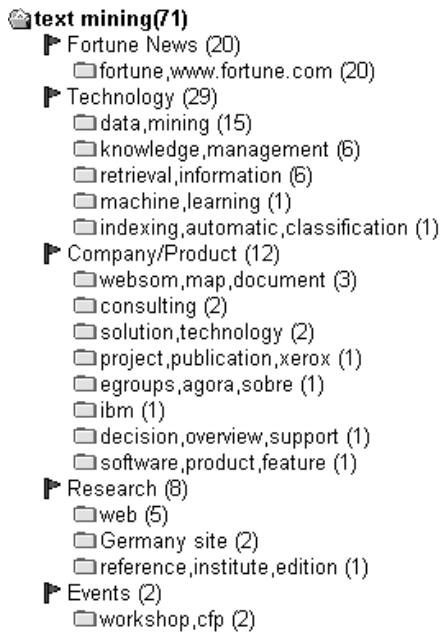
text mining(71)
  Fortune News (20)
    fortune,www.fortune.com (20)
  Technology (29)
    data,mining (15)
    knowledge,management (6)
    retrieval,information (6)
    machine,learning (1)
    indexing,automatic,classification (1)
  Company/Product (12)
    websom,map,document (3)
    consulting (2)
    solution,technology (2)
    project,publication,xerox (1)
    egroups,agora,sobre (1)
    ibm (1)
    decision,overview,support (1)
    software,product,feature (1)
  Research (8)
    web (5)
    Germany site (2)
    reference,institute,edition (1)
  Events (2)
    workshop,cfp (2)

Fig. 7. A personalized portfolio on text mining. All information have been organized into one of the five themes.

text mining new(42)
  businesswire,www.businesswire.com (24)
  search (4)
  software (4)
  information (3)
  knowledge (3)
  group (2)
  industry (2)

Fig. 8. Clusters created by FOCI based on the 42 new documents without personalization.

text mining(113)
  Fortune News (20)
  Technology (36)
    data,mining (17)
    knowledge,management (9)
    retrieval,information (8)
    machine,learning (1)
    indexing,automatic,classification (1)
  Company/Product (20)
    technology (4)
    software (6)
    websom,document,map (3)
    publication (2)
    egroups,agora,sobre (1)
    consulting (2)
    decision,support,content (1)
    ibm (1)
  Research (8)
  Events (2)
  businesswire,www.businesswire.com (24)
  group (2)
  solution,system,analysis (1)

Fig. 9. Organization of the new documents into the personalized portfolio.

new set of 42 documents was collected through three additional search engines. Without prior structure, the documents would be organized into the clusters as shown in figure 8. In contrast, figure 9 shows the clustering result when the new documents are organized based on the personalized cluster structure. There are 113 documents in the combined portfolio. A significant chunk of the new information, especially those in the *search*, *software*, *information*, and *knowledge* clusters (figure 8), have been organized under the themes of *technology* and *Company/Product*. Some of the other clusters prevail, highlighting information that do not fit into the personalized portfolio. The most prominent group is the *businesswire* cluster that contains news articles from the BusinessWire news site. This indicates that the system discovers novel information groupings while organizing familiar information into the user's personalized structure.

V. Conclusions

This paper has presented a system known as FOCI that performs competitive intelligence on the Web through personalized information management and tracking. Personalization in FOCI is achieved via a new information management technique called User-Configurable Clustering that integrates the complementary strengths of clustering and categorization. As illustrated in our experiments, FOCI supports an interactive process of automatic clustering, personalization, tracking, as well as discovery. As a user indicates his/her existing know-how and interpretation of the environment in terms of how he/she wants the information to be organized, any information that falls outside of the defined cluster structure is thus new and potentially interesting to the user. This enables the user to discover information that is novel with respect to prior experience.

References

[1] G. A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 1987.

[2] G. A. Carpenter, S. Grossberg, and D. B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[3] Fuld & Company. Intelligence Software Review 2000. *http://www.fuld.com/softwareguide/index.html*, 2000.

[4] R. Kanagasa and A-H. Tan. Topic detection, tracking and trend analysis using self-organizing neural networks. In *Proceedings, Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01), Hong Kong*, pages 102–107, 2001.

[5] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen. Creating an order in digital libraries with self-organizing maps. In *Proceedings, WCNN'96, San Diego*, 1996.

[6] T. Kohonen. *Self-organization and Associative Memory.* Springer-Verlag, 1988.

[7] A.-H. Tan. Adaptive Resonance Associative Map. *Neural Networks*, 8(3):437–446, 1995.

[8] A-H. Tan. Predictive self-organizing networks for text categorization. In *Proceedings, Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01), Hong Kong*, pages 66–77, 2001.

[9] A-H. Tan and H. Pan. Adding personality to information clustering. Submitted for publication.