Exploiting Personal And Social Network Context For Event Annotation

Bageshree Shevade *Hari Sundaram* *Lexing Xie*

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

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

ABSTRACT

This paper describes our framework to annotate events using personal and social network contexts. The problem is important as the correct context is critical to effective annotation. Social network context is useful as real-world activities of members of the social network are often correlated, within a specific context. There are two main contributions of this paper: (a) development of an event context framework and definition of quantitative measures for contextual correlations based on concept similarity (b) recommendation algorithms based on spreading activations that exploit personal context as well as social network context. We have very good experimental results. Our user study with real world personal images indicates that context (both personal and social) facilitates effective image annotation.

1 INTRODUCTION

In this paper, we develop a novel event annotation system that exploits the user as well as the social network context. The social network context is important when users' real-world activities are highly correlated.

There has been prior work on image annotation using groups [1,6]. In [1] the authors develop an ingenious online game, in which people play against each other to label the image. In [6] the authors take into account the browsing history with respect to an image search for determining the sense associated with the image. In [5], the authors provide label suggestions for identities based on patterns of re-occurrence and co-occurrence of different people in different locations and events. A key limitation of prior is that there is an implicit assumption that there is one correct semantic, that needs to be resolved through group interaction / classification. Secondly, the context in which the annotation is used / labeled is not taken into account.

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 an 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 the current user's context. Our user experiments on real-world personal images indicate that context (both personal and social) can significantly help event annotation when compared to baseline recommendation systems.

In the next section we present the event context framework. In section 3, we present our recommendation algorithms that use personal and social context. We discuss our experiments in section 4 and then present our conclusions.

2 EVENT CONTEXT

An event refers to a real-world occurrence, which are 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 [7].

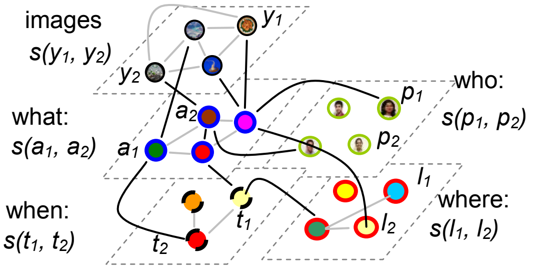

Figure 1: Context plane graphs for the who, where, when, what and the images facets of a context slice. The nodes in the context plane graph are the annotations and the black edges indicate the co-occurrence of the annotations. Note s(.,.) denote the facet similarity between two words/locations/activity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

The notion of "context" has been used in many different ways across applications [2]. Note that set of contextual attributes is always application dependent [3]. For example, in ubiquitous computing applications, location, identity and time are critical aspects of context [2]. 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.

2.1 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 [4] 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.

2.2 Similarity

We now discuss the similarity measures for the different event facets. We first present the ConcepNet based event similarity measure, and then similarity measures over the other facets.

2.2.1 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 [4].

ConceptNet is a large repository of commonsense concepts and its relations. It encompasses useful everyday knowledge possessed by people. The repository represents twenty semantic relations between concepts like *"effect-of"*, *"capable-of"*, *"made-of"* etc. The ConceptNet toolkit allows three basic operations on a concept – (a) finding contextual neighborhoods that determine the context around a concept or around the intersection of several concepts, for example – the context of the concept *"book"* is given by concepts like *"knowledge"*, *"library"*, *"story"*, *"page"* etc. (b) finding analogous concepts, that returns semantically similar concepts for a source concept, for example – that analogous concepts for the concept *"people"* are *"human"*, *"person"*, *"man"* etc. and (c) finding 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*.

Context 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:

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

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:

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

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:

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

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 have defined equal weight for each of the above measures. The final ConceptNet similarity CS is given as follows:

$$CS(e,f) = w_c s_c(e,f) + w_a s_a(e,f) + w_p s_p(e,f), \qquad \text{<4>}$$

where $w_c = w_a = w_p$ and $w_c + w_a + w_p = 1$.

2.2.2 Similarity between two sets of concepts

An event usually contain a number of concepts in a facet, therefore we define the set similarity between two sets of concepts A and B, where A: {a_1, a_2, …} and B: {b_1, b_2, …}, given a similarity measure *m(a,b)* on any two set elements *a* and *b* in the following manner.

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

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 adapted for measuring similarity. We average the similarity instead taking the `min` as in the original Haussdorff distance metric, since averaging is less sensitive to outliers. Note that the similarity measure is asymmetric with respect to the sets $S_H(\mathrm{A},\mathrm{B}|m) \neq S_H(\mathrm{B},\mathrm{A}|m)$.

2.2.3 Similarity across image 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:

$$s(A_1,A_2) = S_H(A_1,A_2 \mid CS), \qquad \text{<6>}$$

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:

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

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 he *where* is given as:

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

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

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

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

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. 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 agreement measure (ES) between two events then is the weighted sum of the similarity measures across each event attribute. The similarity measure $\delta(U_1,U_2)$ between two users U_1 and U_2 is just the Hausdorff similarity with the ES similarity measure ES:

$$\delta(U_1,U_2) = S_H(E_1,E_2 \mid ES). \qquad <10>$$

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.

3 GENERATING RECOMMENDATIONS

In this section we present our algorithms to generate recommendations. We restrict our focus to image attributes as the query attribute, but 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.

3.1 Single User Context

For each user the recommendations are derived from her user context through activation spreading. The seed in our application is the image facet corresponding to the event to be annotated.

We first determine the *k* closest images in the image facet to the seed based on image similarity; we then independently activate (i.e. propagate the weights) the nodes connected via edges to each of these *k* nearest neighbor image nodes. This is done recursively, until the propagated weight falls below a certain threshold. At this time, all the activated nodes are analyzed, and only those nodes whose aggregate weight is above a threshold are retained for recommendations. At this point, we have recommendations for each event facet using the user context.

3.2 Social Network based recommendations

Why should social networks be useful in image annotation? We conjecture that if users tend to agree with each other, 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.

The *optimal recommender is the one member of the network with whom the current user has the highest contextual correlation*. This is easily obtained using eq. <10>. The final recommendation is then filtered with the current users activity context.

Let us assume that the user is trying to annotate an image *a* from an event with the *who*, *where*, *when* and the *what* fields. Let us also assume that the database consists of initial context model for each user in the social network. We first determine the optimal recommender. We proceed as follows:

1. Query the optimal recommender's user context with the image to be annotated.
2. Perform activation spreading using the image to be annotated as a query, and determine recommendations per facet as in section 3.1. Let us denote this as R_o.
3. Filter R_o using the current users context as follows.
4. Use the *who* facet in R_o, as the seed to the activation spreading. Then perform activation spreading as in section 3.1. Let us denote this set as R_f.
5. 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 δ are presented to the user.

We believe that activity correlation is better estimated using *who* facet, as people will name each other consistently. Secondly, if the *who* facet recommendations in R_o are not present in the current user's context model, R_f is an empty set. This is intuitive as we conjecture that people who share activity contexts will also *both* know other people who participate in such contexts.

3.3 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 including adding any new nodes. The system then updates the contextual correlation measures between the current user and the rest of the users in the network and vice versa. Thus, as the users annotate more number of images, the recommendations will more accurately reflect the group dynamics.

4 EXPERIMENTS

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.
2. *Single User Context Model based recommendations:* These recommendations were obtained by activating the currently logged in user's own context model.
3. *Social Network based Recommendations:* The recommendations in this list are determined using the contextual correlation among members of the network.

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. Each method contributes the same number of words to the combined list to avoid bias. 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.

4.1 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).

4.1.1 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 entropy 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. We compute the normalized variability *V(r)* and the utility value *U(r)* of a chosen recommendation *r* as follows:

$$V(r) = \frac{\log K}{\log N}, \quad U(r) = (1-\alpha)V(r) \qquad <11>$$

Here 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. Note that V(*r*) lies between 0 and 1, and the utility value *U(r)*, of a recommendation is inversely related to the variability, with smoothing constant α set to 0.001 to give non-zero credit to shared yet correct recommendations:

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:

$$U(f_i) = \frac{1}{M}\sum_{j=1}^{M} U(r_j \mid f_i), \qquad <12>$$

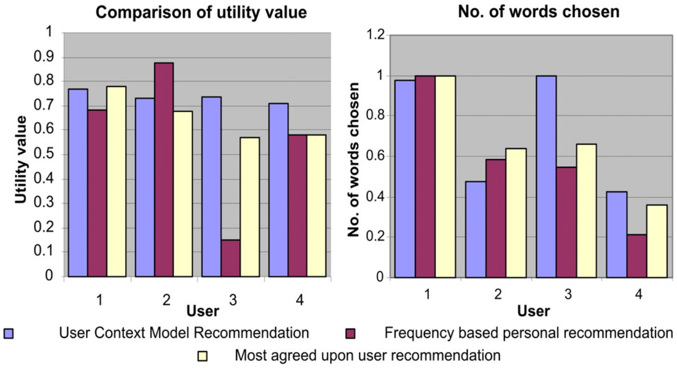

Figure 2: (a) Utility Graph indicating the utility value of each of the three different types of recommendation lists for each user. (b) Performance of each of the three recommendation methods, for each user.

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 2 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.

5 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 network often have correlated activities within a specific context. This increased the ground truth pool for the annotation system. 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 modeling the temporal dynamics of user-context to be used as part of the recommendation algorithm.

6 REFERENCES

[1] 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.

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

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

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

[5] 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.

[6] 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.

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