Automatic Ontology Derivation Using Clustering for Image Classification[1]

Latifur Khan and Lei Wang
Department of Computer Science
University of Texas at Dallas, TX 75083-0688
Email: [lkhan, leiwang]@utdallas.edu

Abstract

Technology in the field of digital media generates huge amounts of non-textual information, audio, video, and images, along with more familiar textual information. The potential for exchange and retrieval of information is vast and daunting. The key problem in achieving efficient and user-friendly retrieval in the domain of image is the development of a search mechanism to guarantee delivery of minimal irrelevant information (high precision) while insuring that relevant information is not overlooked (high recall). The traditional solution to the problem of image retrieval employs content-based search techniques based on color, texture or shape features. The traditional solution works well in performing searches in which the user specifies images containing a sample object, or a sample textural pattern, in which the object or pattern is indexed. One can overcome this restriction by indexing images according to meanings rather than objects that appear in images, although this will entail a way of converting objects to meanings. We have solved this problem of creating a meaning based index structure through the design and implementation of a concept-based model using domain dependent ontologies. An ontology is a collection of concepts and their interrelationships which provide an abstract view of an application domain. With regard to converting objects to meaning the key issue is to identify appropriate concepts that both describe and identify images. We propose a new mechanism that can generate ontologies automatically in order to make our approach scalable. To achieve this we propose a method for the automatic construction of ontologies based on clustering and a vector space model. Similarity of images is based on similarity of objects that appear in images. For object similarity measure, we consider the combination of color and shape similarity together.

1. Introduction

The development of technology in the field of digital media generates huge amounts of non-textual information, such as audio, video, and images, as well as more familiar textual information. The potential for the exchange and retrieval of information is vast, and at times daunting. In general, users can be easily overwhelmed by the amount of information available via electronic means. The need for user-customized information selection is clear. The transfer of irrelevant information in the form of documents (e.g. text, audio, video) retrieved by an information retrieval system and which are of no use to the user wastes network bandwidth and frustrates users. This condition is a result of inaccuracies in the representation of the documents in the database, as well as confusion and imprecision in user queries, since users are frequently unable to express their needs efficiently and accurately. These factors contribute to the loss of information and to the provision of irrelevant information. Therefore, the key problem to be addressed in information selection in the domain of image is the development of a search mechanism which will guarantee the delivery of a minimum of irrelevant information (high precision), as well as insuring that relevant information is not overlooked (high recall).

[1] This study was supported in part by gift from Sun and the National Science Foundation grant NGS-0103709.

Images consist of various objects, each of which may be used to effectively classify the image. The unstructured format of images tends to resist standard categorization and classification techniques. Traditional systems used to store and process multimedia images provide no means of automatic classification. The ability of these systems to retrieve relevant documents based on search criteria could be greatly increased if they were able to provide an accurate/semantic description of an image based on image content.

The traditional solution to the problem of image retrieval employs content-based search technique based on color, histogram, texture or shape features. The traditional solution works well in performing searches in which the user specifies images containing a sample object, or a sample textural pattern. Should a user ask for an image depicting a basketball game, the results become less accurate. This is due to the fact that though an image may contain a basketball, it does not necessarily depict a basketball game. In order to overcome the shortcomings of traditional technique in responding to image classification we have designed and implemented a concept-based model using ontologies. This model, which employs a domain dependent ontology, is presented in this paper. An ontology is a collection of concepts and their interrelationships, which can collectively provide an abstract view of an application domain.

In our system we would like to address two distinct questions: the extraction of the semantic concepts from the images and the construction of an ontology. With regard to the first problem, the extraction of semantic concepts, the key issue is to identify appropriate concepts that describe and identify images. We would like to make sure that irrelevant concepts will not be associated and matched, and that relevant concepts will not be discarded. In other words, it is important to insure that high precision and high recall will be preserved during concept selection. To the best of our knowledge there are no attempts to connect images and concepts through the use of ontologies in any traditional image retrieval systems. We propose an automatic mechanism for the selection of these concepts [16].

With regard to the second problem, we propose a new method for the automatic construction of ontologies based on clustering and vector space model. It is important to note that similarity of images is based on similarity of objects that appear in images. In addition, object similarity takes into account not only color or shape, but both.

Our method constructs ontologies automatically in bottom up fashion. For this, we first construct a hierarchy using some clustering algorithms. Recall that if documents are similar to each other in content they will be associated with the same concept in ontology. After objects detection, for each object, we extract color and shape information and express them using vectors. Then considering both color and shape factors, we can calculate the similarity between objects applying vector space model. In ontologies each concept is described by a set of features (objects). Thus, before image clustering, we cluster objects according to similarities between objects and assign a weight for each object cluster. Next, we construct a vector for each images based on weights of object clusters and calculate similarities between images using vector space model. Finally, based on image similarities, we cluster images and build ontology hierarchy using hierarchy agglomerative clustering algorithm.

Section 2 of this paper discusses work related to content-based image retrieval and ontologies for use in image retrieval, as well as the current systems used for image processing. Section 3 describes ontologies, and how they specify interrelationships among concepts that help draw meaningful conclusions about images. Section 4 describes outline of our approach. Section 5 presents elaborately our approach to cluster images and build ontology hierarchy. Section 6 presents preliminary result of our approach. Section 7 presents our conclusion and possible areas of future work.

2. Related Work

Several systems exist today that attempt to classify images based on their content. Successful classification of an image and its contents relates directly to how well relevant images may be retrieved when a search is preformed. Most image storing systems such as QBIC and VisualSEEK limit classification mechanism to describing an image based on metadata such as color histograms, texture, or shape features [2, 8]. These systems have high success in performing searches in which the user specifies images containing a sample object, or a sample texture pattern. Should a user ask for an image depicting a basketball game, the results become less accurate. This is due to the fact that though an image may contain a basketball, it does not depict a basketball game. Systems that only contain metadata regarding only color and shape features contained in an image cannot provide an accurate classification of the entire image.

Other systems attempt to provide images with a more precise description by analyzing other elements surrounding the images, such as captions [9, 10], or HTML tags on web pages [12]. These systems use this information to help classify the image and give it a meaningful description. This approach, tied together with metadata on images such as texture, and color sampling has the potential to yield high precision results in image classification. Examining the textual descriptions associated with an image provides additional information that may be used to help better classify the image. Unfortunately, this approach does not take into account the connections among individual objects present in a sample image. Such connections provide useful information in the form of relationships among objects present in the image, which could be used to help classify the image's content.

For the construction of ontologies, only a few automatic methods are proposed [4, 6, 7]. Elliman et al. [7] propose a method for constructing ontologies to represent a set of web pages on a specified site. Self organizing map is used to construct hierarchy. In our case we modify self organizing tree and label nodes in the hierarchy. Bodner et al. [4] propose a method to construct hierarchy based on statistical method (frequency of words). Hoothe et al. [6] propose various clustering techniques to view text documents with the help of ontologies. Note that a set of hierarchies will be constructed for multiple views only; not for ontology construction purpose. Furthermore, all these ontology constructions are done in text domain; however, we address this problem in the image domain.

In our system, ontology serves as a taxonomy where similar images are grouped together [3, 16]. This similarity is not based only on color or shape but both along with finer grain (i.e., individual object) rather than coarser grain (i.e., entire image). For example, a football game image may contain green field, goalpost, and football objects. An image containing only a football would be misclassified as a football game based on color similarity analysis. On the other hand, shape similarity may also misclassify image. Based on only shape similarity we may identify a basketball as a football. Therefore, neither color nor shape based similarity is adequate to classify images. We need to combine these two similarities together to understand semantic meaning of images. Therefore, to classify images effectively, we need a knowledgebase where color and shape features of each category will be maintained. In our case ontology serves as a knowledgebase; it contains a set of concepts where each concept corresponds a category. And each category contains a set of images by sharing a set of similar objects. Here image similarity is determined by object similarity based on the combination of color and shape. One may argue that our ontology generation is based on clustering aspect of the problem. However, this clustering groups similar image based on semantic meanings. Thus, concept-based clustering technique has been employed.

3. Ontologies

An ontology is a specification of an abstract, simplified view of the world that we wish to represent for some purpose. Therefore, an ontology defines a set of representational terms that we call *concepts*. Inter-relationships among these concepts describe a target world. An ontology can be constructed in two ways, domain dependent and generic. CYC, WordNet, and Sensus are examples of generic ontologies. For our purposes, we choose a domain-dependent ontology. A domain-dependent ontology provides concepts in a fine grain, while generic ontologies provide concepts in coarser grain. The fine-grained concepts allow us to determine specific relationships among features in images that may be used to effectively classify those images.

Figure 1 illustrates an example ontology for the sports domain [5]. This ontology may be obtained from generic sports terminology and domain experts. The ontology is described by a directed acyclic graph (DAG). Here, each node in the DAG represents a concept. In general, each concept in the ontology contains a label name and feature vector. A feature vector is simply a set of features and their weights. Each feature may represent an object of an image, such as a basketball, a goalpost or a baseball. Note also that this label name connected to the feature is unique in the ontology. Furthermore, this label name is used to serve as an association of concepts to images. The concept of football may be further expanded to objects present in a football game (i.e. the features of the concept). For instance, a green field, goalposts, and football players would indicate the image is a football game. Should only one or two of the features common to a football game (as specified in the ontology) be present, a less specific classification of the image would be given. In other words, a more generic concept will be assigned to the image. An image containing only a football would be classified as an image containing a football, not as a football game. Furthermore, the weight of each feature of a concept may not be equal. In other words, for a particular concept some feature may serve as more discriminating as compared to some other; it will be assigned higher weight. For example, in the concept of a game of football the weight of goalpost feature is higher than the weight of the feature, green field.

In ontologies, concepts are interconnected by means of inter-relationships. If there is a inter-relationship R, between concepts C_i and C_j, then there is also a inter-relationship R' between concepts C_j and C_i. In Figure 1, inter-relationships are represented by labeled arcs/links. Three kinds of inter-relationships are used to create our ontology: IS-A, Instance-of, and Part-of. These correspond to key abstraction primitives in object-based and semantic data models [1].

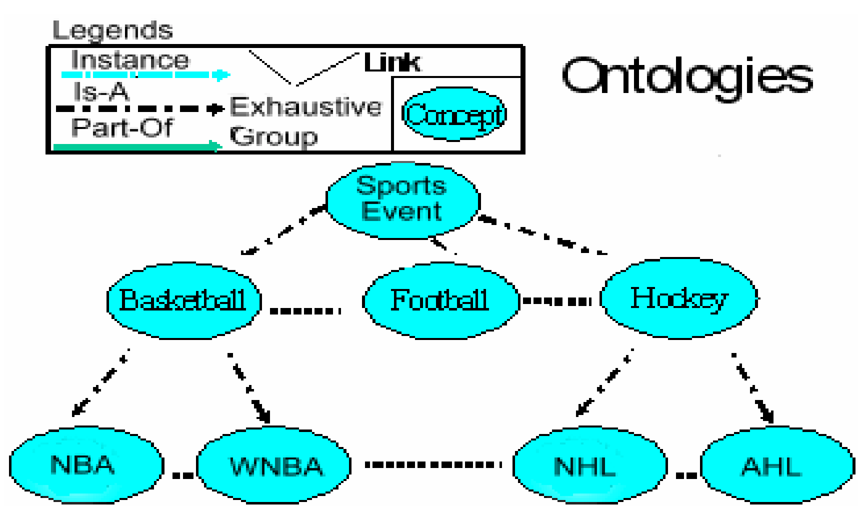

Figure 1: A Portion of an ontology for the Sports Domain

4. Proposed System

Our system circumvents the low precision classification techniques of other systems by examining the actual objects within an image and using them to discover relationships that reveal information useful in classifying the entire image. The concepts behind these relationships are held in our knowledge base of domain-dependant ontologies as described in section 3. We now outline the steps taken to successfully process and classify an input image presented to our system.

To convert objects to meaning or automatic ontology construction, we need to identify all object boundaries accurately (box 1 in Figure 1) that appear in images [3]. In this earlier work [3] we propose an automatic scalable object boundary detection algorithm based on edge detection and region growing techniques. Our algorithm works in three stages. First, we detect all edge pixels in images and divide pixels into two sets, edge pixel and region pixel sets. Second, we grow a region from the region pixel set surrounded by edges taken from the edge pixel set. Finally, we may merge adjacent regions using an adjacency graph to avoid over segmentation of regions and to detect boundary of objects accurately.

After detection of object boundaries, we would like to build ontologies automatically (box 2 in Figure 2). Our method constructs ontologies automatically in bottom up fashion. For this, we first construct a hierarchy using some clustering algorithms. Recall that if documents are similar to each other in content they will be associated with the same concept in ontology. After objects detection, for each object, we extract color and shape information and express them using vectors. Then considering both color and shape factors, we can calculate the similarity between objects applying vector space model. In ontologies each concept is described by a set of features (objects). Thus, before image clustering, we cluster objects according to similarities between objects and assign a weight for each object cluster. Next, we construct a vector for each images based on weights of object clusters and calculate similarities between images using vector space model. Finally, based on image similarities, we cluster images and build ontology hierarchy using hierarchy agglomerative clustering algorithm. Here a set of images will be used for construction of hierarchy. Thus, after construction of hierarchy, all semantically similar images based on object similarity will be grouped together. Furthermore, to classify a query image, first we segment images into objects based on our object boundary detection algorithm. Next, we determine similarity between objects that appear in this query image and objects that appear in concept's centroid image using vector space model (box 3 in Figure 2) , and choose the most similar one [16].

After the objects have been identified, their identifications are fed into a concept selection module (box 4 in Figure 2). The ontologies use this information to provide a meaningful description of the image by selecting concepts based on image content (i.e., individual objects within the image). In our earlier work [15] concept selection mechanism includes a novel, scalable disambiguation algorithm using a domain specific ontology. This algorithm will prune irrelevant concepts while allowing relevant concepts to become associated with images. For example, it is possible an image may be classified as both an NBA basketball game and a college basketball game at the same time. However, we employ the heuristic-based pruning techniques to narrow down the selection of concepts. When the pruning algorithm completes the selected concepts will be sorted based on their ranking in descending order. The concept label will then tell which category the image belongs to.

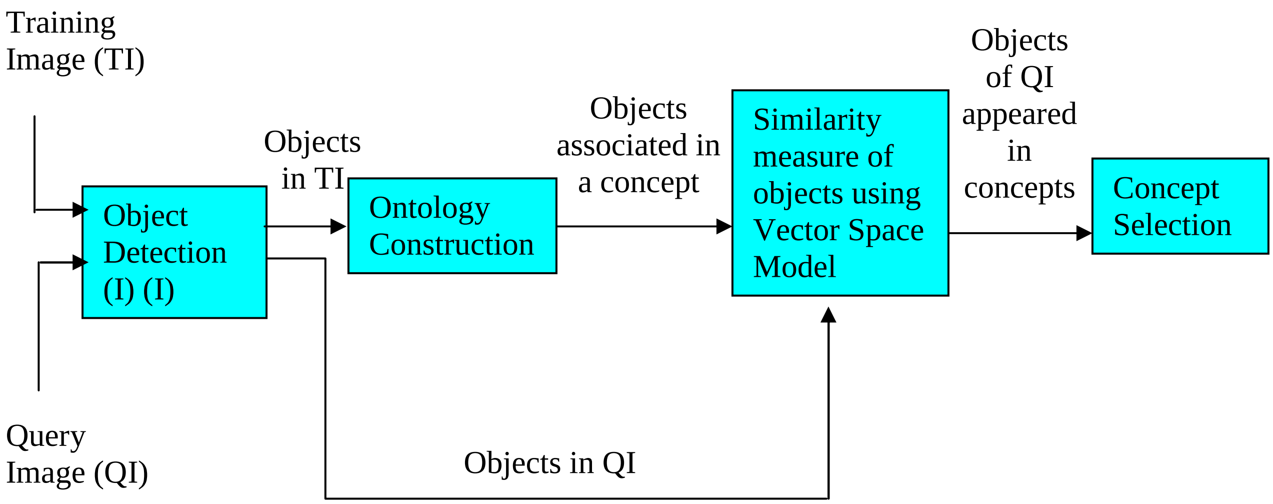

Figure 2. Flow of Our System

5. Ontology Derivation

Our goal is to construct ontologies automatically. For this, we would like to build a hierarchy from a set of images in a bottom up fashion. Note that we do not want to cluster images based on the similarity of color. Hence, similar images will be clustered if they share similar objects. Furthermore, some objects may carry more weight as compared to other objects that appear in images, similar to the way keywords behave in documents. For this, we need to assign weight to objects that appear in images. To determine the weight of objects we first cluster objects based on similarity of color and shape. Next, we determine weights of objects that appear in images based on term frequency and inverse document frequency, similar to IR. Each image will then be represented by a vector, where a vector contains a set of weights that correspond to the importance of the objects that appear in the image. Using this vector we will construct an image hierarchy using agglomerative clustering. We will discuss each stage elaborately in the following sections:

5.1 Object Clustering

By applying segmentation techniques, we segment images into objects. Let us assume we have N images in a database. After segmentation, we detect M number of objects in total. Then we cluster these M objects into a set of groups (say t) C_1, C_2, …. C_t according to similarities between objects. Here, object similarity will be based on the combination of color similarity and shape similarity of individual objects. This is because if visual features of objects such as color and shape are similar it is very possible that these objects have similar semantic meanings. Thus, we first introduce our color similarity measure, next our shape similarity measure, and then the combination of both.

5.1.1 Color Similarity Measure

To compute color similarity, we first construct a vector for an object consisting of a set of values of histogram bins. For each histogram bin (color code), we determine how many pixels of this particular object appear in the histogram bin. Thus, for object i a vector Vi ($v_{1,i}$, $v_{2,i}$, ….. $v_{p,i}$,…$v_{k,i}$) will be constructed to express the color histogram. In the vector each element represents the percentage of pixels whose hue value locates in specific interval. For example, $v_{p,i}$ is the percentage of pixels whose hue values are between $2 * \pi * p / k$ and $2 * \pi * (p + 1) / k$ in object i, because the range of hue value in HSI color space is from 0 to $2 * \pi$. Furthermore, we only consider hue component for similarity measure which is adequate. Now, we can determine the degree of color similarity (*$sim_c(i, j)$*) between object i and object j based on cosine product. Thus,

$$sim_c(i, j) = \frac{\vec{i}_c \bullet \vec{j}_c}{\left|\vec{i}_c\right| \times \left|\vec{j}_c\right|} = \frac{\sum_{p=1}^{k} v_{p,i} \times v_{p,j}}{\sqrt{\sum_{p=1}^{k} v_{p,i}^2} \times \sqrt{\sum_{p=1}^{k} v_{p,j}^2}} \tag{1}$$

The value of k affects the accuracy of color similarity. Along with increasing of k, accuracy will be also increased. However, it will be computationally expensive.

5.1.2 Shape Similarity Measure

The shape similarity is little bit more complicated. To support similarity queries, Lu et al. [14] introduce fixed resolution (FR) representation method. We adopt this idea, but make some change during implementation. First, we need to find the major axis that is the longest line joining two points on the boundary. For normalization purpose we rotate an angle θ around mass centroid of object to make the major axis to be parallel to x-axis and keep the centroid above the major axis. The reason for normalization is to make it invariant to rotation [13]. The coordinates of the centroid are as follows: σ(x,y) is surface density function.

$$\bar{x} = \frac{\iint x\sigma(x, y)\, dA}{M} \qquad \bar{y} = \frac{\iint y\sigma(x, y)\, dA}{M}. \tag{2}$$

After normalization, we divide the object into q * q grid, which is just big enough to cover the entire object, and overlaid on the object where q is an integer number. The size of each cell is same. Then we define a shape vector for object i, Ui ($u_{1,i}$, $u_{2,i}$, ….. $u_{p,i}$,…$u_{q2,i}$) of q^2 size. Each element in the vector stands for the percentage of pixels in the corresponding cell. The higher the q value, the higher the accuracy. Of course, it will then be computationally expensive. Finally, we determine shape similarity (*$sim_s(i, j)$*) between two objects i and j using cosine similarity. Thus,

$$sim_s(i, j) = \frac{\vec{i}_s \bullet \vec{j}_s}{\left|\vec{i}_s\right| \times \left|\vec{j}_s\right|} = \frac{\sum_{p=1}^{q^2} u_{p,i} \times u_{p,j}}{\sqrt{\sum_{p=1}^{q^2} u_{p,i}^2} \times \sqrt{\sum_{p=1}^{q^2} u_{p,j}^2}} \tag{3}$$

5.1.3 Combined Similarity

Now, we would like to determine similarity between two objects based on color similarity and shape similarity. Using Equation 2 and 3, similarity (*$sim(i, j)$*) between object i and object j is as follows:

$$\begin{aligned} &sim(i, j) = sim_c(i, j) \times weight_c + sim_s(i, j) \times weight_s \\ &weight_c + weight_s = 1 \end{aligned} \tag{4}$$

$Weight_c$ is the weight of color and $weight_s$ is the weight of shape. When it is possible that one type of similarity may be more important as compared to another we need to use weight. In the current case, we assume that both $weight_c$ and $weight_s$ are equally important (=0.5).

To construct object clusters we use a threshold T_{obj}. If similarity between two objects is greater than T_{obj}, then the two objects can be in the same group. In other words, similarity between each pair of objects in the same group must be greater than T_{obj}. Furthermore, it is possible for an object to appear in more than one group. It is important to note that a group consisting of a set of objects corresponds to a keyword as opposed to each individual object corresponding to a keyword. This is because in IR match is well defined; in multimedia, match is ill defined (similarity).

5.2 Vector Model for Images

Image clustering is based on image similarity. To calculate image similarity, we construct an image, l vector $W_l\,(w_{1,l}, w_{2,l} \ldots w_{i,l}, \ldots w_{t,l})$ rather than measuring similarity based on color. $W_{i,l}$ is the weight of object cluster C_i in image *l*. In this vector we keep the weight of each group. Thus, the size of vector W is same as the total number of object clusters (=t). It is possible that the weight of a group may be zero. This is because no object of an image may be a participant of that group during clustering.

To determine an image vector, we adopt the idea from the area of information retrieval [39]. Here, images correspond to documents, and object clusters correspond to terms (keywords). Let N be the total number of images and n_i be the number of images in which objects of cluster C_i appear. We define the normalized term frequency $f_{i,j}$:

$$f_{i,l} = \frac{freq_{i,l}}{\max_h \; freq_{h,l}} \tag{5}$$

The $freq_{i,l}$ is the number of times cluster C_i appears in the image *l*. Similarly, for max $freq_{h,l}$ we determine occurrence of a cluster that appears in image, l; for this cluster we get maximum occurrence among all clusters.

We also define an inverse document frequency idf_i for C_i :

$$idf_i = \log \frac{N}{n_i} \tag{6}$$

Considering the above two factors, we have the weight of cluster:

$$w_{i,l} = f_{i,l} \times idf_i = f_{i,l} \times \log \frac{N}{n_i} \tag{7}$$

After computing image vectors, we get similarity between any two images using cosine similarity:

$$sim_{img}(i, j) = \frac{\vec{i} \bullet \vec{j}}{|\vec{i}| \times |\vec{j}|} = \frac{\sum_{h=1}^{t} w_{h,i} \times w_{h,j}}{\sqrt{\sum_{h=1}^{t} w_{h,i}^2} \times \sqrt{\sum_{h=1}^{t} w_{h,j}^2}} \tag{8}$$

5.3 Hierarchical Clustering of Images

Using the above method, we can calculate image similarity between each pair of images. Then we apply a hierarchical agglomerative clustering algorithm (HAC) to construct hierarchy. The HAC algorithm is a commonly employed classical hierarchal clustering algorithm. The result of HAC is a dendrogram representing the nested grouping of images. The general HAC algorithm is as follows:

1) Put each image into a singleton cluster, compute a list of inter cluster distance for all singleton clusters, then sort the list in ascending order.

2) Find the pair of clusters with the most similar, merge them into one cluster and calculate the similarity between the new cluster and the remaining clusters.

3) While there is more than one cluster remaining, go to step 2, otherwise stop.

Based on the calculation of similarity between the non-singletons clusters a variety of hierarchical agglomerative techniques have been proposed. Single-link, complete-link and group-average-link clustering are commonly used. In the single-link cluster the similarity between two clusters is the maximum similarity of all pairs of documents which are in different clusters. In the complete-link cluster, the similarity between two clusters is the minimum similarity of all pairs of documents which are in different clusters. In Group-Average-link clustering the similarity between two clusters is the mean similarity of all pairs of singletons which are in different cluster.

In the hierarchy, a node will be represented by a representative image which is most similar to the node vector. It is important to note that various types of inter-relationships between nodes are blurred in our ontologies; certain types of interconnections are ignored. This is because our prime concern is to facilitate information selection rather than to deduct new knowledge.

6. Experimental Preliminary Results

The purpose of the experiment is to test the accuracy of the clustering. We have used a set of images belonging to 6 different categories, such as basketball game, baseball game, bats, football game, goggle and playground. We have then constructed hierarchy based on the theory discussed in Section 5. We have chosen Group-Average-Link clustering in HAC. We have set k=12 (see Equation 1), and T_{obj} =0.7.

To measure the quality of a cluster, we use precision, recall, and E measure [33]. *Recall* is the ratio of relevant images to total images for a given category. *Precision* is the ratio of relevant images to images that appear in a cluster for a given category. E measure is defined as follows:

$$E(p,r) = 1 - \frac{2}{1/p + 1/r} \qquad (9)$$

Where *p* and *r* are the *Precision* and *Recall* of a cluster. Note that E (p, r) is simply one minus harmonic mean of the precision and recall; E (p, r) ranges from 0 to 1 where E (p, r) =0 corresponds to perfect precision and recall, and E (p, r) corresponds to zero precision and recall. Thus, the smaller the E measure values the better the quality of a cluster.

In Figure 3 X axis represents different categories and the Y axis represents p, r and E. We have observed that precision and recall are higher for simple images (e.g., play ground) as compared to precision and recall of complex images (e.g., baseball game, football game). Thus, E value is lower for simple images (e.g., play ground) as compared to E value of complex images (e.g., baseball game, football game).

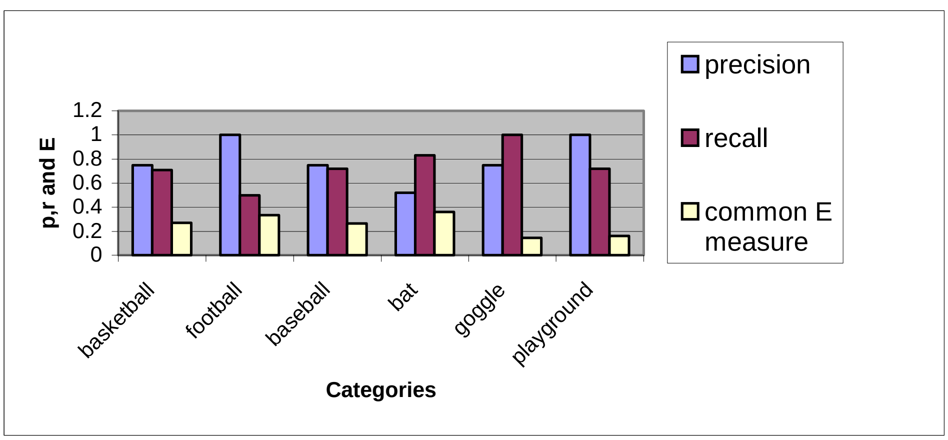

Figure 3. Cluster Quality for Different Categories

7. Conclusion and Future Work

In this paper we have proposed a potentially powerful and novel approach for the automatic construction of ontology. The crux of our innovation is the development of a hierarchy based on object similarity using a vector space model. Furthermore, to determine object similarity we have combined both color similarity and shape similarity. To illustrate the effectiveness of our algorithm in automatic image classification, we implement a very basic system aimed at the classification of images in the sports domain. For developing a hierarchy, we have used an agglomerative clustering algorithm that constructs hierarchies from bottom to up. We would like to extend this work in the following directions. First, we would like to do more experimentation with the clustering techniques. Next, we would like to address this kind of ontology construction in various domains.

References

[1] G. Aslan and D. McLeod, "Semantic Heterogeneity Resolution in Federated Database by Metadata Implantation and Stepwise Evolution", *The International Journal on Very Large Databases*, Vol. 18, No. 2, October 1999.

[2] R. Barber, W. Equitz, C. Faloutsos, M. Fickner, W. Niblack, D. Petkovic, and P. Yanker, "Query by Content for Large On-Line Image Collections", *IEEE Journal*, 1995.

[3] Lei Wang, Latifur Khan, and Casey Breen, Object Boundary Detection for Ontology-based Image Classification, *Third International Workshop on Multimedia Data Mining*, Edmonton, Alberta, Canada, July 2002.

[4] R. Bodner and F. Song, "Knowledge-based Approaches to Query Expansion in Information Retrieval," in Proc. of Advances in Artificial Intelligence, pp. 146-158, New York, Springer.

[5] ESPN CLASSIC, http://www.classicsports.com.

[6] Dave Elliman, J. Rafael G. Pulido. "Automatic Derivation of On-line Document Ontology". International Workshop on Mechanisms for Enterprise Integration: From Objects to Ontology (MERIT 2001) 15th European Conference on Object Oriented Programming, Budapest, Hungary, Jun 2001.

[7] A. Hotho, A. Mädche, A., S. Staab, "Ontology-based Text Clustering," Workshop Text Learning: Beyond Supervision, 2001.

[8] A. Pentland, R.W. Picard, S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases", in Proc. of *Storage and Retrieval for Image and Video Databases II*, Volume 2185, pp. 34-47, Bellingham, WA, 1994.

[9] N. Row, and B. Frew, "Automatic Classification of Objects in Captioned Depictive Photographs for Retrieval", Intelligent Multimedia Information Retrieval, Chapter 7, M. Maybury, AAAI Press, 1997.

[10] A. F. Smeaton and A. Quigley, "Experiments on Using Semantic Distances between Words in Image Caption Retrieval," in Proc. of *The Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.

[11] C.J. van Rijsbergen. Information Retrieval. Butterworths, London, 1979.

[12] C. Frankel, M.J. Swain and V. Athitsos, "WebSeer: An Image Search Engine for the World Wide Web," *University of Chicago Technical Report TR-96-14*, July 31, 1996.

[13] Chakrabarti, K., Ortega-Binderberger, M., Porkaew, K & Mehrotra, S. (2000) Similar shape retrieval in MARS. Proceeding of IEEE International Conference on Multimedia and Expo.

[14] G. Lu and A. Sajjanhar, Region-based shape representation and similarity measure suitable for content-based image retrieval. Springer Verlag Multimedia Systems, 1999.

[15] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval, ISBN 0-201-39829-X, 1999.

[16] C. Breen, L. Khan, A. Ponnusamy, and L. Wang, "Ontology-based Image Classification Using Neural Networks," *Proc. of SPIE Internet Multimedia Management Systems III*, pp. 198-208, Boston, MA, July, 2002.