JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 10, NO. 6, JUNE 2007(pp.699-706)

Web Image Clustering with Text Features and Measuring its Efficiency

Soosun Cho[†]

ABSTRACT

This article is an approach to improving the clustering of Web images by using high-level semantic features from text information relevant to Web images as well as low-level visual features of image itself. These high-level text features can be obtained from image URLs and file names, page titles, hyperlinks, and surrounding text. As a clustering algorithm, a self-organizing map (SOM) proposed by Kohonen is used. To evaluate the clustering efficiencies of SOMs, we propose a simple but effective measure indicating the accumulativeness of same class images and the perplexities of class distributions. Our approach is to advance the existing measures through defining and using new measures *accumulativeness* on the most superior clustering node and *concentricity* to evaluate clustering efficiencies of SOMs. The experimental results show that the high-level text features are more useful in SOM-based Web image clustering.

Keywords: Web Image Clustering, Measure of Clustering Efficiency, Self Organizing Map, Web Image Features

1. INTRODUCTION

For retrieval Web images which usually not annotated using semantic descriptors, many content-based image retrieval (CBIR) systems have been developed. And the field of CBIR has made significant advances during past decade[1,2]. CBIR addresses the problem of finding images relevant to the users' information needs, based principally on low-level visual features for which automatic extraction methods are available. However CBIR has a drawback in that many searches return entirely irrelevant images that just happen to possess similar features, and therefore, the performance of CBIR system are at low levels[3]. The main reason is that low-level visual features cannot represent the high-level semantic content of images. Hence, some research efforts (e.g. [4,5]) focus on how to combine low-level features and high-level semantic features together to retrieve images.

As representative research using text features on Web image retrieval, Z. Chen et al.[4] describe a Web image search engine using Web mining. This work made a contribution to increase the effectiveness of searching by extracting the text information on the Web page to semantically describe the images and combining with other low-level image features.

Instead of using text features in query and searching images, text feature-based clustering methods have also many advantages on feature extraction and indexing methods for particular classes. In this paper, we propose an approach to improving the clustering of Web images by using high-level semantic features from text information relevant to Web images as well as low-level visual features of image itself. These high-level text features can be obtained from image URLs and file

※ Corresponding Author : Soosun Cho, Address : (380-072)123 Iryu Chungju Chungbuk Korea, TEL : +82-43-841-5262, FAX : +82-43-841-5260,
E-mail : sscho@cjnu.ac.kr

Receipt date : March 7, 2007, Approval date : June 5, 2007

[†] Dept. of Computer Science, Chungju National University

※ This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-611-D00026)

names, page titles, hyperlinks, and surrounding text. As a clustering algorithm, a self-organizing map (SOM) proposed by Kohonen[6] is used. Moreover we define a measure for clustering efficiency and evaluate the results from SOM clusters.

The rest of this paper is organized as follows: In Section 2, we introduce the works related with our approaches and the technical background of SOM. In Section 3, we represent the system architecture and functions of our Web image clustering prototype, including feature extractors and the SOM-based cluster. The proposed measure for clustering efficiency is explained in Section 4. The experiments and the evaluations are discussed in Section 5, and we conclude in Section 6, finally.

2. RELATED WORKS AND TECHNICAL BACKGROUNDS

2.1 Web Image Retrieval using SOM algorithm

There have been several Web image clustering and retrievals using SOMs. S.W.K. Chan et al.[7] presented a content-sensitive classifier using SOMs. They used four classes of image features, such as color-based, edge-based, region-based, and texture-based features. They also have evaluated their approach using hundred images. Evaluation showed that similar images will fall into the same region. Although various low-level image features are chosen to overcome the weakness of image classifying, they are not interested in high-level text features relevant to content of images.

The representative CBIR system with a self-organizing map is PicSOM[8,9]. PicSOM is based on tree structured self-organizing maps (TS-SOMs). Given a set of reference images, PicSOM is able to retrieve another set of images which are similar to the given ones. As the measure of similarity, they used the TS-SOMs. They used image features from average color, color moments, texture neighborhood, shape histogram, and shape FFT. High-level text features have not been considered as image features in PicSOMs.

2.2 Measuring Efficiencies

For the development of CBIR applications, one of the important issues is to have efficient and objective performance assessment methods for different features and techniques. In the CBIR applications using SOMs, the mapping of feature vectors and their associated images to their best matching units (BMUs) can be interpreted as clustering. M. Koskela et al.[10] have studied how clustered image class distributions can be interpreted in terms of probability densities and how the effectiveness of a clustering method can be assessed with entropy-based methods. From the experiments, they found that with larger SOMs, the measure is less informative as the number of images sharing a BMU becomes overly small and the perplexity value mostly reflects just the size of the image class.

To evaluate how much the clustering efficiency is improved by using text features on the SOM-based cluster, we propose a simple but effective measure to complement the perplexities. It is more informative because it represents the accumulativeness into the most superior clustering node as well as the perplexities. In the experiments, we evaluate the clustering efficiencies of SOMs using the text features and the visual features.

2.3 Self-Organizing Map

A self-organizing map is a special kind of neural network in the sense that it constructs a topology preserving mapping from the high-dimensional space onto map units in such a way that relative distances between data points are preserved. The SOM can thus serve as a clustering tool of high-dimensional data.

In the SOM algorithm, the set of input samples is described by a real vector $\mathbf{x}(t) \in R^n$ where t is the index of the sample. Each neuron i in the map contains a weight vector $\boldsymbol{w}_i(t) \in R^n$, which has the same number of elements as the input vector $\boldsymbol{x}(t)$. Adjacent neurons belong to the neighborhood N_i of the neuron i.

The learning process of the SOM goes as follows:

1. An input vector $\boldsymbol{x}(t)$ is compared with all the weight vectors $\boldsymbol{w}_i(t)$. The best-matching unit (neuron) on the map, i.e., the neuron where the weight vector is most similar to the input vector in some metric (e.g. Euclidean) is identified. This best matching unit (BMU) is often called the winner.
2. The basic idea in the SOM learning process is that, for each sample input vector $\boldsymbol{x}(t)$, the winner and the neurons in its neighborhood are changed closer to $\boldsymbol{x}(t)$ in the input data space. During the learning process, individual changes may be contradictory, but the net outcome in the process is that ordered values for the $\boldsymbol{w}_i(t)$ emerge over the array. Adaptation of weight vectors in the learning process may take place according to the following equations:

$$\begin{aligned} w_i(t+1) &= w_i(t) + \alpha(t)[x(t) - w_i(t)] \quad \text{for each } i \in N_c(t), \\ w_i(t+1) &= w_i(t) \qquad \text{otherwise}, \end{aligned} \tag{1}$$

where t is the discrete-time index of the variables, the factor $\alpha(t) \in [0,1]$ is a scalar that defines the relative size of the learning step, and $N_c(t)$ specifies the neighborhood around the winner in the map array.

At the beginning of the learning process the radius of the neighborhood is fairly large, but it is made to shrink during learning. The factor $\alpha(t)$ also decreases. After the learning is over, the map should be topologically ordered. This means that n topologically close (using some distance measure e.g. Euclidean) input data vectors map to n adjacent map neurons or even to the same single neuron. Figure 1 shows basic concept of SOM.

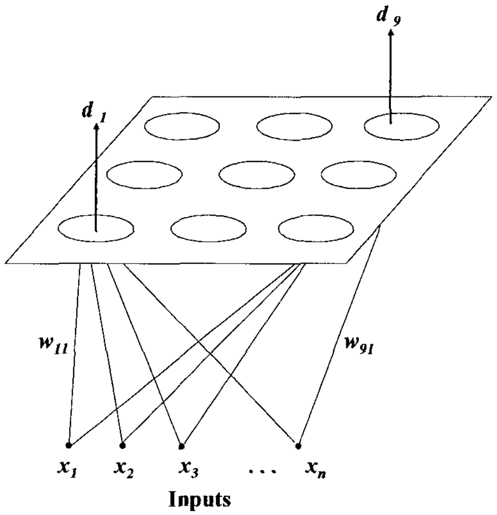

Fig. 1. The basic concept of SOM. The input vectors are connected to an array of neurons in the feature map (usually 2 dimensional): When an input is presented, certain neuron of the array will "fire" and the weights connecting the inputs to that neuron will be strengthened.

3. SOM-BASED WEB IMAGE CLUSTERING

Our Web image clustering system consists of three main components: the Text Feature Extractor, the Visual Feature Extractor, and the SOM-based Cluster. Before feature extraction, Web images and relevant text information are collected and stored in the Web Image Database. After the process by the Feature Extractor, the extracted features are stored in the Image Feature Database. The SOM-based cluster is an intelligent unit serving as the brain of the clustering system. In the training of SOM-based clusters, the weight matrix for each object class is stored as a training result and it can be used for test of constructed SOM model.

3.1 Image collection

For the clustering, we used two hundred object images collected from 10 object classes whose names are 'monkey', 'tomato', 'building', 'tree', 'boat', 'vehicle', 'bird', 'fish', 'bridge', and 'rose'. Collecting candidate images is made possible by a Web spider, which is a kind of mobile agent. The Web spider is able to travel along various designated Web pages as well as their hyperlinks to analyze and download interesting images to a local Web image databases. From the candidate images, we select 20 images and their relevant text information for each class. Thus the total number of images becomes two hundreds. Figure 2 shows 10 representative object images.

3.2 Extraction of low-level visual features

In general, images have the features of color, texture, shape, edge, shadows, temporal details etc. The features that were most promising were color, texture and edge. For our Web image clustering, we use 5 color histogram features and 5 texture features.

Color histogram features - Color has been used extensively as a low-level feature for image retrieval[1]. Color features are invariant to rotation, shift, and scaling. The 5 color histogram features in our visual feature extractor, are image mean, standard deviation, skewness, energy, and entropy. The color histogram features are computed by treating the color values in different color bands (e.g. red, green, and blue) as separate probability distributions and then calculating the first three color moments (mean, standard deviation, and skewness) of each color band. Image entropy and energy are measures of the complexity of image color distributions. Images with simple color distribution have low entropy value, while images with complex color distribution have high entropy value. Energy and entropy are also generated in 3 color bands. Therefore, the feature extraction produces a 5 x 3 = 15-dimensional color histogram feature vector.

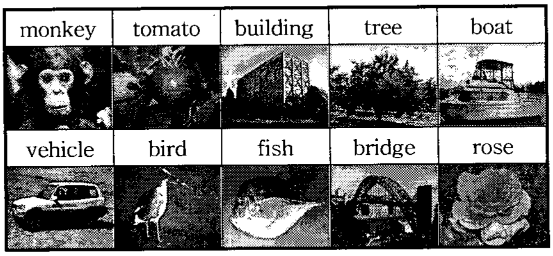

Fig. 2. Representative object images from 10 classes. From each class 20 object images are selected and used for clustering data.

Texture features - Texture is defined as a neighborhood feature -as a region or a block. The variation of each pixel with respect to its neighboring pixels defines texture. Thus texture feature requires a value for pixel distance. This distance defines which pairs of pixels are used to determine the co-occurrence matrix. A larger distance will define a coarser texture, while a smaller distance will define a finer texture. The most commonly used feature for texture analysis is the energy of the sub-band coefficients[11]. The 5 texture features in our visual feature extractor, are energy, inertia, correlation, inverse difference, and entropy. Because each texture feature consists of range and average, the generated vector is a 5 × 2 = 10-dimensional texture feature vector.

3.3 Extraction of high-level text features

The qualities and types of feature extraction have significant impact on the performance of the SOM cluster. The text features we used are extracted from the following sources on the Web page that contains the image, according to some empirical rules:

1. Image Filename and URL: to increase the usefulness of the filename and URL, we use some heuristic rules including the followings: (1) Filename filtering: some cutter letters in filenames, such as, digits, hyphens, underbars, filename extensions, etc., should be discarded. (2) URL parsing: a URL usually represents the hierarchy information of an image on the Web page. URL parsing is also needed to obtain useful information from URL.

2. Surrounding Text: in many Web pages, images are used to enhance the content that the authors want to present. Therefore, some text in the surrounding areas must be considered as semantically relevant to the content of the image. We used the text from all of the four possible areas (above, below, left, right).
3. Web Page Title and hyperlinks: the page title and hyperlinks on images can be relevant to image content. Therefore, they are also used for source of the text features for the image.

The text feature vector is generated using the TF*IDF (term frequency & inverse document frequency) method[12] which gives weight to each keyword in the text feature vector. As mentioned before, we use the 10 class object images for our SOM-based clustering, thus, the keywords are assigned with the 10 class names, such as 'monkey', 'tomato', etc. The rules of constructing text feature vectors are as follows:

$$D_i = TF_i * IDF_i$$
$$= (t_{i1} * \log(\frac{N}{n_1}),...,t_{ij} * \log(\frac{N}{n_j}),...,t_{im} * \log(\frac{N}{n_m})) \quad (2)$$

where D_i is the text feature vector of image i. t_{ij} stands for the frequency of keyword j appearing in the text description of the image I. n_j represents the number of images that include the keyword j. N is the total number of images.

4. MEASURES FOR CLUSTERING EFFICIENCY

Our SOM-based cluster is constructed by a series of training data which consists of images from 10 object classes. To investigate the clustering efficiency of the SOM, we define a simple but effective measure. The basic idea has come from the work by M. Koskela et al.[10] which defined measures for SOM. They studied how clustered image class distributions can be interpreted in terms of probability densities and how the effectiveness of a clustering method can be assessed with entropy-based methods. Because a simple and commonly used measure for the randomness of a symbol distribution is its *entropy* they use the entropy of a class distribution as equation (3). In this case, the cluster indices for the vectors of the training set play the role of symbols. The entropy H of a distribution $P = (P_0, P_1, \cdots, P_{k-1})$ is calculated as:

$$H(P) = -\sum_{i=0}^{k-1} P_i \log P_i \quad (3)$$

where k is the total number of clusters. P_i is the probability of cluster i being the correct one for an input vector. Usually logarithm base of two is used.

Instead of using entropy directly, they used a more illustrative measure perplexity PPL = 2H and defined a normalized perplexity $\overline{PPL}$ = 2H /k by employing PPLmax = k. In general it can be assumed that the clustering distributes the input vectors roughly evenly to all clusters and the normalized perplexity of the whole data should thus be near unity. On the other hand, images with semantic similarity should be mapped to a small cluster subset, provided that the clustering methods have been favorable to that specific class. In this case, normalized perplexity should be <<1.

When testing images are mapped to a small cluster subset it can be assumed that the numbers of images in such clustering nodes are bigger than others. So it is helpful to use a measure of accumulation which indicates how many same class images into one clustering node, i.e. *the most superior clustering node*. Our approach is to define *accumulativeness* on the most superior clustering node, A_i for image class i as follows:

$$A_i = \underset{j=1}{\overset{c_i}{Max}}\, M_{ij} \quad (4)$$

where c_i is the number of clustering nodes in which images from class i are included. M_{ij} is the number of images which come from class i and belong to clustering node j.

A_i becomes larger when more images from class *i* are accumulated into the most superior clustering node. The ideal case is that all images from class *i* are accumulated into the most superior clustering node. The modified A_i combined with the perplexity can be more informative measure because it represents the accumulativeness into the most superior clustering node as well as the perplexities. So we define and use a new measure *concentricity* $C_i = A_i/PPL_i$ to evaluate clustering efficiencies of SOMs. In our case, as the total number of clusters k is same in all experiments, the normalized perplexity is not needed.

5. EXPERIMENTS AND EVALUATIONS

One of major objectives in experiments is to reveal the effectiveness of Web image data features. Properly speaking, the question is whether the clustering efficiency using text features is greater than that with other visual features and if so, how much it increases. To evaluate SOM clusters using different feature vectors we have made several clustering experiments with different combinations of features. First, we use the 15-dimensional color histogram feature vector, the 10-dimensional texture feature vector, and the 10-dimensional text feature vector respectively to cluster two hundred object images. Second, the 3 combinations of each feature vectors are used: the 25-dimensional color and texture feature vector, the 25-dimensional color and text feature vector, and the 20-dimensional texture and text feature vector. Then the results from the experiments are compared each other.

The SOM with 64 nodes organized with 8x8 grid is used for training. The number of nodes used here is empirically determined by considering the number of similar clusters in the training sets. 10 training cycles are taken and learning parameter a begins at 0.5 and ends at 0.1 with exponentially decreasing.

Table 1 shows the clustering efficiencies of the SOM with the 15-dimensional color histogram feature vector, and the 25-dimensional color and text feature vector. Table 2 shows the results from the 10-dimensional texture feature vector, and the 20-dimensional texture and text feature vector. Finally, Table 3 presents the results from SOM cluster with the 25-dimensional color and texture feature vector. The better clustering efficiency can be defined as a greater sum of C_i s. From the sum of C_i s in three tables, we can find following results:

1. With the whole visual features, the 25-dimensional color and texture feature vector produces greater clustering efficiency than that from each color or texture feature vector. Comparing Table 1, 2, and 3, we can find that the sum of Ci s increases from 18.0 or 22.7 to 26.1.

Table 1. Improved clustering efficiency (from using the 15-dimensional color histogram feature vector to using the 25-dimensional color and text feature vector): When the text features are added, the sum of Ci s increases from 18.0 to 34.1.

Feature vectors	Class *i*	Accumula -tiveness A_i	Perplexity PPL_i	Concentricity $C_i = A_i/PPL_i$
15-dim color feature vector (Sum of C_i s =18.0)	1	8	6	1.3
	2	7	6	1.2
	3	7	5	1.4
	4	6	6	1.0
	5	11	3	3.7
	6	9	4	2.3
	7	6	5	1.2
	8	8	6	1.3
	9	9	4	2.3
	10	9	4	2.3
25-dim color and text feature vector (Sum of C_i s =34.1)	1	7	4	1.8
	2	15	3	5.0
	3	10	4	2.5
	4	10	5	2.0
	5	10	5	2.0
	6	14	3	4.7
	7	11	2	5.5
	8	14	4	3.5
	9	15	4	3.8
	10	13	4	3.3

Table 2. Improved clustering efficiency (from using the 10-dimensional texture feature vector to using the 20-dimensional texture and text feature vector): When the text features are added, the sum of Ci s increases from 22.7 to 39.5.

Feature vectors	Class i	Accumula -tiveness A_i	Perplexity PPL_i	Concentricity $C_i = A_i/PPL_i$
10-dim texture feature vector (Sum of C_i s =22.7)	1	9	5	1.8
	2	10	5	2.0
	3	11	5	2.2
	4	9	5	1.8
	5	12	4	3.0
	6	10	4	2.5
	7	8	5	1.6
	8	9	5	1.8
	9	11	3	3.7
	10	9	4	2.3
20-dim texture and text feature vector (Sum of C_i s =39.5)	1	10	4	2.5
	2	12	3	4.0
	3	13	4	3.3
	4	11	3	3.7
	5	16	2	8.0
	6	15	3	5.0
	7	15	3	5.0
	8	12	4	3.0
	9	12	4	3.0
	10	10	5	2.0

Table 3. The clustering efficiency by using the 25-dimensional color and texture feature vector: When two visual feature vectors are combined, the sum of Ci s increases, but the amount is smaller than that from one visual feature vector combined with text feature vector.

Feature vectors	Class i	Accumula -tiveness A_i	Perplexity PPL_i	Concentricity $C_i = A_i/PPL_i$
25-dim color and texture feature vector (Sum of C_i s =26.1)	1	10	4	2.5
	2	10	5	2.0
	3	7	5	1.4
	4	7	5	1.4
	5	11	3	3.7
	6	14	3	4.7
	7	9	5	1.8
	8	10	4	2.5
	9	13	5	2.6
	10	14	4	3.5

2. Two combined feature vectors, the 25-dimensional color and text feature vector and the 20-dimensional texture and text feature vector produce greater clustering efficiency than that from whole visual feature vector in Table 3. From the comparing Table 1 and 3, we can find that the sum of Ci s with the 25-dimensional color and text feature vector is larger than that with whole visual feature vector. From comparing Table 2 and 3, we can find the same results. They increase from 26.1 to 34.1 and 39.5.

Of course, if we use the whole feature vector, 35-dimensional color, texture, and text feature vector, the effectiveness would increase. In that case the sum of C_i s was 41.2. But the increased value is not so much compared to the sum of C_i s from 20-dimensional texture and text feature vector (Sum of C_i s =39.5). Generally speaking, smaller dimensional feature vector is more effective in image clustering if there is no big difference in results from clustering. So we may not say the efficiency from 35-dimensional feature vector is greater than that from 20-dimensional texture and text feature vector.

6. CONCLUSIONS

In this article, we have presented an approach to improving the clustering of Web images based on self-organizing maps. This approach utilizes not only low-level visual features, but also high-level text features. We have evaluated the SOM-based clustering with different combinations of image features. For the evaluation, we proposed a simple but effective measure indicating the accumulativeness of same class images and the perplexities of class distributions. From the experimental results, we can find that improved clustering efficiency by adding high-level text features is greater than that by adding another low-level visual feature set. For example, in spite of a smaller dimension, the clustering efficiency with the 20-dimensional texture and text feature vector is greater than that with the 25-dimensional color and texture feature vector.

Therefore, we can conclude that the high-level text features in our approaches are more effective in SOM-based image clustering. To overcome the limits of current CBIR systems, adopting appropriate high-level text features in classification or indexing images can be one of the good solutions.

REFERENCES

[1] Y. Rui, T. Huang, and S. Chang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues," *Journal of Visual Communication and Image Representation*, Vol. 10, No. 4, pp. 39-62, 1999.

[2] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at The End of The Early Years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, pp. 1349-1379, 2000.

[3] N. Gudivada and V. Raghavan, "Content-Based Image Retrieval Systems," *IEEE Computer*, Vol. 28, No. 9, pp. 18-22, 1995.

[4] Z. Chen, L. Wenyin, F. Zhang, M. Li, and H. Zhang, "Web Mining for Web Image Retrieval," *Journal of the American Society for Information Science and Technology*, Vol. 52, No. 10, pp. 831-839, 2001.

[5] Y. Lu, C. Hu, X. Zhu, H. Zhang, and Q. Yang, "A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems," *Proceedings of the 8^{th} ACM international Conference on Multimedia*, pp. 31-38, 2000.

[6] T. Kohonen, *Self-Organizing Maps*, Vol. 30 of Springer Series in Information Sciences, Springer, Berlin, 1995.

[7] S.W.K. Chan and M.W.C. Chong, "Unsupervised Clustering for Nontextual Web Document Classification," *Decision Support Systems*, Vol. 37, pp. 377-396, 2004.

[8] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, "PicSOM-Content-based Image Retrieval with Self-Organizing Maps," *Pattern Recognition Letters*, Vol. 21, pp. 1199-1207, 2000.

[9] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, "Self-Organizing Maps as a Relevance Feedback Technique in Content-based Image Retrieval," *Pattern analysis & Applications*, Vol. 4, pp. 140-152, 2001.

[10] M. Koskela, J. Laaksonen, and E. Oja, "Entropy-Based Measures for Clustering and SOM Topology Preservation Applied to Content-based Image Indexing and Retrieval," *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

[11] T. Chang and C.-C.J. Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," *IEEE Transactions on Image Processing*, Vol. 2, pp. 429-441, 1993.

[12] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, 1989.

Soosun Cho

1987 B.S. (Computer Science and Statistics) Seoul National University
1989 M.S. (Statistics) Seoul National University
1989~1994 Researcher, Woongjin Media Co. Ltd.
1994~2004 Senior Researcher, ETRI (Electronics and Telecommunications Research Institute)
2004 Ph.D. (Computer Science) Chungnam National University
2004~present Assistant Professor, Department of Computer Science, Chungju National University
2006~2007 Visiting Scholar, Department of Statistics, University of Michigan - Ann Arbor