Visualization of Document Retrieval using External Cluster Relationship

TAECHANG JEE1, HYUNJIN LEE2 AND YILBYUNG LEE1

1School of Computer Science
Yonsei University
Seoul, 120-749 Korea
2Department of Computers, Information and Communication
Korea Soongsil Cyber University
Seoul, 110-340 Korea

Owing to the limitations of existing visualization schemes, existing document retrieval systems display limited results, often showing only document titles, short summaries, and keywords. This makes it difficult to examine multiple results at once or to find a meaningful relationship between results. This study proposes a new method for the real-time visualization of document retrieval results via clustering. The method clusters similar documents into groups, making it easier to understand the relationship between the retrieved documents. This study also proposes a two-level visualization algorithm which projects the cluster centers onto a two-dimensional space using multidimensional scaling in order to illustrate the relationships among different clusters, and displays individual documents at locations determined by the external cluster relationship in low dimensional space in order to allow the comparison of individual documents. The method was tested on benchmark data and real-world data, and the results show that it is possible to visualize the search results in real time.

Keywords: clustering, information visualization, on-line computation, text mining, document visualization

1. INTRODUCTION

Document clustering is a common data mining technique for information retrieval, and the subject of extensive research in the fields of knowledge classification and visualization [1-4]. Clustering assigns data objects to groups on the basis of a similarity measure, leading to a better understanding of the structure of the data [5-7]. Many research projects including SONIA and Scatter/Gather focus on the application of clustering algorithms to documents [8-11].

In the current clustering and information search environment, users generally enter queries in textual form and receive the results in the same textual form, facing the inconvenience of having to manually extract the required information from the results using the title, author, content, or summary. According to various studies, however, these substitutes do not play a decisive role in helping users understand the contents of the text [12]. Thus, if clustering results are provided in text form, it becomes equally difficult for common users to utilize the clustered result. Therefore, although the visualization of clustering results requires further investigation, it has not been studied extensively thus far [13].
This paper proposes a visualization algorithm that enables users to easily relate clustering results with actual documents by expressing the clustering results in visual form. Since users require quick responses from an online information retrieval system, a system that uses clustering but takes too long to produce its result is effectively unusable. The proposed algorithm uses clustering results directly instead of reducing them to lower dimensions, thereby producing much faster visualization results than existing methods. For high dimensional documents, it is difficult to view the real distribution of documents; hence, we need to reduce the dimensions of the document vectors by mapping them to a low dimensional plane. For this, we propose a fast method to locate and distribute documents using external cluster relationships.

The rest of the paper is organized as follows. In section 2, the existing literature on clustering visualizations is briefly reviewed. The characteristics of proposed method are described in detail in section 3. In section 4, the experimental results are shown. The interface and the operators are described in section 5. Finally, a conclusion is given in section 6.

2. RELATED WORK

The major approaches to dimensionality reduction in the text visualization community use topology preserving algorithms such as Principal Component Analysis (PCA), multidimensional scaling (MDS) and self-organizing maps (SOMs) [14]. Topology preserving algorithms aim to represent high dimensional data spaces in a low dimensional space while preserving the structure of the data as far as possible. This is achieved by mapping “points in one space to points in another space such that nearby points map to nearby points (and sometimes in addition far away points map to far away points)” [15].

The Galaxies visualization [13, 16] displays clusters and document relationships by reducing a high dimensional representation of documents into a two dimensional scatter plot. The documents are clustered in the high dimensional space through a metric of similarity such as Euclidean distance or cosine measure, and projected onto a 2D space that reflects these document clusters. The ThemeScape visualization [16] uses an abstract, three dimensional landscapes to convey information about the themes in the document corpus. The ground plane projects the document sets, while the peaks represent document clusters and the valleys represent the distances between these document clusters as found in the raw document sets. For small document sets (up to 1500 documents), the Sheppard multidimensional scaling algorithm is typically used, while for large document sets, the Anchored Least Stress algorithm has been developed.

Substantial research has been conducted on the utilization of SOMs for the interactive exploration of document collections [17-19], one example being the WEBSOM project [20]. SOMs are used to represent documents on a map to provide an insightful view of the document collection. This view visualizes similarity relations between the documents. The complete WEB-SOM method involves a two-level SOM architecture comprising a word category map and a document map. SOMs are used to construct a word category map, where related words that have similar context appear close to each other. The documents are encoded by mapping their text onto the word category map. The document map is then formed with a SOM algorithm using the document vectors in word
category map space. [15] discusses the use of SOMs for clustering and visualization and presents a comparative study on the quality and effectiveness of SOMs and Sammon’s mapping for classification and visualization. The Aduna Cluster Map [21] serves as a detailed, interactive visualization for hierarchically classified objects, instantiated taxonomies and concept hierarchies, and is suitable for interactive visualizations. However, it is computationally intensive and hence impractical for large data sets.

The Bead system displays bibliographic documents as cubes where the documents are placed on the surface of the cubes in clusters according to similarity [22]. The ISI-DOR interface represents the information space in the form of a cone where the axes correspond to query terms [23]. A similar approach is used in TOFIR [24]. Some researchers have analyzed the content of a document collection in order to enhance keyword based information retrieval and tried reducing the document content to a keyword dimension [25, 26]. T. Keller et al. studied the impact of dimensionality and color coding [27], concluding that information visualizations support knowledge acquisition, two dimensional information visualization is better for knowledge acquisition and color coded information visualizations slightly increase performance of knowledge acquisition.

3. PROPOSED METHOD

This paper proposes a visualization algorithm whose calculation time increases linearly with the volume of data and the dimensionality of the document feature vector [28].

3.1 System Architecture

The architecture of a document clustering system is presented in Fig. 1. First, the entire document corpus or search result corpus is retrieved. Then, the documents’ feature vectors are extracted by applying morphological analysis and parsing on each document. We then apply a clustering algorithm to the document feature vectors. K-Means was used as the primary clustering algorithm [29] in this study.

![Fig. 1. System architecture.](image)

After clustering is complete, the center point for each cluster is extracted and each document’s degree of membership to every cluster is calculated. Our visualization is
based on the concept of universal gravitation and represented in vector form. A document vector has both magnitude and direction. The direction may be described by an origin and a coordinates, while the magnitude is given by the document’s degree of membership to its cluster. The origin of document vector, in our case, is the center of the cluster. To calculate the coordinates, or the actual document coordinate, we carry out two-step visualization.

In the first step, each cluster center is mapped on a low dimensional plane using Multidimensional scaling (MDS) [10]. In the second step, individual documents are marked on the low dimensional plane using the individual documents’ degree of membership to each cluster. The individual document is expressed by the magnitude and the unit vector signifying the direction.

3.2 Summary of Proposed Method – External Cluster Relationship

Once the locations of the cluster centers are known, the locations of the document vectors are determined. Data belonging to the same cluster have a similar degree of membership to that cluster, but different relationships to other clusters. Therefore, when determining the fidelity of clustering, these ‘external cluster relationships’ should be considered. In Fig. 2, the distance between document \( d_1 \) and cluster \( C_1 \) is determined by \( d_1 \)’s degree of membership to \( C_1 \) (see dotted circle). The location of the document is fixed by the external cluster relationship \( \vec{ER}_1 \) as explained in sections 3.3 and 3.4.

The proposed method uses the \( O(KN) \) algorithm in Fig. 3. \( K \) is the number of clusters and \( N \) is the number of documents. \( C_i \) denotes the \( i \)th cluster and \( d_j \) denotes the \( j \)th document. Vector \( \vec{C_iC_j} \) joins clusters \( i \) and \( j \), expressed as \((x_j, y_j)\), is described by the direction \( \vec{C_jd_j} \) and magnitude \( ||\vec{C_jd_j}|| \) as given in Eqs. (6) and (1) respectively.

The document vector is then calculated as follows:

The coordinates of the cluster center \( C_i \) where \( i \in \{1, \ldots, K\} \) are calculated using MDS. The coordinates of individual documents belonging to each cluster are then calculated as document vectors. The document vectors consist of a magnitude \( ||\vec{C_i d_k}|| \) and a direction \( \vec{C_i d_k} \). Here, \( \vec{C_i d_k} \) is a vector from the \( i \)th cluster \( C_i \)'s center to the \( k \)th document \( d_k \), where \( k \in \{1, \ldots, N_i\} \), \( N_i \) being the number of documents in \( i \)th cluster.
3.3 Magnitude of the Document Vector

The magnitude of the document vector is calculated using the distance between the document and its cluster center.

$$\| \overrightarrow{C_i} - d_i \| = SC_i \times n_{coef_i}(d_i)$$  \hspace{1cm} (1)

Here, $SC_i$ is the size of the $i$th cluster and $coef_i(d_i)$ is document $d_i$'s degree of membership to the cluster $C_i$. This degree of membership is obtained using the cosine coefficient of correlation. The cosine coefficient is a common measure for computing the similarity between two documents, and uses their word counts [31]. It ranges from 0 to 1, with 1 indicating the highest and 0 the lowest degree of membership. The $coef_i(d_i)$ is normalized to $n_{coef_i}(d_i)$.

$$n_{coef_i}(d_i) = \frac{coef_i(d_i)}{\max_{k \in N(C_i)}(coef_k(d_i)) - \min_{k \in N(C_i)}(coef_k(d_i))}$$  \hspace{1cm} (2)

where $\max(coef)$ and $\min(coef)$ are the maximum and minimum value respectively of the degree of membership, and $N(C_i)$ is the number of documents in cluster $C_i$.

The size of cluster is calculated as follows:

$$SC_{max} = \frac{\min_{k=1,2} \| C_{max} - C_k \|}{2}$$  \hspace{1cm} (3)

$$SC_i = SC_{max} \times \frac{\sqrt{num(C_i)}}{\sqrt{num(C_{max})}}$$  \hspace{1cm} (4)

$SC_{max}$, the size of the largest cluster $C_{max}$, is calculated first. It is given by half of the distance to the nearest cluster. Then, the size of each individual cluster $SC_i$ is calculated using the ratio of number of documents in $C_i$ and $C_{max}$. The square root function is used to reduce the sizing effect; that is to prevent the smallest-sized cluster from becoming too insignificant.

3.4 Direction of the Document Vector

The direction of the document vector is calculated using the relationship between the documents and the external clusters. For example, let us assume that documents $d_{i1}$ and $d_{i2}$ belong to cluster $C_i$, and

(1) $d_{i1}$ and $d_{i2}$ have the same degree of membership with $C_i$
(2) the degree of membership between
   (a) $d_{i1}$ and $C_{j1}$ is large,
   (b) $d_{i1}$ and $C_{j2}$ is small,
   (c) $d_{i2}$ and $C_{j1}$ is small,
   (d) $d_{i2}$ and $C_{j2}$ is large.
In this case, $d_{k1}$ is placed near $C_{j1}$, while $d_{k2}$ is placed near $C_{j2}$. As shown in the above example, the direction of a document vector is determined not only by the value of its cluster center, but also by its similarity to values of other cluster centers. In order to determine the direction of a document vector, the directions of the vectors between cluster centers should first be determined.

$$
\overrightarrow{C_iC_j} = \frac{(x_i - x_j, y_i - y_j)}{\| \overrightarrow{C_iC_j} \|}
$$

(5)

$\overrightarrow{C_iC_j}$ is the vector between the $i$th and $j$th clusters. Here, $j \in \{1, \ldots, K\}, j \neq i$, and $x_i$ and $y_i$ are the x- and y-coordinate of the cluster $C_i$, respectively. (For a three-dimensional space, we can simply add the appropriate z coordinates.) Finally, the direction of the document vector is calculated as follows:

$$
\overrightarrow{C_id_k} = \sum_{j=1,j \neq i}^{K} (\text{coef}_{C_i}(d_k) \times \overrightarrow{C_iC_j}).
$$

(6)

4. EXPERIMENTAL RESULTS

To verify the proposed method, we tested it using several sets of documents. As shown in Table 1, we used four data sets from the Reuter-21578 document set [32] and four query results from the patent document search engine [33]. The algorithms were implemented in the C# programming language on Microsoft .NET Framework 2.0, and run on a Core2Duo 2.0 GHz CPU with 1 GB of RAM.

4.1 Visualization Results

Table 2 shows the execution time of proposed method. As the Reuter-April data set is larger than the other data sets, it is seen that the execution time for it is a bit longer as well.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Num. of Instances</th>
<th>Num. of Features</th>
<th>Num. of Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>February</td>
<td>132</td>
<td>250</td>
<td>6</td>
</tr>
<tr>
<td>April</td>
<td>2,106</td>
<td>2,671</td>
<td>5</td>
</tr>
<tr>
<td>June</td>
<td>988</td>
<td>1,740</td>
<td>5</td>
</tr>
<tr>
<td>October</td>
<td>605</td>
<td>828</td>
<td>5</td>
</tr>
<tr>
<td>Patent</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Car Navigation</td>
<td>1,000</td>
<td>765</td>
<td>5</td>
</tr>
<tr>
<td>Computer</td>
<td>1,000</td>
<td>739</td>
<td>6</td>
</tr>
<tr>
<td>Iris Recognition</td>
<td>1,000</td>
<td>736</td>
<td>5</td>
</tr>
<tr>
<td>Printer</td>
<td>1,000</td>
<td>642</td>
<td>5</td>
</tr>
</tbody>
</table>
Table 2. Execution time.

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Visualize Centers of Clusters (sec)</th>
<th>Visualize Each Documents (sec)</th>
<th>Total Elapsed Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuter</td>
<td>February 0.046</td>
<td>0.001</td>
<td>0.047</td>
</tr>
<tr>
<td></td>
<td>April 0.082</td>
<td>0.002</td>
<td>0.084</td>
</tr>
<tr>
<td></td>
<td>June 0.042</td>
<td>0.001</td>
<td>0.043</td>
</tr>
<tr>
<td></td>
<td>October 0.032</td>
<td>0.001</td>
<td>0.033</td>
</tr>
<tr>
<td>Patent</td>
<td>Car Navigation 0.042</td>
<td>0.001</td>
<td>0.043</td>
</tr>
<tr>
<td></td>
<td>Computer 0.042</td>
<td>0.001</td>
<td>0.043</td>
</tr>
<tr>
<td></td>
<td>Iris Recognition 0.042</td>
<td>0.001</td>
<td>0.043</td>
</tr>
<tr>
<td></td>
<td>Printer 0.042</td>
<td>0.001</td>
<td>0.043</td>
</tr>
</tbody>
</table>

Figs. 4 and 5 are the result of visualizing individual documents using the proposed method. In this figure, since the document distribution was not tilted toward any one specific direction, it was concluded that documents belonging to individual clusters did not necessarily show the same tendencies. It could also be seen that there was a mixture of similar and different documents even within a same cluster.

In Fig. 4, the documents of cluster 0 are located in the lower part of the circle, meaning that those documents are mutually similar and also similar to those in cluster 5 (which is the lowest cluster). In cluster 5, there are some documents related to cluster 0, but most of them are not related to cluster 0. Therefore cluster ‘0’ and ‘5’ are not united together.

In Fig. 5, the documents of cluster 0 are scattered, indicating that they have low similarity with external clusters. The documents in the upper left region of cluster 0 are tightly grouped, indicating that they share similar relationships with external clusters.

An illustration of the relationship between clusters and documents is shown in Fig. 6 (a) and (b) are document groups within cluster 1. Group (a) is closer to cluster 2, and group (b) is closer to cluster 3. Although groups (a) and (b) are both in cluster 1, we can see their similarity to cluster 2 and cluster 3 respectively.

4.2 Time Complexity

The time complexity of MDS is $O(lK^2)$, where $K$ is the number of clusters and $l$ is the iteration time; thus the execution time is proportional to the square of the number of clusters. The time complexity of the proposed method is $O(KN)$, where $K$ is the number of clusters and $N$ is the number of documents; in other words, execution time is proportional to the number of clusters and to the number of documents.

Experimental results for execution time are shown in Fig. 7. In our experiment, the number of clusters was varied from 10 to 1,000 while the number of documents was fixed at 1,000. The execution time for MDS is seen to increase in proportion to the square of the number of clusters, while the execution time for the proposed method increases linearly with the number of clusters. Consequently, while the MDS-only method took 23 seconds to visualize 1,000 documents, it took less than one second when combined with the proposed method (using MDS for visualizing cluster centers and the pro-
Fig. 4. Visualization results for Reuter – April.

Fig. 5. Visualization results for patent – car navigation.

Fig. 6. Analysis of result.
The proposed method for visualizing the documents). This clearly shows that the proposed method can efficiently visualize a large volume of data in an online system.

5. INTERFACE

5.1 Main Page

The visualizations described in section 3 provide a starting point to the user (see Fig. 8). The user can then browse the document cluster using roll-up and drill-down tools until he reaches the level he is interested in. The density of color indicates the number of documents in the cluster.

Because there are many documents, they tend to overlap in a cluster map. The user can right-click to view individual documents in such regions (see section 5.3), and use roll-up and drill-down tools to easily recognize individual documents.

5.2 Keyword Tree

Fig. 8 (a) illustrates a keyword tree. The keyword is obtained by clustering, with the number of keywords specified by the user. The number following each keyword indicates the number of documents in that cluster. If the user double-clicks a keyword, clustering is performed for the data in the selected keyword cluster. The time required for this next-level clustering is usually very low, i.e., less than 1s. After the clustering, a child keyword tree and a cluster map are generated (see Fig. 9). When a keyword is selected in the keyword tree, the corresponding cluster is automatically selected in the cluster map (and vice versa).

5.3 Cluster Map

Fig. 8 (b) illustrates the cluster map generated by the proposed method. Double-clicking a circle triggers next-level clustering, as described in section 5.2 (see Fig. 9).

Right-clicking on a cluster shows the number of data points and keywords for that cluster along with actions for view documents roll-up and drill-down (see Fig. 10).
Using ‘View Documents’, the user can view the individual documents contained in the cluster (see Fig. 11). Roll-up and drill-down allow the user to move up and down the levels (see Fig. 9).

Right-clicking on the document shows the document code, and command for view documents close to selected document (see Fig. 12).
6. CONCLUSION

In this study, we proposed a method to visualize document search results by clustering the search results, mapping the cluster centers to a low dimensional plane using MDS, and finally calculating the coordinates of individual documents using the cluster centers.

In the first stage of visualization, the clustering process was carried out using terms contained in the individual documents. In the second stage, we visualized the relationships between individual documents was designed by applying MDS to the cluster center data created in the first stage and placing the documents in a low dimensional space using the relationship between external clusters and individual documents. As a result, users could easily discover not only the relationships between clusters of similar documents but also the similarity between individual documents. In addition, our experiment showed that this method can be a very effective and time-saving solution for online document search systems. The interface is based on a live clustering result, so the initial visualization requires no user input. For online multi-level clustering, sub-cluster visualization consumes less than 1s.

This paper presented the clusters and documents on a two-dimensional plane. However, dimensionality reduction methods do not limit the number of resulting dimensions. Therefore, if cluster centers are projected to a three-dimensional space, it is possible to visualize documents in three-dimensional space using the proposed method. For future studies, we intend to investigate the suitability of three-dimensional visualization in online retrieval systems through further experiments. Another area of investigation involves automatically incorporating a new document into a cluster, i.e. incremental clustering, which would reduce clustering time in most cases.
REFERENCES


**Taechang Jee** received his B.S., M.S. and Ph.D. degrees in Computer Science from Yonsei University, Korea, in 1997, 1999, and 2012, respectively. Since 1999, he has been a Research and Development Advisory with LG CNS, Korea, for data mining, data warehousing and advanced analytics. His current interests are in advanced analytics, text analytics and artificial intelligence.
Hyunjin Lee received the B.S. degree in Computer Science from Soonchunhyang University in 1996, and the M.S. and Ph.D. degrees in Computer Science from Yonsei University in 1998 and 2002 respectively. She is now an Associate Professor in Department of Computer, Information and Communication at Korea Soongsil Cyber University. She is interested in neural networks, data mining and e-learning.

Yillbyung Lee received the B.S. degree in Electronic Engineering from Yonsei University in 1976 and M.S. degree in Computer Science from University of Illinois in 1980, and Ph.D. degree in Computer and Information Science from University of Massachusetts in 1985. He is now a Professor in the Department of Computer Science at Yonsei University. He is interested in document recognition, data mining, and biometrics.