A Visual Annotation Framework Using Common-Sensical and Linguistic Relationships for Semantic Media Retrieval

Bageshree Shevade[1], Hari Sundaram[1]

[1] Arizona State University
{Bageshree.Shevade, Hari.Sundaram}@asu.edu

Abstract. In this paper, we present a novel image annotation approach with an emphasis on – (a) common sense based semantic propagation, (b) visual annotation interfaces and (c) novel evaluation schemes.

The annotation system is interactive, intuitive and real-time. We attempt to propagate semantics of the annotations, by using WordNet and ConceptNet, and low-level features extracted from the images. We introduce novel semantic dissimilarity measures, and propagation frameworks. We develop a novel visual annotation interface that allows a user to group images by creating *visual* concepts using direct manipulation metaphors without manual annotation. We also develop a new evaluation technique for annotation that is based on relationship between concepts based on commonsensical relationships. Our Experimental results on three different datasets, indicate that the annotation system performs very well. The semantic propagation results are good – we converge close to the semantics of the image by annotating a small number (~16.8%) of database images.

1. Introduction

In our work, annotation is considered to be the meta-data or the keyword description that is given to images. For example – if you have an image of a cat, then the keywords *"cat"*, or *"animal"* etc. that users would give to describe that image is considered to be annotation. The goal of our work is to create novel semi-automated, intelligent annotation algorithms that bridge manual methods for annotation and fully automatic techniques. Though fully automatic techniques based on pattern recognition and content based retrieval are efficient, they are not very accurate. On the other hand, manual techniques though accurate are very tedious and time consuming.

There has been prior work in semi-automatic image annotation using relevance feedback [3,4,9,11,13]. While there are rich mathematical models used, we believe that there are three shortcomings of the current work:

- **Semantics**: Current approaches to annotate images [3,4,11,13] essentially treat words as symbols regardless of their semantic relationships with other words, which is no different than any normal image feature. The lexical meaning of the keywords/annotations is not exploited.

- **Intuitive Interfaces**: There are number of tools and drag and drop interfaces [7,11] for annotating images. However, these tools do not allow users to group images based on *visual* concepts without the use of manual annotations. Current annotation tools also lack any kind of label propagation.

- **Novel evaluation schemes**: Presently, all CBIR systems[3,4,9,13] predominantly use the Precision-Recall metric, to evaluate their system. However, this measure does not take into account the semantic relationship (e.g. though linguistic ontologies) among words. Hence, there is a need to develop new evaluation techniques that incorporate semantics.

In this work we address issues relating to both semantics and visual annotation interfaces. We establish the semantic inter-relationships amongst the annotations by using WordNet [8] and ConceptNet[5]. We also develop an intuitive visual annotation interface.

The annotation procedure is as follows. Our annotation system uses a combination of low-level features such as color, texture and edge as well as WordNet synsets and ConceptNet distances to propagate semantics. As the user begins to annotate the images, the system creates positive example (or negative examples) image sets for the associated WordNet meanings. These are then propagated to the entire database, using low-level features as well as ConceptNet distances. The system then determines the image that is least likely to have been annotated correctly and presents the image to the user for relevance feedback.

Our annotation interface allows users to group images by creating *visual* concepts without requiring text annotation. A visual concept is an abstract idea that the user associates with an image. For example – the user could associate the concept of *"garden"* to a group of semantically related images depicting flowers. The user can then add positive (or negative) examples to these visual concepts. The system then creates visual clusters based on low-level features. The user can also add text annotations to these visual concepts. The system then propagates these annotations to all the visual concept clusters based on low-level features and WordNet.

The annotation system is evaluated using a novel evaluation scheme that is not based on the Precision-Recall metric. Our system uses ConceptNet similarity measure, to determine the performance of our annotation algorithm. This is done since the Precision-Recall measure does not incorporate the semantic relationship between words. Traditional precision-recall measures are useful in the context of classification. However, in annotation, the semantics of the annotation words are important. For example – if the system associates the word *"apartment"* to an image that is annotated as *"house"*, then it is not such a large error. We therefore, use semantic relationship between words to evaluate the accuracy of the annotation algorithm. Our results indicate that the system performs very well and is much better than the baseline case of binary concept membership.

The rest of this paper is organized as follows. In the next section, we discuss the features used in our system. In section 3, we present briefly the semantic propagation algorithm. In Section 4 we present a visual annotation interface while in Section 5 we present the experimental results. And finally, we present our conclusions in Section 6.

2. Features

In this section, we shall describe the low-level features as well as the semantic features used in our annotation system.

2.1 Media Features

In our work, the feature vector for images comprises of color, texture and edge histograms. The color histogram consists of 166 bins in HSV space. The HSV space is used, as it is perceptually continuous. The system extracts Tamura texture [6] from images. The texture histogram consists of 3 bins corresponding to contrast, coarseness and directionality of the image. The edge histogram [6] consists of 71 bins that incorporates curvature and edge directionality. We then concatenate the three histograms to get a final composite histogram of 240 bins. The low level feature distance between two images *i* and *j* is then given as:

$$d(i,j) = \sqrt{\sum_{k=1}^{N} (h_i^k - h_j^k)^2}, \qquad <1>$$

where N is the total number of bins, and h_i and h_j are the corresponding bins of images *i* and *j*.

2.2 Media Semantics

In our prior work[2], semantics were incorporated through the use of WordNet ontology, which is an online lexical database[8]. WordNet organizes the English nouns, verbs and adjectives into synonym sets (synsets), which represent a unique lexical concept. A given English word can belong to multiple synsets and conversely, each synset has multiple words or word forms, which are synonyms of each other. WordNet supports different relationships between synsets like hypernym/hyponym, synonym/antonym, meronym/holonym etc. In our prior work[12], we exploited the hypernym/hyponym relationship between synsets to compute the implication distance measure between two synsets.

In this work, semantics are incorporated through the use of ConceptNet [5]. ConceptNet is a large repository of common-sense knowledge. ConceptNet supports 20 different semantic relationships between concepts like “capableOf”, “effectOf”, “userFor”, “isA”, “partOf” etc. We use ConceptNet, as it captures a very rich set of semantic relationships as compared to WordNet. This leads to a very uniform distribution of implication distance measures between two synsets, when computed using Conceptnet instead of WordNet [10].

Semantic Distance. In this subsection, we shall describe the procedure to compute the implication distance measure between two synsets s_i and s_j using ConceptNet. A WordNet synset has multiple synonym words associated with it. Let us assume that synset s_i is associated with words $w_{i,1}$, $w_{i,2}$ and $w_{i,3}$; and synset s_j is associated with words $w_{j,1}$, $w_{j,2}$ and $w_{j,3}$. Let us also assume that $d_c(w_1,w_2)$ denotes the ConceptNet distance between two words (concepts) [1]. The distance between two synsets s_i and s_j is then given as:

$$d(s_i, s_j) = \max_q \min_k d_c(w_{i,q}, w_{j,k}), \quad <2>$$

where $w_{i,q}$ is the word associated with synset s_i and $w_{j,k}$ is the word associated with synset s_j. The distance $d(s_i,s_j)$ is the hausdorff distance between two sets containing synonym words. The implication measure between two synsets s_i and s_j is given as:

$$I(s_i, s_j) = 1 - d(s_i, s_j), \quad <3>$$

where $d(s_i,s_j)$ is as defined in equation <2>.

3. Propagation of Semantics

In this section, we present the details of our algorithm on semantic propagation. Let us assume that the user wishes to annotate an image *a* and that there are N images in the database. When the user enters annotations for image *a*, she is asked to fix the sense (i.e. the semantics) of the word that she is using for annotation using WordNet. For example, if she annotates an image with the word "suit", then she fixes the sense to be either a "lawsuit" or "clothing" or "pack of cards" etc. Fixing the sense of the word exploits the hierarchical relationship among synsets in WordNet. The current image *a* is considered as a positive example image for this synset.

3.1 Algorithm Details

We shall now discuss three key aspects of the algorithm – (a) local trees, which identify the semantic space of the fixed synset and help in computing semantic distance between synsets, (b) propagation of the synsets across the database and (c) providing feedback by presenting the user with an image that will maximize the rate of semantic propagation.

Local Trees. For each fixed synset *k*, we need to define a local tree T_k. A local tree is a subset of the WordNet ontology. It is a hierarchy of nodes, with synset *k* being the node at the center of the hierarchy. Synset *k* is also called the root node of the tree. Nodes above the center are the generalizations of synset *k* and nodes below it are its specializations. Formally, a local tree T_k is defined as follows:

$$T_k = \{s \mid d(s,k) \le m\}, \quad <4>$$

where $d(s,k)$ is the hop distance between the node representing synset *s* and synset *k*, and *m* is the number of specialization and generalization levels to be considered. In

our case, we set m = 2, based on a trade off between computational complexity and accuracy.
Thus a local tree efficiently partitions the semantic space of WordNet – it helps to quickly identify if two synsets are semantically far apart.

Computing semantic propagation likelihoods. The new synset k, entered by the user is then propagated to other images based on low-level features using color, texture and edge histograms, WordNet local trees and ConceptNet implication measure [ref. Section 2.2]. This is done as follows: we determine $L_f(k|i)$, that is the low-level feature likelihood that image i belongs to synset k.

$$L_f(k \mid i) = L_f(k^+ \mid i) - L_f(k^- \mid i), \quad <5>$$

where, k^+ denotes the set of all the positive example images directly associated with synset k, as well as the positive example images associated with the synsets present in the local tree of k. k^- denotes the set of all the negative example images of synset k, as well as all the positive example images of the synsets, not present in the local tree of k. $L_f(k^+|i)$ denotes the low-level likelihood that the image i belongs to the set k^+. $L_f(k^-|i)$ denotes the low-level likelihood that the image i belongs to the set k^-. $L_f(k^+|i)$ is defined as follows:

$$L_f(k^+ \mid i) = \sum_{j=1}^{P} \exp(-\beta d_{ij}) I(k, s_j), \;\; s_j \in T_k, \quad <6>$$

where P is the total number of synsets in T_k (local tree of synset k) [ref. Section 3.1], whose positive examples are being considered. d_{ij} is the average low-level feature distance between image i and the positive examples of synset s_j. β is a constant and $I(k,s_j)$ is the implication between synsets k and s_j using ConceptNet [ref Section 2.2]. $L_f(k^-|i)$, denotes the low-level likelihood that the image i belongs to set k^- and is given as follows:

$$L_f(k^- \mid i) = w_1 \exp(-\beta d_{ik}) + \frac{w_2}{N} \sum_{j=1}^{Q} L_f(s_j^+ \mid i), \; s_j \notin T_k, \quad <7>$$

where β, is a constant, d_{ik} is the average low-level feature distance between image i and the negative examples of synset k. w_1 and w_2 are weights where $w_1+w_2=1$ and Q is the total number of synsets not in the local tree of synset k.

We now show how to compute the ConceptNet likelihood. This is done by calculating the likelihood of other synsets already present in the database (but not manually associated with the image) to image i. These other synsets are present in the database, as the user has entered them as manual annotations for some other images in the database. This likelihood is the ConceptNet likelihood and is given as follows:

$$C_l(s_1, s_2, \ldots s_M \mid k) = \frac{1}{M} \sum_{j=1}^{M} I(s_j, k), \quad <8>$$
$$L_c(k \mid i) = C_l(s_1, s_2, \ldots s_M \mid k)$$

where $s_1, s_2 \ldots s_M$ are the synsets which are manually associated with the image i by the user. $I(s_j,k)$ is the ConceptNet implication between manual synset s_j and synset k which we want to propagate and M is the total number of synsets manually associated with image i.

Feedback. The system now presents an image to the user for relevance feedback. This is an image that is least likely to be associated with the annotations that accurately reflect the semantics of the image. The final likelihood, for picking the least likely image, for an image *i* is computed as follows:

$$l_i = \frac{1}{M}\sum_{j=1}^{M} \alpha L_f(s_j \mid i) + \beta L_c(s_j \mid i), \quad <9>$$

where M is the number of synsets which were automatically propagated to image *i*. $L_f(s_j|i)$ is the low-level feature likelihood of image *i* with respect to synset s_j and $L_c(s_j|i)$ is the ConceptNet likelihood of image *i* with respect to synset s_j. α and β are constants where α+β=1. The least likely image j* is picked as follows:

$$j^* = \arg\min_j l_j, \quad <10>$$

where *j* varies over the total number of images in the database, and l_j is the final likelihood of image *j*.

During relevance feedback, the user can either delete an associated synset, or confirm an automatically propagated synset or add another synset. If the user deletes an associated synset, then we mark the image as a negative example of the synset. We then recompute the distance of every other image in the database, with respect to the newly updated positive and negative example sets of that synset. On the other hand, if the user confirms an automatically associated synset, then we consider this image as a positive example of the synset, and follow the same procedure of recomputing the distance of every other image, as explained earlier. Same holds true if the user adds a synset to the image. Thus with each iteration of relevance feedback the association between images and synsets gets refined, and becomes more accurate with respect to the image semantics.

In this section, we have discussed our annotation algorithm in detail. The algorithm focuses on (a) semantic propagation using ConceptNet and WordNet and (b) maximizing the rate of propagation by providing relevance feedback for the least likely image. We now describe a front end that uses the algorithm as the computational backbone and allows the user to annotate media.

4. Visual Annotation Interface

In this section we shall discuss the visualization interface that allows users to group images based on concepts without manual annotations. Here a *concept* is an abstract idea that the user associates with an image. For example the user can associate a concept of "peace" with the image of a dove.

Our user interface is shown in Figure 1. The left panel of the interface shows the images in the form of small circles, colored with the dominant (mean) color of the image. The bottom pane shows the various concepts created in the system and their positive examples. As the user moves the mouse over the image circles in the left pane, the right pane shows the details viz. the image, the concept images of which the image is a positive or a negative example, and the annotations of the image. Note that

this interface is in its preliminary stage and we plan to conduct extensive evaluation of the various aspects of the interface in the future.

4.1 Creating Visual Concepts

Initially all the images in the form of circles are scattered at random in the interface. Now the user can click on one of these image circles and create a concept in the system. The clicked image is considered to be a positive example of the concept and is treated as an anchor point for the concept. Every concept is represented by a unique color in the system. All image circles representing images belonging to the same concept are colored with the same color to indicate semantic closeness.

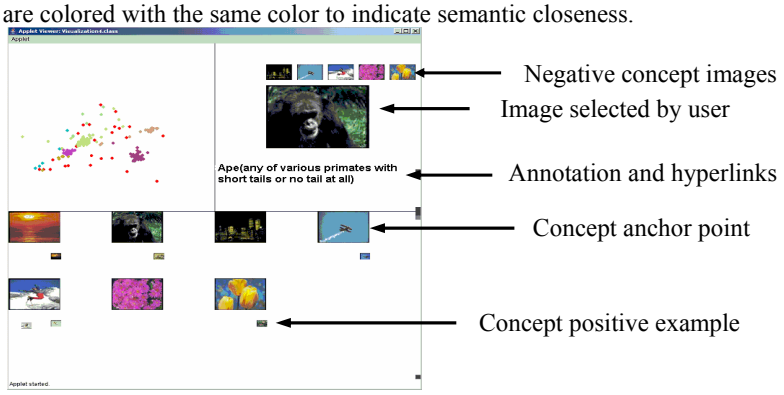

Figure 1: Visual Annotation Interface

Visual feedback. After the creation of the first concept, all the other unlabeled images in the database viz. images that are neither positive nor negative examples of any concept, move towards the anchor point of this concept. This movement is based on their color histogram distance with respect to the positive (and negative) examples of the concept. However, if the color histogram distance is above a threshold; the image movement is restricted to be δ of the current display distance between the image and the anchor point. This is done to avoid cluttering and to prevent all images getting merged into a single image circle on screen. Thus a concept cluster is formed with the anchor point being the center of the cluster.

Creating positive visual clusters. The user can then add positive examples to the concept by dragging the image circles on the concept anchor point in the bottom pane. If the color histogram distance between the dragged image and the positive example images of the concept is within a certain threshold, the image circle will move towards an already established concept cluster, else it will create another cluster for the same concept by being another anchor point. Thus you can have multiple clusters of the same concept which are visually at a larger display distance but semantically close. For example, if the user first created a concept anchor point with an image of a red flower, and then associated the image of a yellow flower with it then

they would visually appear at different points in the pane, but the color of the image circles would still be same(since they belong to the same *concept*). Also, all the unlabeled images will move towards the closest cluster of the closest concept by the same mechanism as above. The closest concept c^* is then defined as the one that maximizes the difference between positive and negative likelihood for the unlabelled image.

$$c^* = \arg\max_c diff_c, \qquad \text{<11>}$$
$$diff_c = L_f(c^+ \mid i)(1 - L_f(c^- \mid i)),$$

where c^+ denotes the set of positive examples associated with visual concept c and c^- denotes the set of negative examples associated with concept c. $L_f(c^+|i)$ is the positive feature likelihood of the unlabelled image i with respect to the set c^+ and $L_f(c^-|i)$ is the negative feature likelihood of the unlabelled image i with respect to the set c^-. $L_f(c^+|i)$ is then given as:

$$L_f(c^+ \mid i) = \frac{1}{M}\sum_{j=1}^{M} \exp(-\beta d_{ij}), \qquad \text{<12>}$$

where β is a constant, M is the total number of positive examples of the visual concept
c, and d_{ij} is the color histogram distance between images i and j. $L_f(c^-|i)$ is given as:

$$L_f(c^- \mid i) = \frac{1}{N}\sum_{k=1}^{N} \exp(-\beta d_{ik}), \qquad \text{<13>}$$

where β is a constant, N is the total number of negative examples of the visual concept c and d_{ik} is the color histogram distance between images i and k. The closest cluster within the closest concept c^* is then determined as the one that minimizes the feature distance between the image i and the positive examples of the cluster.

Creating negative visual clusters. The user can also specify negative examples for a visual concept. This creates a negative cluster for the concept and the image becomes a negative anchor point or moves towards the anchor point of an already created negative cluster of the concept. If the image becomes a negative anchor point then it moves away from the closest cluster of the concept by an amount equal to the average feature distance between the image and the positive examples of the cluster. If the average feature distance is above a certain threshold then the moving away is restricted to be δ of the current display distance so that the image circles don't go out of the display screen. Unlike multiple positive clusters per concept, there is only one negative cluster per concept. When an image becomes a negative example, it is duplicated on screen and is colored in red. So, all negative clusters of all visual concepts appear in red. Also, an image could be a positive or a negative example of more than one visual concept, in which case it is duplicated on screen.

4.2 Associating Annotations to Visual Concepts

The user can also add annotations to visual concepts by clicking on the concept anchor point in the bottom pane. As the user enters the annotations, she is asked to

pick the sense of the word using WordNet. These senses (synsets) are then propagated to other unannotated images in the system based on feature likelihood and WordNet likelihood. Let us assume that the user annotates the visual concept *c* with the synset *k*. The positive examples of the concept *c* then become the positive examples of synset *k* and negative examples of *c* become negative examples of *k*. The system then propagates synset *k* to all the unannotated images that had moved towards concept *c*, with a feature likelihood, $L_f(k|i)$ that is given as:

$$L_f(k \mid i) = L_f(c^+ \mid i) - L_f(c^- \mid i), \quad <14>$$

where $L_f(c^+|i)$ is the positive feature likelihood of image *i* with respect to the set c^+ and is given as:

$$L_f(c^+ \mid i) = \exp(-\beta \frac{1}{M} \sum_{j=1}^{M} d_{ij}), \quad <15>$$

where β is a constant and M is the number of positive examples associated with concept *c* and d_{ij} is the feature distance between images *i* and *j*. $L_f(c^-|i)$ is the negative feature likelihood of image *i* with respect to set c^- and is given as:

$$L_f(c^- \mid i) = \exp(-\beta \frac{1}{N} \sum_{j=1}^{N} d_{ik}), \quad <16>$$

where N is the total number of negative examples associated with concept *c*.

Propagating synset to other visual concepts. The system then propagates synset *k* to all the other images in the system that are neither positive, negative nor automatic examples of visual concept *c*. For every such image *i*, the system determines all the distinct visual concepts to which image *i* belongs. Let us denote one such visual concept as *v*. The system then determines the feature likelihood of synset *k* with respect to the visual concept *v*, $L_f(k|v)$ as:

$$L_f(k \mid v) = L_f(k^+ \mid v) - L_f(k^- \mid v), \quad <17>$$

where k^+ denotes the positive examples associated with synset *k*, and k^- denotes the negative examples of synset *k*. $L_f(k^+|v)$ denotes the feature likelihood of the visual concept *v* with respect to the set k^+ and is given as:

$$L_f(k^+ \mid v) = \sum_{j=1}^{M} \exp(-\beta d_{vj}) I(k, s_j), \quad s_j \in T_k, \quad <18>$$

where β is a constant and M is the total number of synsets in the local tree of *k*, i.e. T_k [ref. Section 3.1] whose positive examples are being considered and d_{vj} is the average feature distance between all the positive examples of visual concept *v* and positive examples of synset s_j. Similarly, the feature likelihood of the visual concept *v* with respect to the set k^- is given as:

$$L_f(k^- \mid v) = w_1 \exp(-\beta d_{vk}) + \frac{w_2}{N} \sum_{j=1}^{N} L_f(k^+ \mid v), \quad s_j \notin T_k, \quad <19>$$

where β is a constant and d_{vk} is the average feature distance between positive examples
of visual concept *v* and the negative examples of synset *k* and N is the total number of synsets not in the local tree of synset *k*. The feature likelihood of synset *k* with respect to image *i*, $L_f(k|i)$ is then given as:

$$L_f(k \mid i) = \frac{1}{M}\sum_{j=1}^{M} L_f(v_j \mid i) L_f(k \mid v_j), \quad <20>$$

where $L_f(v_j|i)$ is the feature likelihood that image i belongs to visual concept v and M is the total number of visual concepts to which image i belongs. The system then calculates the WordNet likelihood of other synsets already present in the database, but not manually associated with the positive examples of the visual concept c. These other synsets are present in the database, as the user has entered them as manual annotations for some other visual concepts in the database. This wordnet likelihood for an image i which is a positive example of concept c, is then given as:

$$W_l(s_1, s_2, \ldots s_M \mid k) = \frac{1}{M}\sum_{j=1}^{M} I(s_j, k), \quad <21>$$

$$L_w(k \mid i) = W_l(s_1, s_2, \ldots s_M \mid k)$$

where $s_1, s_2, \ldots s_M$ are synsets which are manually associated with image i. $I(s_j, k)$ is the Wordnet implication [12] between manual synset s_j and synset k which we want to propagate and M is the total number of synsets manually associated with image i.

In this section, we have described an interface that allows the user to create concepts, without entering annotations. It does not require the presence of high quality text (annotations) to organize images. Moreover, the user is only required to drag-and-drop in order to expand the positive and the negative image data set of a concept and thus organize images. This is much more intuitive and easier than entering text for every image. We shall now discuss our experimental results and evaluation scheme.

5. Experiments

The experiments were conducted on three different datasets – (a) a set of 242 images containing 90 distinct ground truth synsets, (b) a set of 500 personal image collection containing 107 distinct ground truth synsets and (c) and set of 1000 Corel image collection consisting of 44 distinct ground truth synsets. The ground truth for these image sets was created manually, by fixing the sense of the annotation. We refer to this fixed sense of the ground truth annotation as the ground truth synset. The entire prototype was developed in Visual J# in Microsoft Windows platform.

We now enumerate the steps taken to test our system:

- In order to test the system, we created an automatic test script that simulated a user annotating the images and providing relevance feedback. We picked a random image initially.
- We exposed the ground truth of this random image and annotated it with the ground truth synset. This is equivalent to the user picking an image and annotating it.
- After performing all the required propagations [ref. Section 3.1, equation<5> ,<8>], the system picked an image that was least likely to be associated accurately, with the semantics of its annotations, and uncovered its ground truth.

- If the automatic annotation synsets of this image coincide with the ground truth, then the system confirms the automatic annotations and updates the positive example images for that synset.
- However, if the automatic synsets and the ground truth synsets don't match, i.e. they are not present in each other's local tree [ref. Section 3.1], then the system considers the image as a negative example of the automatically propagated synsets, and a positive example of the ground truth synset. This is equivalent to the user giving relevance feedback for the least likely image, where he deletes the automatically propagated synsets, and adds the ground truth synsets to the image.
- Also, if the ground truth is present in the local tree of the automatically propagated synset, then the system treats the image as a weaker positive example of the automatically propagated synset and updates its likelihood with respect to the automatic synsets. This is intuitive because of the semantic relationship between words that is captured by WordNet ontology. For example, if the ground truth of an image is "car" and the automatic propagation is "automobile" then since "car" is present in the local tree of "automobile", as "car" is semantically related to "automobile" through is-a relationship, we can treat the image as a weaker positive example of "automobile".

5.1 Evaluation

We chose to test our system by defining a new evaluation scheme. We used Average ConceptNet similarity measure as opposed to precision-recall. This is done since precision-recall does not take into account the semantic relationship between concepts and is therefore inadequate in determining the performance of our algorithm. This is intuitive since an image of "flower", mislabeled as "plant" is not a large error and this fact should be addressed by the evaluation procedure.

We now describe our evaluation mechanism. During each iteration, the ConceptNet similarity between automatically propagated synsets and the ground truth was computed for all images. Let us assume that image i has N automatically propagated synsets. The system then picked a subset G, of these automatic synsets such that

$$G = \{k \mid L_f(k \mid i) \geq \alpha\}, \qquad <22>$$

where α is a constant and $L_f(k|i)$ is the feature likelihood of propagating synset k to image i. This is intuitive since we do no want to consider those synsets which have a very low propagation likelihood, as they do not help in bridging the gap between semantics and low-level features of an image and in tasks such as search. The system then computed the expected ConceptNet similarity s_i for an image i as:

$$s_i = 1 - \min_{k,j} d(g_{i,j}, a_{i,k}), \quad k \in G,\ \mathrm{G} \neq \varnothing,\ j \in 1..M,$$
$$s_i = 0,\ if\ \mathrm{G} = \varnothing, \qquad <23>$$

where M is the number of ground truth synsets associated with image i, $a_{i,k}$ is the automatically propagated synset, $g_{i,j}$ is the ground truth synset and $d(g_{i,j}, a_{i,k})$ is the distance measure between two synsets computed using ConceptNet[ref. Section 2.2].

The ConceptNet similarity was then averaged over all the images in the database to

determine the performance of the algorithm. The average ConceptNet similarity is:

$$\overline{D_c} = \frac{1}{M}\sum_{i=1}^{M} s_i, \qquad <24>$$

where M is the total number of images in the database. We plotted the average ConceptNet similarity against number of iterations for three different data sets when only one least likely image was picked for relevance feedback. This was done to mimic an ordinary end user who provides feedback on only one image.

Figure 2 show the ConceptNet similarity graphs for an optimized value of α that was determined by extensively testing the system for different values of α. The graphs represent an average of 10 experiments, each carried out with a different initial random image. This was done to avoid the chance exposure of an initial image that could be semantically close to the other images in the database.

The baseline similarity curve in the graphs indicates the lower bound on the performance of the algorithm. The baseline similarity was computed by exposing and annotating a single image in each relevance feedback cycle without any semantic propagation. So, after exposing *k*

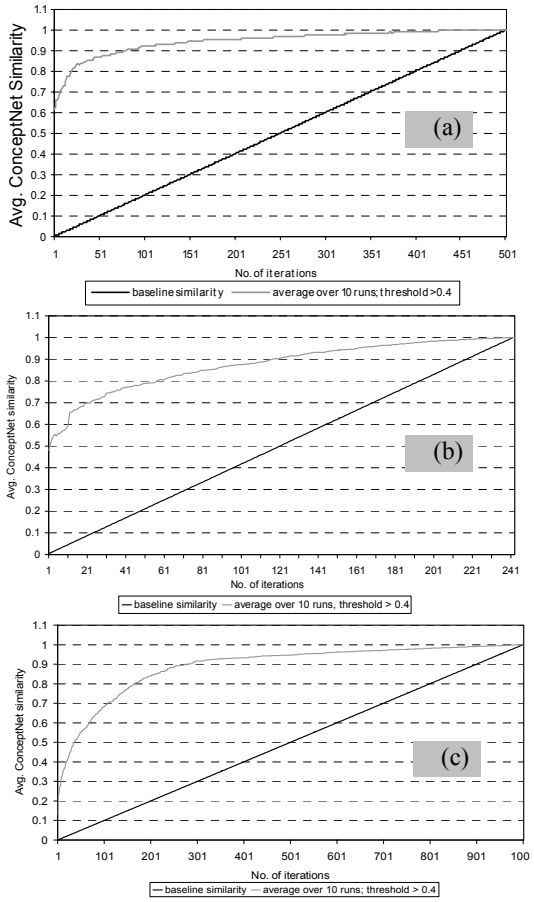

Figure 2: Average ConceptNet similarity against No. of iterations for – (a) a set of 242 images. Ground truth contains 90 distinct WordNet synsets. One image is exposed in each relevance feedback cycle. Semantic similarity rises to 0.8, when only 58/242 (~23.9%) of the images are annotated, (b) 500 personal image collection. Ground truth contains 107 distinct WordNet synsets. One image is exposed in each relevance feedback cycle. Semantic similarity rises to 0.8, when 20/500(~4%) of the images are annotated and (c) 1000 image dataset. Ground truth contains 44 distinct WordNet synsets. One image is exposed in each relevance feedback cycle. Semantic similarity rises to 0.8, when 168/1000 (~16.8%) of the images are annotated.

images, the baseline similarity will be *k/M*, where M is the total number of images in the database.

As the graphs suggest, the semantic similarity between the annotations and the images increases with an increase in relevance feedback iterations. This is intuitive since we take into account the semantic relationships between annotations, using ConceptNet similarity measures that work very well, as ConceptNet captures a wide variety of semantic relationships.

Our results compare well with related work [3,14]. For example, in [14] the authors show that the algorithm converges to 90% accuracy within four iterations. However, in *each* iteration, the system evaluates a 100 images for relevance feedback. In our system, we achieve a ConceptNet similarity of 0.8, using only 58/242 iterations. In [3] the authors report achieving a 50% accuracy by annotating only 20% of the images; this compares well with our result. However, since the authors use only class membership and not ConceptNet to present their results, we believe that their results will improve with the use of ConceptNet.

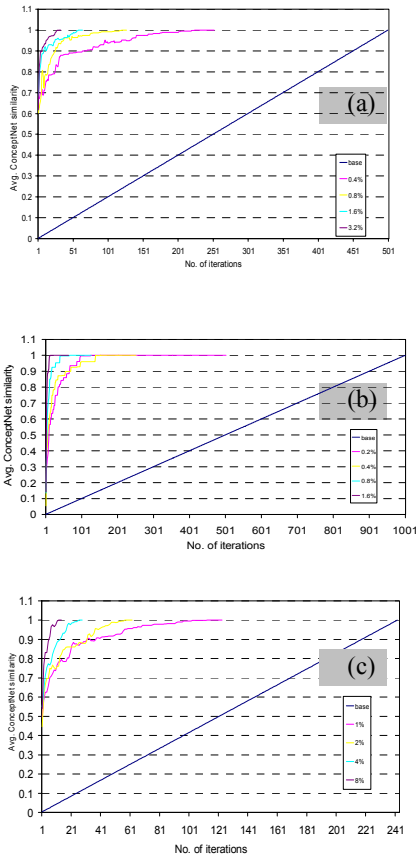

Figure 3: Average ConceptNet similarity v/s No. of iterations, when different percentages of images are exposed as least likely for relevance feedback on the dataset of – (a) 242 images, (b) 500 personal image collection and (c) Corel dataset of 1000 images. ConceptNet similarity increases faster with an increase in number of exposed images per cycle. For all datasets, the threshold is greater than 0.4.

Providing relevance feedback for larger number of images. We also conducted experiments to study the performance of the algorithm when different fraction of the images were exposed in each cycle for relevance feedback. Figure 3 shows the average ConceptNet similarity graphs for the three datasets when different percentage of images were picked as least likely**.**

As the graph suggests, the semantic similarity rises faster with an increase in number of images that are exposed for relevance feedback. This is intuitive since higher number of synsets get introduced and propagated in the system at each relevance feedback cycle and therefore, the similarity between the annotations and the image semantics rises faster.

6. Conclusion

In this paper, we have presented – (a) a novel strategy for semantic propagation, (b) a visual annotation interface and (c) novel evaluation scheme. The semantic propagation is done using – (1) low-level features (2) WordNet ontology, (3) ConceptNet repository and (4) relevance feedback. The visualization interface allows the user to browse, organize and annotate a large collection of multimedia images using *visual concepts*. This interface is interactive, real-time and scalable. Our evaluation scheme was based on ConceptNet similarity measure and *not* on precision-recall as we believe that precision-recall does not fully utilize the linguistic relations amongst words when evaluating the performance of our annotation algorithm.

We then conducted experiments to show how the semantics propagate across the database. The results indicate that the system performs much better than the baseline case, and as the number of relevance feedback cycles increases, the semantic association between the images and the annotations, becomes more refined and accurate.

In the future, we plan on incorporating sophisticated machine learning algorithms that use Support Vector Machines, for better semantic propagation. We are also looking at incorporating user-context for personal ontologies and incorporating event structures associated with personal media collection into the system. We also plan to conduct extensive evaluation of our visual annotation interface.

References

[1] P. APPAN, B. SHEVADE, H. SUNDARAM, et al. (2005). *Interfaces for Networked Media Exploration and Collaborative Annotation.* International Conference on Intelligent User Interfaces, 2005.

[2] B.SHEVADE and H.SUNDARAM (2003). *Vidya: An Experiential Annotation System*, 1st ACM Workshop on Experiential Telepresence, in conjunction with ACM Multimedia 2003, Berkeley CA,

[3] E. CHANG, K. GOH, G. SYCHAY, et al. (2003). *CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines.* IEEE Transactions on Circuits and Systems for Video Technology **13**(1): 26-38.

[4] X. HE, W.-Y. MA, O. KING, et al. (2002). *Learning and inferring a semantic space from user's relevance feedback for image retrieval*, Proc. of the 10th international conference on Multimedia, Juan Les-Pins, France, 343-346, Dec. 2002.

[5] P. S. HUGO LIU (to appear 2004). *ConceptNet: a practical commonsense reasoning toolkit.* BT Technology Journal **22(4)**: 211-226.

[6] A. K. JAIN (1989). Fundamentals of digital image processing. Englewood Cliffs, NJ, Prentice Hall.

[7] JIA LI., CATHERINE PLAISANT and B. SHNEIDERMAN (1998). *Data Object and Label Placement for Information Abundant Visualizations.* Workshop on New Paradigms in Information Visualization and Manipulation (NPIV '98),ACM, New York.: 41-48.

[8] G. A. MILLER, R. BECKWITH, C. FELLBAUM, et al. (1993). *Introduction to WordNet: An On-line Lexical Database.* International Journal of Lexicography **3**(4): 235-244.

[9] Y. RUI and T. HUANG (1999). *A Novel Relevance Feedback Technique in Image Retrieval.*, ACM Multimedia 1999,

[10] B. SHEVADE and H. SUNDARAM (2005). *A Visual Annotation Framework using Common-Sensical and Linguistic Relationship for Semantic Media Retrieval.* AME-TR-07,2005.

[11] B. SHNEIDERMAN and H. KANG (2000). *Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos*, In: Proc. International Conference Information Visualization (IV2000). London, England,

[12] H. SRIDHARAN, H. SUNDARAM and T. RIKAKIS (2003). *Context, memory and Hyper-mediation in Experiential Systems*, 1st ACM Workshop on Experiential Telepresence, in conjunction with ACM Multimedia 2003,, Berkeley CA, Nov. 2003.

[13] L. WENYIN, S. DUMAIS, Y. SEN, et al. (2001). *Semi-Automatic Image Annotation*, Proc. Human-Computer Interaction--Interact 01, pp.326-333,

[14] L. YE, C. HU, X. ZHU, et al. (2000). *A Unified Framework for Semantics and Feature Based Relevance Feedback in Image Retrieval Systems.*, In: Proc. ACM MM2000, 31-38,