Utilization of Keyword Map for Intelligent Web Interaction based on Retrieval, Browsing, Analysis Operations

Yasufumi Takama Tomoki Kajinami
Tokyo Metropolitan Institute of Technology

Abstract: Both Browsing and Retrieval with search engines are major operations that establish the interactions between users and the Web. Although both operations are usually combined for locating information from the Web, recent growth of the Web has overtaken the potential of this conventional interaction. This paper proposes the concept of Retrieval, Browsing, and Analysis (RBA)-based interaction, as the improvement of the conventional Retrieval and Browsing (RB)-based interaction. The prototype interface for RBA-based interaction is implemented, which employs keyword map visualization system. The relevance feedback based on keyword map is also proposed, and experimental results show that user’s intention can be extracted from the keyword map modified by the user.

Keywords: Web Intelligence, Information Visualization, Relevance Feedback, Keyword Map, Immune Network

1. Introduction

A Web information visualization system that employs RBA (Retrieval, Browsing, and Analysis)-based interaction is presented for assisting user’s Web interaction. A Web interaction is defined as users’ activities for viewing and collecting web pages with using search engines and Web browsers. There exists vast amount of information in the Web, from which users usually gather information without definite information needs. Therefore, it is difficult for users to organize and understand what they have gathered from the Web. In this paper, we propose the concept of RBA-based interaction. The Web information visualization system proposed in this paper employs both keyword map visualization and document clustering, which respectively present users the topic distribution and document clusters within gathered document set. Employing the immune network-based clustering algorithm, which has been already proposed, makes it possible to find relationship between document space and topical space (represented by keyword map). The relevance feedback based on keyword map is also proposed so that the interface can be more interactive.

2. Related Work

Browsing and Retrieval are the major operations that users perform on the Web. Browsing is typical operations in hyperspace (i.e., the Web). In the Web hyperspace, documents are linked to others by hyperlinks, and we can move from current document to others by clicking a hyperlink. On the other hand, we can also get a set of documents related with our information needs from search engines. This operation is called retrieval hereinafter.

It seems that the systems that support users’ browsing operations (browsing support system[1,2,6]) have been major approaches in early stage of web intelligence research. However, recent success of commercial search engines such as Google has let us shift from browsing to retrieval.

Although retrieval operation has potential for user-to-web interaction, current search engines have limitation of presenting results as only a list of documents. That is, getting retrieved results is just a starting point of interaction, and we have to make much effort for investigating individual pages. Therefore, browsing is still important, which is started with using the retrieved document as the seed for browsing. Of course, we often hit on a new query while browsing the retrieved results.

2.1. Browsing Support Systems

Browsing support systems assist users in selecting a link to follow within the current page. Typical browsing support systems, such as Letzia [6], syskili&Webert [1], Webwatcher [2] add the information to each link in a document, based on which a user can select the link that will lead to the popular page, or the page of interest. This kind of systems has been developed in early stage of the Web, in which most of links are static ones. According to the spread of dynamic Web and commercial search engines of huge volume, another type of support systems that visualize the partial Web hyperspace [3, 4, 17] have become popular. BookMap [4] visualizes the user’s personal hyperspace of bookmark and navigation history. It is based on the facts that (1) 92% of users have their own bookmark, and (2) more than 50% of page visits are page re-visits. It employs global fisheye and zooming operations, by which the system can show the detail of the part of hyperspace, while preserving the context (global structure).

Another example of browsing support systems is Comparative Web Browser (CWB), which is designed to assist users compare the contents of a site with another site [8]. The CWB uses two browsing displays for viewing the contents of two sites simultaneously. When a user reads a page of the
site on one of the displays, the corresponding page of another site is automatically displayed on another display.

2.2. Clustering-based Information Visualization Systems

Compared with above-mentioned browsing support systems that handle the hyperlinked structure of the Web, the systems that support the user's retrieval process handle a set of documents that contain the query terms. As most of documents in the set have no hyperlink to others within the set, they should be organized in other structure than hyperlinked structure. In particular, when a large number of documents are retrieved, they should be divided into closely related subsets [5, 19]. Scatter/Gather [5] and Grouper [19] employ document-clustering approach. Scatter/Gather applies the clustering method interactively, i.e., when a user select one of the generated document cluster, the selected one is further divided into several document clusters.

The clustering result is usually presented as a list, as most search engines do. Visualization technique can also be utilized for improving the user's accessibility to the generated document clusters. CardVis [7] handles the retrieved results as a graph, where vertices denote pages and edges denote the hyperlinks between these pages. As retrieved documents do not always form a single graph, several sub-graphs are generated. CardVis is based on the metaphor of a pack of playing cards, and each card shows a sub-graph. Cards are arranged in the 3D space, with which a user can interact by focus+context technique.

RF-Cone [17] generates the tree structures when the documents of a certain topic are given, based on the similarity among documents and path length from root document to each document, and visualizes them with 3D RF (relationship focused) cone tree representation.

The Category Map [18] employs SOM (self-organizing map), based on which documents are mapped onto 2D category map. Each region (a group of neighboring nodes with the same concept) corresponds to the document cluster of the concept. As SOM preserves the topological properties of document space, the Category Map can show users a relationship among document clusters.

2.3. Relevance Feedback

Interaction should be bidirectional. That is, interactive interface should not only provide users with information in understandable manner, but also get their intentions and preferences. Relevance feedback (RF) is one of major approaches for implicitly obtaining the users' preferences.

Conventional RF algorithms [9] modify a profile (query) vector based on user's judgment (relevant or irrelevant) on the retrieved documents. In this case, the user's intention that is represented by keywords is estimated indirectly from the document space. The FISH View system [11] extracts the user's viewpoint from the diagram, in which the user groups documents hierarchically. There also exists the system that supports the user's query generation by presenting the related keywords [10]. However, it is not RF approach in the sense the user has to generate the Boolean query manually.

In this paper, we propose keyword map-based RF, which infers the user's intention from the keyword space. Compared with the conventional RF algorithms applied on document space, it is expected to infer the user's fuzzy preferences. Furthermore, keyword arrangement can reflect user's intention more implicitly than the diagram used by the FISH View system.

3. Web Interface for RBA-based Interaction

3.1. Concept of Retrieval, Browsing, Analysis (RBA)-based Interaction

One of the essential properties of our activities in the Web is that we do not always have the topics of interest while surfing on the Web. Therefore, not only submitting relevant queries, but also evaluating the relevance of web pages is difficult for us. Through the interaction with the Web, we find the topics of interest, and acquire the background knowledge about the topics, based on which the relevance of pages is evaluated. Visualizing (partial) Web hyperspace as well as document clustering can improve the interaction between a user and the Web, as shown in Section 2.

Considering the commercial success of web search engines, it is rational that we assume the following steps for locating and gathering information in the Web:

[Retrieval] Obtain a set of pages by submitting tentative query to a search engine.
[Browsing] Starting from individual documents in the retrieved results, browse their neighboring pages and collect (save) the relevant ones.

We call the interaction based on these two steps RB-based interaction. It should be noted that a user cannot always evaluate the relevance of pages correctly, and the evaluation criteria frequently changes while interacting with the Web. In other words, the context that affects the evaluation criteria is composed of the pages that have been gathered so far. Therefore, we claim that the “analysis” step should be combined with RB-based interaction. We call the interaction based on these three steps RBA-based interaction. Although Gershon [3] has already denoted the importance of the analysis step, in which the properties within a single page is analyzed. Our focus is on analyzing the set of gathered documents.

From this viewpoint, some of information visualization systems denoted in the previous section contribute for supporting RBA-based interaction. However, they put the analysis step inside retrieval and browsing steps. That is, the visualized space by browsing support systems is mainly used for users to browse the hyperspace. The space visualized by clustering-based information visualization systems helps user explore the retrieved space. On the other hand, we propose to visualize the set of documents that is gathered as a result of the user’s RB-based interaction.
3.2. System Architecture

Document clustering-based visualization is employed in the proposed system, because it is assumed that a user usually gathers the pages of interest from various Web sites, and most documents have no direct hyperlink to others. In particular, this assumption becomes valid in retrieval step.

In order for users to understand context information from the visualized results, presenting only document clusters is not enough, but the relationship among clusters should also be presented. The SOM-based visualization systems can satisfy this to some extent, but the obtained structure seems to be fixed, even if users can manipulate the visualized space with fisheye or fractal operation [18]. Furthermore, we think that the obtained document clusters should be presented to users as the lists, because the Web users are familiar with the document lists that are returned by most of search engines. Therefore, we propose to visualize both of document and keyword space. Document clusters are presented to users as lists, while keyword space is visualized so that the relationship among document clusters can be reflected (Fig. 1). For visualizing the keyword space, we employed the keyword map [14], on which the keywords extracted from documents are arranged so that the pair of keywords that frequently appears in the same documents can be arranged closely to each other.

The point is how to relate the keyword map with document space, and we propose a landmark-based approach, called plastic clustering method [12,13], which is described in Section 3.5.

Figure 1 Correspondence between Document Space and Keyword Map

3.3. Keyword Map Visualization System

A keyword map-based information visualization system is developed for visualizing the topic distribution within a document set [14]. The developed system called TMIT (Topic Map Idea Tool) employs the spring model [16] to arrange keywords on 2D space. Although a number of information visualization systems employ the 3D graphics, they seem to be suitable for the facilities such as museum, where visitors use the systems. We claim that the system that can be in daily use should be simple. Therefore, we employ the 2D graphics. The basic algorithm of TMIT is as follows.

1. Define the distance $l_{ij}$ between keyword $i$ and $j$ based on their similarity $R_{ij}$ by Eq. (1) ($m$ is positive constant).

   $$ l_{ij} = m(1 - R_{ij}) $$

2. The moving distance of keyword $i$ in each step, $(d_{xi}, d_{yi})$, is calculated by Eq. (2).

   $$ (d_{xi}, d_{yi}) = \left( c \frac{\partial E}{\partial x_i}, c \frac{\partial E}{\partial y_i} \right) $$

   $$ E = \sum_{i} \sum_{j} \frac{1}{2} k_{ij} \left( d_{ij} - l_{ij} \right)^2 \left( x_i - x_j \right)^2 + \left( y_i - y_j \right)^2 \right) $$

3. In each step, the center of gravity is adjusted to the center of 2D space.

In addition to this basic algorithm, an arrangement priority based on spring constant is introduced [14]. It can be understood from Eq. (3) that the influence of strong spring (with large spring constant) is greater than that of weak ones. Here, the springs connecting to focused keywords (such as landmarks) are given larger spring constant than others, so that they can have priority than other keywords in terms of arrangement. Experimental results have shown that the keyword arrangement highlighting the focused keywords can be constantly obtained [15].

3.4. Immune Network Metaphor for Keyword Map Generation

A plastering clustering method [12,13] has been proposed to generate a keyword map as well as document clusters. On the keyword map, the keywords related with the same topic are assumed to gather and form a cluster. The plastic clustering method extracts a representative keyword, called landmark, from each cluster. As the border of keyword clusters on the keyword map is usually not obvious, the constraints for extracting a landmark is adopted from the viewpoint of document clustering. That is, when documents containing the same landmark are classified into the same cluster, there should not exist overlapping among clusters. The algorithm of the plastic clustering method is as follows:

1. Extraction of keywords (nouns) from a document set, by using the morphological analyzer and the stop-word list. In this paper, only the keywords contained in 3 or more documents are extracted.

2. Construction of the keyword network by connecting the extracted keywords $k_i$ to other keywords $k_j$ or documents $d_j$.

   $\text{Another arrangement priority based on frictional force is also introduced for considering the topic stream, which is out of scope and omitted in this paper.}$
(a) Connection between \( k_i \) and \( k_j \): \( D_{ij} \) indicates the number of documents containing both keywords.

**Strong connection (SC):** \( D_{ij} \geq T_k \).

**Weak connection (WC):** \( 0 < D_{ij} < T_k \).

(b) Connection between \( k_i \) and \( d_j \): \( TF_{ij} \) indicates the term frequency of \( k_i \) in \( d_j \).

**SC:** \( TF_{ij} \geq T_d \).

**WC:** \( 0 < TF_{ij} < T_d \).

3. Calculation of keywords' activation values on the constructed network, based on the immune network model (Eq. (5)-(9)).
4. Extraction of the keywords that activate much higher than others as landmarks, after convergence.
5. Generation of document clusters according to the landmarks

The algorithm is also shown in Fig. 2. In step 4, a convergence means that the same set of keywords always becomes active (having much higher activation values (about 100 times higher in the experiments) than others [12]), which is observed after at most 1000 times calculation in most of the experiments. As for the immune network model in Step 3, the simple model that has been proposed in the field of computational biology is adopted (Eq. (5)-(9)).

![Landmark Finding Algorithm](image)

**Figure 2** Landmark Finding Algorithm

\[
\frac{dX_i}{dt} = s + X_i \left( f \left( h^k \right) - k_i \right),
\]

\[
h^k = \sum_j J_{ij}^k X_j + \sum_j J_{ij}^A A_j,
\]

\[
\frac{dA_i}{dt} = (r - k_A h^A) X_i,
\]

\[
h^A = \sum_j J_{ij}^A X_j,
\]

\[
f \left( h \right) = p \left( \frac{h}{h + \theta_1} \right) \left( \frac{h + \theta_2}{h + \theta_2} \right),
\]

where \( X_i \) and \( A_i \) are the concentration (activation) values of antibody \( i \) and antigen \( i \), respectively. The \( s \) is a source term modeling a constant cell flux from the bone marrow and \( r \) is a reproduction rate of the antigen, while \( k_A \) and \( k_k \) are the decay terms of the antibody and antigen, respectively. The \( J_{ij}^A \) and \( J_{ij}^k \) (\( \in \{0, WC, SC\} \)) indicate the strength of the connectivity between the antibodies \( i \) and \( j \), and that between antibody \( i \) and antigen \( j \), respectively. The influence on antibody \( i \) by other connected antibodies and antigens is calculated by the proliferation function (9), which has a log-bell form with the maximum proliferation rate \( p \).

The correspondence between the objects of immune system and those of the plastic clustering method is shown in Table I. The quality of extracted landmark keywords has been verified based on questionnaires [15].

<table>
<thead>
<tr>
<th>Plastic Clustering</th>
<th>Immune System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>Antigen</td>
</tr>
<tr>
<td>Keyword</td>
<td>Antibody</td>
</tr>
<tr>
<td>Landmark</td>
<td>Active antibody (winner)</td>
</tr>
</tbody>
</table>

**Table I** Object Correspondence between Plastic Clustering & Immune System

### 3.5. Relevance Feedback based on Keyword Map

Section 3.4 considers the information flow from the system to a user. In this subsection, the information flow from a user to the system is considered.

When a keyword map is presented to a user, he usually finds the difference between the keyword arrangement on the map and his background knowledge. Therefore, he wants to modify the arrangement, as he likes. If the system can infer the user's intention from the keyword map modified by him, relevance feedback can be available.

Let us consider the following cases.

1. A user rearranges the keywords A close to keyword B, which were initially arranged far away from each other.
2. A user moves apart keyword A and B, which were initially arranged close to each other.

In the first case, the user estimates the relationship between keyword A and B closer than the initial keyword map. Therefore, collecting new document that contain both keywords should satisfy the user's interest. The latter case might be more complicated, and there will be several possibilities. As for one possibility, the user might want to divide the topic represented by keyword A and B into two detailed topics. In this case, Finding new keywords that bridge keyword A and B will be useful for the user.

In this paper, we only focus on the first case and propose the algorithm that can infer the user's intention from the modification on a keyword map. In the following algorithm, an input data file for keyword map (KData) and the data file for keywords' coordinates in the modified map (XYData) are given. KData stores the similarity \( S_{kA} \in [-1,1] \) for every keyword co-occur within documents, and \( S_{kA} = 1 \) when they do not appear within the same document.

\( ^2 \) In the current keyword map system, \( S_{kA} \in [0,1] \) when keywords co-occur within documents, and \( S_{kA} = 1 \) when they do not appear within the same document.
keyword pair \( p_i \) \((w_n, w_m)\), and XYData stores the coordinates \((x_i, y_i)\) of every keyword \( w_i \) on the map modified by a user.

1. Calculate similarities \( S_{W_d} \) for each keyword pair \( p_i \), based on the distance \( d_i \) between the keywords, which is calculated from XYData. The \( d_M \) is the maximum distance among all keyword pairs.

\[
S_{W_d} = 1 - \left( \frac{d_i}{d_M} \right) 
\]

2. Translate \( S_{W_d} \) into value within \([0, 1]\) by Eq. (11).

\[
S'_{KI} = 0.5 \left( 1 + S_{KI} \right) 
\]

3. For each keyword pair \( p_i \), calculate the degree of "farness" \( \text{Far}(S'_K) \) in KData, and the degree of "nearness" \( \text{Near}(S'_K) \) in XYData by Eq. (12) and (13).

\[
\text{Far}(x) = \max \left( \frac{x}{1-t}, 0 \right), \quad (12)
\]

\[
\text{Near}(x) = \max \left( \frac{x-t}{1-t}, 0 \right). \quad (13)
\]

4. Sort the keyword pairs in descending order of the value calculated by Eq. (14).

\[
V_i = \text{Near}(S'_K) \cdot \text{Far}(S'_K) > 0, \quad 0 \ldots \text{otherwise}. \quad (14)
\]

4. System Implementation and Experiments

4.1. Outline of RBA-based Interface

A prototype system is developed based on the description in the previous section. When designing the system, we consider the followings:

1. The system should be used by users, in combination with Web browsers for everyday use, such as IE and Netscape.
2. It should be used independent of platform (OS, hardware, etc.).
3. Further improvement or addition of new analyzing functionality should be possible in future.

Therefore, we employ server-side programming technique, as shown in Fig. 3.

Figure 3 System Configuration

In Fig. 3, a user can interact with the Web with ordinary Web browsers as usual. The system displays a small control panel on a separate browser window, on which the user gives several instructions to CGI programs, such as follows:

- **[get_page]** Collects the information of the page specified by a user.
- **[get_link]** Extracts and displays the link information within the specified page.
- **[get_pages]** The page returned by "get_link" instruction adds checkboxes to individual links, by checking which a user can collect several pages in one instruction.
- **[analyze]** The collected page information is stored in the user information DB, to which the plastic clustering method is applied and the results including document clusters and keyword map data are returned to the user.

As the result of "analyze" instruction, the document clusters are returned as the Web page consisting of clusters with URL lists and landmarks. The page also contains the link to the data set of generated keyword map.

A user can download the data set and display it with TMIT, which is implemented with JAVA, as independent tool, not as an applet. We decide not to implement the keyword map as an applet, but to provide users with the data set. The reason of providing the data set is that it consists of the connection strength of each keyword pair, which can be utilized by users with other tools than keyword map.

Fig. 4 shows the example of keyword map generated by the prototype interface. The keywords within white rectangle are landmark keywords. On the other hand, Fig. 5 shows the keyword map of same document set when no landmark is used. The comparison of both figures clearly shows landmarks improve the readability of keyword map.
4.2. Experiments on Keyword Map-based Relevance Feedback

The experiments on keyword map-based relevance feedback are performed with using the prototype interface shown in Section 4.1, combined with the algorithm described in Section 3.5. It should be noted that the experiments are performed on Japanese Web pages, and results are translated from Japanese into English.

A query “Artificial Intelligence (AI)” is submitted to the Google, and top 10 pages are collected as an initial page set, from retrieved result. Here, the pages that have apparently no relation with AI are skipped. Figure 6 shows the keyword map that the prototype interface generates from the initial page set. In Figure 6, the word “Robot” (indicated with rectangle) and “Logic” (indicated with circle) are arranged far away from each other.

The keyword map system used in the experiments is improved to be interactive, so that the relevance feedback can be available. In particular, the system can output the XYData as noted in Section 3.5. Figure 7 shows the map that is modified by a user from the initial arrangement. In Figure 7, “Robot” and “Logic” are close to each other.

Table II and III show the list of pairs extracted with the proposed algorithm, from the initial map (Figure 6) and the map modified by the user (Figure 7), respectively. Table III shows that the pair “Robot” and “Logic” is extracted with the highest rank. It is also worth noting that the number of extracted pairs from Figure 6 is less than that from Figure 7. It means that the initial arrangement faithfully represents the relationship among keywords in KData.
New query “Robot AND Logic” is submitted to Google, and 10 pages are collected in the same manner the initial pages are collected. The initial page set is updated by adding them to it. Figure 8 shows the keyword map that is generated from new page set (containing 20 pages). In Figure 8, keywords are arranged automatically by the system without user’s modification. Therefore, newly corrected pages lead the keyword “Robot” and “Logic” close to each other, as in Figure 7. It means that new pages are collected so that the user’s intention can be satisfied.

5. Conclusion

The concept of Retrieval, Browsing, and Analysis (RBA)-based interaction is proposed, for which the prototype web interface is implemented. The prototype interface employs the keyword map visualization system so that users can easily understand the context of their interaction with the Web. The relevance feedback based on interactive keyword map system is also proposed, and the experimental results show user’s intention can be extracted from the modified keyword map, and relevant pages can be collected. Consideration of more complicated feedback as well as further experiments will be our future works.

References

[10] Sunayama, W., Ohsawa, Y. and Yachida, M. “A Search Interface with Supplying Search Keywords by Using Struc-


