Modeling Personal and Social Network Context for Event Annotation in Images

Bageshree Shevade *Hari Sundaram* *Lexing Xie*

Arts Media and Engineering, Arizona State University IBM TJ Watson Research Center

{bageshree.shevade, hari.sundaram}@asu.edu xlx@us.ibm.com

ABSTRACT

This paper describes a framework to annotate images using personal and social network contexts. The problem is important as the correct context reduces the number of image annotation choices.. Social network context is useful as real-world activities of members of the social network are often correlated within a specific context. The correlation can serve as a powerful resource to effectively increase the ground truth available for annotation. There are three main contributions of this paper: (a) development of an event context framework and definition of quantitative measures for contextual correlations based on concept similarity in each facet of event context; (b) recommendation algorithms based on spreading activations that exploit personal context as well as social network context; (c) experiments on real-world, everyday images that verified both the existence of inter-user semantic disagreement and the improvement in annotation when incorporating both the user and social network context. We have conducted two user studies, and our quantitative and qualitative results indicate that context (both personal and social) facilitates effective image annotation.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**] Information filtering, search process

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Social networks, context, event annotation, images, content management, multimedia

1 INTRODUCTION

In this paper, we develop a novel collaborative annotative system that exploits the correlation in user context and the social network context. This work enables members of a social network to effectively annotate images. The problem is important since online image sharing frameworks such as Flickr [1] have become extremely popular, yet the user-supplied tags are relatively scarce compared to the number of annotated images. The same tag can additionally be used in different senses, making the problem even more challenging. In such systems, text tags are the primary means used to search for photos, hence robust annotation schemes are very much desired. The social network context is important when different users' annotations and their corresponding semantics are highly correlated.

JCDL '07, June 17–22, 2007, Vancouver, British Columbia, Canada.

The annotation problem in social networks has several unique characteristics different from the traditional annotation problem.

- The participants in the network are often family, friends, or co-workers, and know each other well. They participate in common activities – e.g. traveling, attending a seminar, going to a film, parties etc. *There is a significant overlap in their real world activities.*
- Social networks involve multiple users – this implies that each user may have a distinctly different annotation scheme, and different mechanisms for assigning labels. There may be significant *semantic disagreement* amongst the users, over the same set of images.

The traditional image annotation problem is a very challenging one – only a small fraction of the images are annotated by the user, severely restricting the ground truth available. This makes the problem of developing robust classifiers hard. It may seem fruitless to develop annotation mechanisms, for social networks – where the problems seem to have multiplied.

Counter intuitively, the annotation problem is *helped* by the formation of the social network. The key observation here is that members of the social network have highly correlated real-world activities – i.e. they will participate in common activities together, and often repeatedly. For example two users may be good friends who always do things together – e.g. shop together at the McAllister mall. For user1, "shopping" and "user2" and "McAllister" go together. Also, the shopping event may recur many times over their friendship. Since their activities are correlated, such correlation can have effects on the data – they will often take images of the same event / activity, *thus effectively increasing the ground truth* available. Detecting correlation amongst members of the social network, and the *specific context* in which these common activities occur, can greatly help the annotation algorithms.

In our approach we define event context – the set of facets / attributes (image, who, when, where, what) that support the understanding of everyday events. Then we develop measures of similarity for each event facet, as well as compute event-event and user-user correlation. The user context is then obtained by aggregating event contexts and is represented using a graph. Recommendations are generated using a spreading activation algorithm on the user context, when given a query event attribute. For social network based recommendations, we first find the optimal recommender, by computing the correlations between the personal context models of the network members. Then we perform activation spreading on the recommender, but filter the recommendations with a salient subset of the current user's context.

We have conducted two experiments – one to verify the empirical observation that there exists semantic disagreement and the

second on our proposed personal and social context based annotation system. Our first user experiment on real-world personal images indicates that semantic diversity is greater on everyday personal photos when compared to the Corel dataset.. The second experiments indicate that context (both personal and social) can significantly help event annotation when compared to baseline recommendation systems.

In the next section we review related work in this area. In section 3 we present the event context framework. In section 4, we present our recommendation algorithms that use personal and social context. We discuss two experiments in section 5 and section 6, the former presents the experimental evidence for inter-user disagreement, and the latter shows the improvement in annotation when the user and social network context are incorporated.

2 RELATED WORK

There has been recent interest in '*folksonomy*' [8,15,17]. It has been noted that a large number of ordinary untrained folk are tagging media as part of their everyday encounters with the web (http://del.icio.us), or with media collections (http://www.flickr.com). The attraction of folksonomy lies in the idea that collective tagging can significantly reduce the time to determine media that are semantically relevant to the users, for example as part of a search. Our research for image annotation broadly falls under this umbrella – the key issue is that we believe the annotations to be recommended are only interesting / relevant, *provided the context is correct.*

There has been prior work in using groups for the purposes of image annotation / labeling [2,16]. In the ESP game [2], the authors develop an ingenious online game, in which people play against each other to label the image. In [16] the authors take into account browsing history with respect to an image search for determining the sense associated with the image. Both work aims at recovering one *correct* sense either shared by common knowledge or the user's own history. The context in which the annotation is used / labeled is not taken into account. In [19] the authors explore a collaborative annotation system for mobile devices. There they used appearance based recommendations as well as location context to suggest annotations to mobile users.

In [13], the authors provide label suggestions for identities based on patterns of re-occurrence and co-occurrence of different people in different locations and events. However, they do not make use of user-context, or commonsensical and linguistic relationships and group semantics.

In [4,10], the authors use sophisticated classification techniques for image annotation. However, they do not investigate collaborative annotation within a social network. The image based classifier schemes run into two broad problems: (a) scalability – each tag, requires its own classifier, and (b) the fact that people may use a tag in very different senses makes the classifiers difficult to build.

A key limitation of prior work is that there is an implicit assumption that there is one correct semantic, that needs to be resolved through group interaction / classification. In social networks the assumption of consistent labeling of images (thus implying semantic agreement) over the dataset may not hold over a diverse set of concepts. In prior work [14], we have observed that there is non-negligible disagreement among users, particularly on concepts that are more abstract rather than concrete. For example, people are more likely to disagree on abstract concepts such as "love", "anger", "anxiety" etc. as compared to everyday concepts such as "pen", "light bulb", "ball" etc.

Secondly, the context in which the annotation is used / labeled is not taken into account. We argue that for effective annotation we need to extend the feature / text based approaches to annotation as they do not exploit *the context in which the annotations have been made.* Users may annotate very similar images (say from the workplace) with very different tags, while they may use the same tags to describe very different activities. Thus, in order to understand these differences, we need to understand the context in which these annotations were used.

We believe that in general, both traditional taxonomies (as implied by traditional image based classifiers) as well as folksonomies are needed in semantic annotation. However, both frameworks will benefit through the incorporation of event context. We next present our event context framework and its relationship to user context.

.

3 EVENT CONTEXT

An event refers to a real-world occurrence, which is described using attributes such as images, and facets such as who, where, when, what. We refer to these attributes as the event context – *the set of attributes / facets that support the understanding of everyday events*. This event model definition draws upon recent work by Jain and Westermann [18]. While the notion of an event can be abstract in general, in this paper we restrict our discussion of events to being associated with *a single time and place* only.

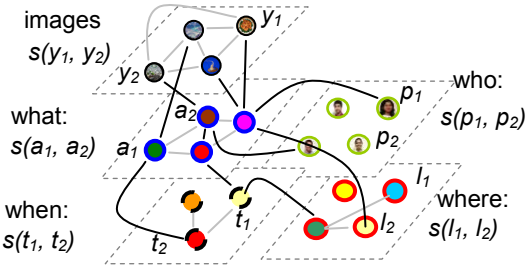

Figure 1: Context plane graphs for the who, where, when, what and the images facets of a context slice. The nodes in the context plane graph are the annotations and the black edges indicate the co-occurrence of the annotations. Note s(.,.) denote the facet similarity between two words/locations/activity etc. The strong (black) links denote association, i.e., nodes in different planes are connected if they co-occur in one image; the weak (gray) links denote similarity, i.e., edge strength obtained by evaluating the similarity function between two nodes in the same facet.

The notion of "context" has been used in many different ways across applications [6]. Note that set of contextual attributes is always application dependent [7]. For example, in ubiquitous computing applications, location, identity and time are critical aspects of context [6]. In describing everyday events the *who*, *where*, *when*, *what* are among the most useful attributes, just as news reporting 101 would teach "3w -- who when where" as the basic background context elements for reporting any real-world event.

3.1 The user context model

In our approach the user context is derived through aggregation over the contexts of the events in which the user has participated. This can be conceptualized as a graph, where the semantics of the nodes are from each different event facet (who, where, what, when and image), and the value of each node then is the corresponding image feature / text annotation. The edges of the graph encode the co-occurrence relationship as weights. So if "Mary" and "Mall" co-occur twice, then the strength of the edge between the nodes is 2. Figure 1 show the user context.

ConceptNet [11] is used to get contextual neighborhood nodes for the *what facet* nodes that are already present in the graph. This enables us to obtain additional relevant recommendations for the user. For every *what* node in the graph, the system introduces top five most relevant contextual neighborhood concepts obtained from ConceptNet as new nodes in the graph. These nodes are connected to the existing nodes with an edge strength of 1. These nodes now become a part of the context model.

3.2 Concept similarity

We now discuss the similarity measures for the different event facets. We first derive a new ConceptNet based event similarity measure for a pair of concepts. We then extend this similarity measure to two sets of concepts. Similarity measures over the context facets are then defined using the two above measures.

3.2.1 The ConceptNet based semantic distance

In this section, we shall determine a procedure to compute semantic distance between any two concepts using ConceptNet – a popular commonsense reasoning toolkit [11].

ConceptNet has several desirable characteristics that distinguish it from the other popular knowledge network – WordNet [12]. First, it expands on pure lexical terms to include higher order compound concepts ("buy food"). Secondly, it greatly expands on three relations found in WordNet, to twenty. The repository represents semantic relations between concepts like *"effect-of"*, *"capable-of"*, *"made-of",* etc. Finally, ConceptNet is powerful because it contains practical knowledge – it will make the association that "students are found in a library" whereas WordNet cannot make such associations. Since our research is focused on recommending annotations to images from everyday events, ConceptNet is very useful.

The ConceptNet toolkit [11] allows three basic functions on a concept node [11]:

- `GetContext(node)` – this finds the neighboring relevant concepts using spreading activation around the node. For example – the neighborhood of the concept *"book"* includes *"knowledge"*, *"library"*, *"story"*, *"page"* etc. ConceptNet terms this operation as "contextual neighborhood" of a node.
- `GetAnalogousConcepts(node)` – Two nodes are analogous if they derive incoming edges (note that each edge is a specific relation) from the same set of concepts. For example – analogous concepts for the concept *"people"* are *"human"*, *"person"*, *"man"* etc.
- `FindPathsBetweenNodes(node1,node2)` – Find paths in the semantic network graph between two concepts, for example – path between the concepts *"apple"* and *"tree"* is given as *apple [isA] fruit*, *fruit [oftenNear] tree*.

Neighbors of Concepts: Given two concepts *e* and *f*, the system determines all the concepts in the contextual neighborhood of *e*, as well as all the concepts in the contextual neighborhood of *f*. Let us assume that the toolkit returns the sets C_e and C_f containing the contextual neighborhood concepts of *e* and *f* respectively. The context-based semantic similarity $s_c(e,f)$ between concepts *e* and *f* is now defined as follows:

$$s_c(e,f) = \frac{|C_e \cap C_f|}{|C_e \cup C_f|}, \quad <1>$$

where $|C_e \cap C_f|$ is the cardinality of the set consisting of common concepts in C_e and C_f and $|C_e \cup C_f|$ is the cardinality of the set consisting of union of C_e and C_f.

Analogous Concepts: Given concepts *e* and *f* the system determines all the analogous concepts of concept *e* as well as concept *f*. Let us assume that the returned sets A_e and A_f contain the analogous concepts for *e* and *f* respectively. The semantic similarity $s_a(e,f)$ between concepts *e* and *f* based on analogous concepts is then defined as follows:

$$s_a(e,f) = \frac{|A_e \cap A_f|}{|A_e \cup A_f|}, \quad <2>$$

where $|A_e \cap A_f|$ is the cardinality of the set consisting of common concepts in A_e and A_f and $|A_e \cup A_f|$ is the cardinality of the set consisting of union of A_e and A_f.

Number of paths between two concepts: Given concepts *e* and *f*, the system determines the path between them. The system extracts the total number of paths between the two concepts as well as the number of hops in each path. The path-based semantic similarity $s_p(e,f)$ between concepts *e* and *f* is then given as follows:

$$s_p(e,f) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h_i}, \quad <3>$$

where N is the total number of paths between concepts *e* and *f* in the semantic network graph of ConceptNet and h_i is the number of hops in path *i*.

The final semantic similarity between concepts *e* and *f* is then computed as the weighted sum of the above measures. We use equal weight on each of the above measures (in the absence of a strong reason to support otherwise), and write the concept similarity *CS* the as follows:

$$CS(e,f) = w_c s_c(e,f) + w_a s_a(e,f) + w_p s_p(e,f), \quad <4>$$

where $w_c = w_a = w_p = 1/3$.

In the next subsections, we use ConceptNet distances to compute distances in the *where* and *what* facets of the user and event context, since these two facets are described with a free-form natural vocabulary on which ConceptNet similarities are meaningful, while other facets such as *who* and *when* use quantitatively distances on time, or intersection on proper nouns.

3.2.2 Similarity between two sets of concepts

An event usually contains a number of concepts in a facet; therefore we also need a similarity measure between sets of concepts based on that between two individual concepts. We define the set similarity between two sets of concepts A and B, where A: $\{a_1, a_2, \ldots\}$ and B: $\{b_1, b_2, \ldots\}$, given a similarity measure $m(a,b)$ on any two set elements *a* and *b* in the following manner.

$$S_H(A,B \mid m) = \frac{1}{|A|}\sum_{k=1}^{|A|}\max_i\{m(a_k,b_i)\}, \quad <5>$$

This is the average of the maximum similarity of the concepts in set A with respect to the concepts in set B, where |A| is the cardinality of set A. The equation indicates that the similarity of set A with respect to set B is computed by first finding the most similar element in set B, for *each* element in set A, and then averaging the similarity scores with the cardinality of set A. S_H is a variant of the familiar Hausdorff point set distance measure used to compare sets of image features [9] from which we adapt for measuring similarity. We average the similarity instead of using the `min` operator as used in the original Hausdorff distance metric, since averaging is less sensitive to outliers. Like the original Hausdorff distance metric, this similarity measure is asymmetric with respect to the sets: $S_H(\text{A,B|s}) \neq S_H(\text{B,A|s})$.

3.2.3 Similarity across event attributes

We now briefly summarize the similarity measures used for each attribute of an event. This is useful in determining if one event is similar to another, as well as user to user similarity. Let us assume that we have two events e_1 and e_2. Note that measures are asymmetric and *conditioned on event* e_2.

- **what**: The similarity in the *what* facet is given as:

$$s(A_1,A_2) = S_H(A_1,A_2 \mid CS), \quad <6>$$

where A_1 and A_2 refer to the sets of concepts for the *what* facets of events e_1 and e_2 respectively.

- **who**: The similarity $s(P_1,P_2)$ for the *who* facet is defined as:

$$s(P_1,P_2) = \frac{|P_1 \cap P_2|}{|P_2|}, \quad <7>$$

where p_1 and p_2 are the set of annotations in the *who* facet of events e_1 and e_2.

- **where**: The similarity $s(l_1, l_2)$ for the *where* facet is given as:

$$s(L_1,L_2) = \frac{1}{2}\left(\frac{|L_1 \cap L_2|}{|L_2|} + S_H(L_1,L_2 \mid CS)\right), \quad <8>$$

Where L_1 and L_2 refer to the sets of concepts for the "location" facets of events e_1 and e_2 respectively The equation states that the total similarity between L_1 and L_2 is the average of the exact location intersection with the modified Hausdorff similarity.

- **when**: The similarity $s(t_1,t_2)$ for the *when* facet is given as:

$$s(t_1,t_2) = 1 - |t_1 - t_2| / T_{\max}, \quad <9>$$

where t_1 and t_2 are the event times, and T_{max} is a normalizing constant.

- **Image:** In our work, the feature vector for images comprises of color, texture and edge histograms. The color histogram comprises of 166 bins in the HSV space [3]. The edge histogram consists of 71 bins and the texture histogram consists of 3 bins. We then concatenate these three histograms with an equal weight to get the final composite feature vector. We then use the Euclidean distance between the feature histograms as the low-level distance between two images.

The event similarity measure (ES) between two events can then be defined as a weighted sum of the similarity measures across each event attribute.

$$ES(e_1,e_2) = \sum_{i=1}^{5}\omega_i s(e_1,e_2;i) \quad <10>$$

Where, s_i is the similarity measure of each attribute described in the preceding paragraph and ω_i is the weight of each similarity measure. The similarity measure $\delta(U_1,U_2)$ between two users U_1 and U_2 is just the Hausdorff event similarity with the ES similarity measure ES:

$$\delta(U_1,U_2) = S_H(E_1,E_2 \mid ES). \quad <11>$$

In this section we discussed how to measure similarity between any two events, overall similarity between any two users. We next discuss how these measures can be used for generating annotation recommendations.

4 GENERATING RECOMMENDATIONS

In this section we present our algorithms to generate recommendations. We use image attributes as the example query attribute in the discussion throughout this section, while this is easily generalized to an arbitrary event facet. We investigate two types of recommendations – based on a single user context, and based on a social network.

4.1 Single User Context

We first show how to derive recommendations for each user, given an image, from her context (ref Section 3.1). The single user context is important, as the user is most correlated to herself, than over any member of her social network. The user context essentially aggregates knowledge about the user, by using the past annotations made by the user (thus deriving user-related concepts, and important concept-concept co-occurrences for this user).

- *Initialization:* Given an image seed (query), we first determine the *k* closest images in the image facet to the seed based on image similarity. The query is assigned a unit weight and initially the weights of the neighboring images are set to zero. We then assign the weight to each *edge* connecting its *k* neighbors to be proportional to the similarity between the query and its neighbor. Each image in the user context that has already been annotated contains words corresponding to each facet. The initial weight of the concepts in all the facets is set to zero. The edge weight between any node in any facet (e.g. what) and its neighbors is set proportional to the similarity between the nodes. The edge weight *between* facets is set to unit weight (i.e. image connected to a specific word).
- *Weight propagation:* Given a weight at a particular node, the weight propagated to its neighbors is the product of the node weight, with the edge weight between the nodes.
- *Termination:* The spreading is done recursively, for *p* times. At this time, all the activated nodes are analyzed, and only those nodes whose aggregate weight is above a threshold are retained for recommendations. In our implementation $p = 3$ and the weight threshold is determined experimentally.

We use activation spreading since this is one of the preferred models for modeling memory and semantic processing [5]. At this point, we have recommendations for each event facet using the user context, assuming that at least one seed image activated each facet.

4.2 Social Network based recommendations

Why should social networks be useful in image annotation? We conjecture that if users tend to agree with each other in real-world conversations (as opposed to in image annotations), and share the same activity context (i.e. they behave similarly under similar circumstances), then they are likely to use similar annotations to describe similar events. Hence, contextual correlation is useful is determining the recommender(s) for a given user as she annotates her media.

Recommendations from a social network are obtained in two steps: determining an optimal recommender, followed by context filtering using the current user's activity context. Let us assume that the user is trying to annotate an image *e* from an event with the *who*, *where*, *when* and *what* fields. Let us also assume that the database consists of the initial context model for each user in the social network. We proceed as follows:

1. Use eq. <11> to find the *optimal recommender, i.e., the one member of the network with whom the current user has the highest contextual correlation*.
2. Query the optimal recommender's user context with the to be annotated image *e*.
3. Perform activation spreading using image e as a query, and determine recommendations per facet as in section 4.1. Let us denote this set of per facet recommendations as R_o.
4. Filter R_o using the *who* and *what* facet of the current users context as follows: (3a) Use the *who* facet in R_o, as the seed to the activation spreading. Then perform activation spreading as in section 4.1. Let us denote this set as R_f. (3b) Examine the *what* facet in R_f, and compute the ConceptNet similarity with the *what* facets in R_o. All the recommendations that exceed a threshold ε are presented to the user.

We use the *who* facet to start the filtering process, since people tend to name each other more consistently than, say, the *where* facet – people might annotate the place where they live as "home", or "apartment" etc. Therefore the *who* facet is good indicator or activity correlation. If the *who* facet recommendations in R_o are not present in the current user's context model, R_f will be empty. This intuition lies in the observation that that people who share activity contexts will also *both* know other people who participate in same contexts. Here we use one optimal recommender for simplicity, while the framework can easily be extended to account for multiple recommenders.

4.3 Updating User Context

After the user has annotated an image with the *who*, *where*, *when* and *what* fields, the system updates the context model for the current user by adding the corresponding new nodes. The system also updates the contextual correlation measures between the current user and the rest of the users in the network. Thus, as the users annotate more number of images, the recommendations will more accurately reflect the group dynamics.

In the two sections that follow, we present two sets of experiments that (1) attempt to quantify the degree of agreement amongst members of a media sharing social network over a common set of photographs, (2) evaluate the performance of context-based recommendations by measuring their utility and quality with respect to a frequency-based baseline model.

5 DO PEOPLE AGREE?

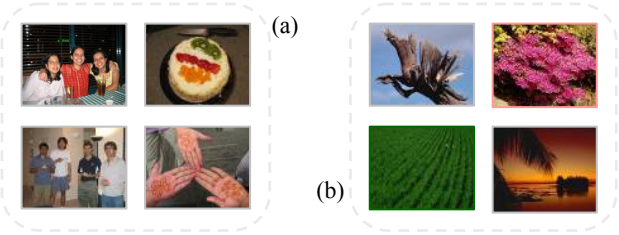

Figure 2: Example images from (a) personal image collection and (b) Corel dataset.

We conduct some preliminary experiments to understand how users in a group agree with each other as well as with the group as a whole. In this section, we shall use social network and group interchangeably. We asked a group of six graduate students to annotate (provide labels for) a set of 100 images. The students are part of *a social network* and know each other well from their shared academic environment and common daily activities. We use *a control group* of six professionals who do not form a social network. Both groups annotate the images independently and are not shown annotations of other users. The system also does not provide any kind of recommendations except the user's own frequency based annotations (as is common in web-browsers) to aid the process of annotation.

The images consisted of 60 photographs from a collection of shared events that were attended by members of the group as well as 40 images from the Corel dataset. This was done to understand the differences in agreement on a well-defined class of labels such as that belonging to the Corel dataset as well as a personal image collection. The shared events consisted of everyday events like birthday, farewell, get-together etc. Example user annotations for these set of images were "party", "fun", "cake", "dinner", "celebration" etc. The Corel dataset had images that were classified as agriculture, plants, desert etc. Example user annotations for these images included concepts like "irrigation", "harvest", "crop", "dry", "sand" etc. Figure 2 shows example images from the Corel dataset and from the personal image collection.

We measure how individual members agreed with the rest of the members of the social network as a whole. We compute several agreement measures for the members – (a) agreement measure of each user with the social network on a per image basis, (b) average agreement measure of all users per image, and (c) pair wise agreement measures among users per image.

5.1 Agreement of a user with group

We now present a measure for quantifying how individual members agreed with the rest of the members of the social network as a whole. Let us assume that the user Jane has annotated image *i* with annotations $a_1 \ldots a_m$. Let us also assume that *the other members of the group* have annotated the image *i* with annotations $g_1 \ldots g_n$. Then the group agreement measure of user agreement of U_1 with the rest of the social network on image *i*, $\gamma(U_1,i)$ is then given as:

$$\gamma(U_1,i) = S_H(A_1, G_1 \mid CS), \qquad <12>$$

Where S_H is the Hausdorff similarity measure (ref. Section 3.2.2), using the ConceptNet (CS) similarity measure between the

annotations. The agreement measure in equation <12> depends on the semantic similarity between the user's annotations as well as the annotations of the rest of the group. It also depends on the set size of the annotations. This is intuitive since the agreement of the user with the group should increase if she uses enough number of same or similar concepts as the rest of the group. A value of 0 indicates disagreement whereas a value of 1 indicates full agreement.

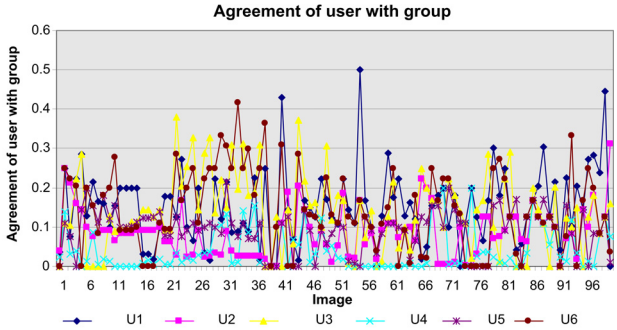

Figure 3: The six curves show the agreement of each of the six users with the group, per image.

Figure 3 shows the graph of agreement measures of each of the six users with respect to the group for the set of 100 images. The graph shows two things: (a) the agreement of each user to the rest of the group are image dependent (b) the agreement measures are low over most images.

5.2 Agreement of all users per image

We now present a measure to compute agreement of all users on a single image. We computed agreement measure on a particular image by taking into account all the annotations of the image as independent of the users who used those annotations for the image. This was done to understand how the group as a whole agreed on a particular media element.

Let us assume that the image *i* has been annotated with annotations $g_1 \ldots g_n$. These annotations are due to all the members of the group. We then compute the average agreement of all users on the image *i* as:

$$\gamma(i) = \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{N} CS(g_k, g_j)\frac{(w_k + w_j)}{2}, \qquad w_j = \frac{f_j}{\sum_{j=1}^{N} f_j}, \qquad \text{<13>}$$

where N is the number of all annotations associated with image *i*, $CS(g_k,g_j)$ is the ConceptNet similarity and f_i is the frequency of occurrence of annotation g_i in the set N. This agreement measure essentially is taking a weighted average of each pair-wise distance for all concepts present in both images, and the weights are the fractions that this pair occupies in all annotations. The average agreement on an image depends on the similarity between the annotations of the image as well as the frequency with which they occur. This is intuitive since the agreement on the image should increase with an increase in the number of same or similar annotations. A value of 0 indicates disagreement on an image i.e. all users annotated the media element with semantically dissimilar concepts, while a value of 1 indicates agreement.

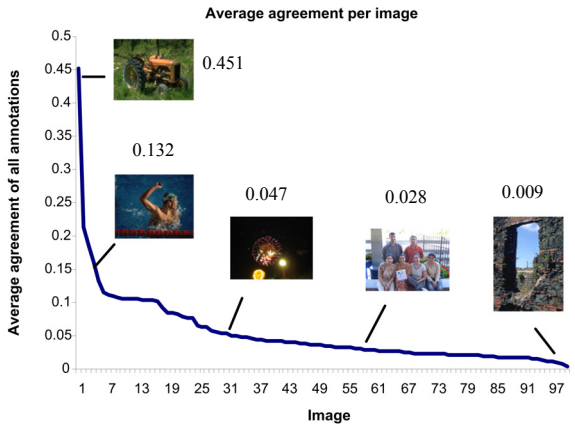

Figure 4: Average agreement value of all annotations per image. The annotations were obtained from all members of the group. The graph indicates that there is lack of consensus among members of the group on annotations of images.

Figure 4 shows the average agreement values of all users on an image. As the figure suggests, there is a higher disagreement among users on images of shared events that consisted of everyday activities. However, note that not all the images that had high agreement were Corel images. This implies that there is lack of consensus on even simple concepts.

Table 1: Mean agreement measure of all users – social network as well as the control group, on the personal and Corel dataset. Note that a larger number indicates a higher agreement score.

Dataset	Social Network	Control Group	No. of images
Corel	0.276	0.131	40
Personal	0.228	0.110	60
All	0.247	0.119	100

Table 1 shows the mean agreement measures of all users on the personal and Corel dataset. We can see that the mean agreement scores for a social network is nearly twice that of the control group of strangers on both datasets.

5.3 Pair-wise Agreement among users

We also computed pair wise agreement among users on a per image basis. This was done to gain insight into how users agree with each other or with a subset of the group as compared to the entire group.

Let us assume we want to compute the degree to which Jane agrees with Mary. Let us also assume that Jane has annotated an image *i* with annotations B={ $b_1, b_2 \ldots b_m$ } and Mary has annotated image *i* with annotations A={$a_1, a_2 \ldots a_n$}. Then agreement of user Jane (U1) to user Mary (U2), $\gamma(U_1,U_2)$ is given as:

$$\gamma(U_1, U_2 \mid i) = S_H(B, A \mid CS, i) \qquad \text{<14>}$$

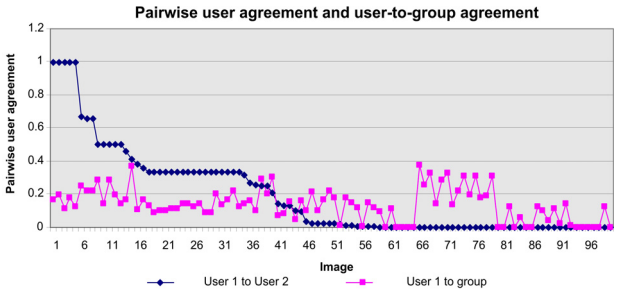

Figure 5: Pair wise agreement of user 1 to user 2 as well as agreement of user 1 with the entire group. The values are sorted in descending order of user-user agreement values.

The agreement measure between a pair of users depends on the ConceptNet similarity between the set of annotations of both users as well as the number of annotations. The agreement measure between two users is asymmetric. Our measure captures the "expressivity" of the user when adding annotations. The measure of agreement of Jane to Mary should increase if Jane uses enough number of same or similar annotations as Mary. Figure 5 shows the plots for two pairs of user-user as well as user-group agreement for a given user, sorted by user-user agreement values.

5.4 Lack of consensual agreement

Our experiments indicate that (Figure 5) there is a lack of consensual agreement among members of the group on their annotations. This occurs on both the Corel and the personal collection. Our results indicate that while users may have a low agreement with the entire group, a user may have a significantly higher agreement with a subset of the social network. While these results are indicative of the utility of social network correlation, we plan to conduct more large-scale experiments with publicly available datasets such as flickr. We next present our experimental results on image annotation using personal and social contexts.

6 EXPERIMENTS ON ANNOTATION RECOMMENTDATION

We conducted experiments to evaluate the quality of recommendations provided by measuring the utility and performance of three different recommendation methods. The three methods include our single user and social network context based recommendation algorithms, and a baseline frequency based recommendation (used in web browsers) algorithm.

1. *Frequency based personal recommendations:* These recommendations were based on the frequency of words used by the user while annotating her images. The system picks the three most frequently used words within each field (i.e. who, where, when and what) to generate the personal list for each field.
2. *Single User Context Model based recommendations:* These recommendations were obtained by activating the context models of the current user (ref. Section 4.1).
3. *Social Network based Recommendations:* The recommendations in this list are determined first by finding the optimal recommender, activating her context model and then filtering the recommendations with the current user's context. (ref. Section 4.2).

We compute these three types of recommendations on a set of newly uploaded images from the same set of users but a different set of events. After determining these three different types of recommendations, the system computes the union of the three recommendation lists and presents one combined list, L, for each of the who, where, when and what fields, as the final recommendation list to the user. We combine the different recommendation lists into one list to avoid any bias that might be introduced by the presentation order. The list is also sorted alphabetically to enable easy search of words within the list. Now, if the word chosen by the user is originally present in all the three lists, then the system gives credit to all the three lists. As the user annotates images through the web interface the system updates the user context model; the networked correlation is only updated at the end of the session.

6.1 Quantitative Results

We asked four graduate students to upload and annotate shared media using this system. The system was seeded with initial contextual correlation among users that was used to obtain the contextual correlation based recommendations. The users were presented all images that they had previously uploaded but not yet annotated, in the upload order. The users could choose to annotate any number of images as well as any of the images they liked. The context model of the users was updated as and when they annotated images. The users annotated a total of 132 images, with an average of 33 images per user. These images belonged to different kinds of events (22 distinct events across all users).

6.1.1 The utility of a recommendation method

We now show how to compute the utility value of the three recommendation methods. For each recommendation that was chosen by the user to annotate an image, we computed its variability value, i.e., the spread/distribution of that recommendation across the three different kinds of lists. *Intuitively, a recommendation method has high utility, if its recommendation is chosen by the user, and the recommendation is unique*. The recommendation is not common to the other methods. Conversely, if the recommendation is common to all three methods, then utility of each method is poor – the sophisticated algorithms are no better than the frequency based algorithms. The normalized variability *V(r)* of a chosen recommendation *r*, is given as:

$$V(r) = \frac{\log K}{\log N}, \qquad <15>$$

where N is the number of different kinds of recommendation lists (in our case, N = 3) and K is the number of different kinds of recommendation lists to which the chosen recommendation *r* belongs. where V(*r*) lies between 0 and 1. The utility value *U(r)*, of a recommendation is inversely related to the variability:

$$U(r) = (1-\alpha)V(r) \qquad <16>$$

where α is a constant and is set to 0.001. When entropy is 0, its utility value is 1, whereas when entropy is 1, its utility value is α. We have chosen α as the utility value, instead of 0, for the case when entropy is 1, because we wanted to give at least some small credit to the algorithm for suggesting the chosen recommendation, even though the chosen recommendation belonged to all three lists.

We compute the final utility value of the recommendation type, $U(f_i)$, as the average of all the utility values of the recommendations chosen from that type. $U(f_i)$ is given as:

$$U(f_i) = \frac{1}{M}\sum_{j=1}^{M} U(r_j \mid f_i), \qquad <17>$$

where M is the number of recommendations r_i that were chosen by the user from the given recommendation type list. We computed utility value for each recommendation type for each user.

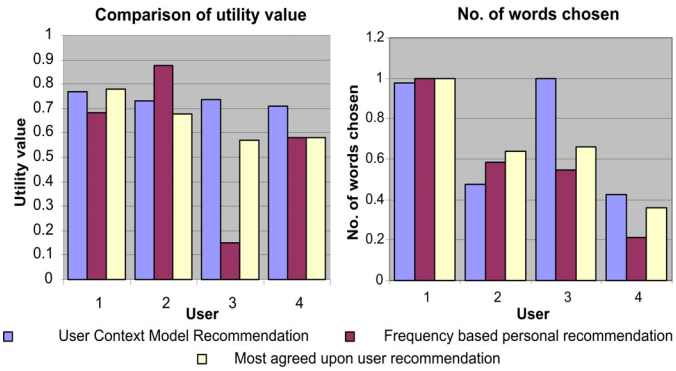

Figure 6: (a) Utility Graph indicating the utility value of each of the three different types of recommendation lists for each user. (b) Performance of each of the three recommendation methods, for each user.

Figure 6 shows the scaled performance of the three different lists. As the graph indicates, the performance of user context model based recommendations and contextual correlation based recommendations is much better than frequency based recommendations.

There are some key observations here: (a) context based recommendations (user or group) perform very well – contextual recommenders work well when there an a significant event overlap (b) frequency based recommendations are useful, when the users are annotating many images from the *same event.* (This was true for user 2). This is because it is highly likely that who, when, where fields will not change much between photos. (c) when there is little event overlap between members of the social network, the single user context framework is very useful.

7 CONCLUSIONS

In this paper, we described our approach to annotate events. We defined event context as comparing of image, who, where, what and when facets. The user context model was defined as an aggregate of event contexts. Then we developed similarity measures per facet as well as event-event and user-user similarity measures. Our recommendation algorithms incorporated activation spreading, when given an event facet as a query. The key observation in this paper was that people within a social networks often have correlated activities within a specific context, which in turn leads to correlated annotations for events and media artifacts (such as images) associated with the event.. These correlated annotations can be leveraged to increase the ground truth pool for the annotation system..

We experimentally showed that people significantly disagreed as a group over the semantics of a shared media collection. Furthermore, we showed that the agreement over a social network was significantly higher than a control group. However there were correlations amongst subsets of members. We conducted experiments to evaluate the utility and performance of each of the three different recommendation types. The results indicate that context based approaches work very well. The context based recommendation works especially well across events; within the same event a frequency based recommendation system also works well. We plan to extend this work by using exploiting contextual correlation across specific facets only, as well as model the temporal dynamics of user-context to be used as part of the recommendation algorithm.

8 REFERENCES

[1] *Flickr* http://www.flickr.com.

[2] L. V. AHN and L. DABBISH (2004). *Labeling images with a computer game*, Proceedings of the SIGCHI conference on Human factors in computing systems, 1-58113-702-8, ACM Press, 319-326, Vienna, Austria.

[3] A. B. BENITEZ, J. R. SMITH and S.-F. CHANG (2000). *MediaNet: A Multimedia Information Network for Knowledge Representation*, Proceedings of the 2000 SPIE Conference on Internet Multimedia Management Systems (IS&T/SPIE-2000), Nov 6-8, 2000., Boston MA.

[4] E. CHANG, K. GOH, G. SYCHAY and G. WU (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.

[5] A. M. COLLINS and E. F. LOFTUS (1975). *A Spreading Activation Theory of Semantic Processing.* Psychological Review **82**: pp. 407-428.

[6] A. K. DEY (2001). *Understanding and Using Context.* Personal and Ubiquitous Computing Journal **5**(1): 4-7.

[7] P. DOURISH (2004). *What we talk about when we talk about context.* Personal and Ubiquitous Computing **8**(1): 19-30.

[8] T. GRUBER (2005). *Folksonomy of Ontology: A Mash-up of Apples and Oranges*, First on-Line conference on Metadata and Semantics Research (MTSR'05). http://tomgruber.org/writing/mtsr05-ontology-of-folksonomy.htm,

[9] D. HUTTENLOCHER, G. KLANDERMAN and W. RUCKLIDGE (1993). *Comparing Images Using the Hausdorff Distance.* IEEETPAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence **15**(9): pp. 850-863.

[10] B. LI, K. GOH and E. CHANG (2003.). *Confidence-based Dynamic Ensemble for Image Annotation and Semantics Discovery.*, ACM International Conference on Multimedia., 195-206, Berkeley, CA.

[11] H. LIU and P. SINGH (2004). *ConceptNet: a practical commonsense reasoning toolkit.* BT Technology Journal **22**(4): pp. 211-226.

[12] G. A. MILLER, R.BECKWITH and C.FELLBAUM (1993). *Introduction to WordNet : An on-Line Lexical Database.* International Journal of Lexicography **3**(4): 235-244.

[13] M. NAAMAN, H. GARCIA-MOLINA, A. PAEPCKE and R. B. YEH (2005). *Leveraging Context to Resolve Identity in Photo Albums*, Proc. of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2005), June 2005, Denver, CO.

[14] B. SHEVADE, H. SUNDARAM and M.-Y. KAN (2005). *A Collaborative Annotation Framework*, Proc. International Conference on Multimedia and Expo 2005, also AME-TR-2005-04, Jan. 2005, Amsterdam, The Netherlands.

[15] B. STERLING (2005). *Order Out of Chaos*. Wired. **13.04** http://www.wired.com/wired/archive/13.04/view.html?pg=4.

[16] M. TRURAN, J. GOULDING and H. ASHMAN (2005). *Co-active intelligence for image retrieval*, Proceedings of the 13th annual ACM international conference on Multimedia, 1-59593-044-2, ACM Press, 547-550, Hilton, Singapore.

[17] T. V. WAL (2006) *Off the Top: Folksonomy* http://www.vanderwal.net/random/category.php?cat=153.

[18] U. WESTERMANN and R. JAIN (2007). *Toward a Common Event Model for Multimedia Applications.* IEEE Multimedia **14**(1): 19-29.

[19] A. WILHELM, Y. TAKHTEYEV, R. SARVAS, N. V. HOUSE and M. DAVIS (2004). *Photo Annotation on a Camera Phone*, ACM Conference on Human Computer Interaction, Apr. 2004, Vienna, Austria.