Transforming Personal Artifacts into Probabilistic Narratives

Setareh Rafatirad
George Mason University

Kathryn Laskey
George Mason University

***Abstract*—An approach focused on inferring probabilistic narratives from personal artifacts (including photographs) is presented in this work using personal photos metadata (timestamp, location, and camera parameters), formal event models, mobile device connectivity, external data sources and web services. We describe a new automated technique to discover data from multiple sources, and transform it into expressive, and probabilistic event-based semantics (narratives); the output is a graph of events. We introduce plausibility measure that indicates the occurrence-likelihood of an event node in the output graph. This measure is used in a ranking process used to find the best event among the merely possible candidates. In addition, we propose a new agglomerative clustering method that uses timestamp, location, and camera parameters in the EXIF header of the input photos to create event boundaries used to detect events.**

I. Introduction

The advancement of smart phone technology in recent years has turned it into a device used to record and capture personal experiences of real world events. Facebook statistics show that photos are the most popular form of personal artifacts[1]. The technology of current smart phones comes with multiple sensors like camera, and GPS, which enables the device to record time, GPS location, and camera parameters with the photo's EXIF header. Capturing photos is as easy as push of a button; this is followed by a high-demand for searching through personal photo archives to relive the events evidenced by the photos; an important information management paradigm that helps to fulfill this objective is image retrieval. Annotating personal artifacts with expressive tags supports this paradigm. Our goal in this work is to bridge the semantic gap which exists between high-level events (like watching movies, visiting a landmark) and photos produced by the machine. We propose a technique that automatically creates a context-aware event graph by combining event models with contextual information related to personal photos, personal information, and heterogeneous data sources. Our technique automatically computes the occurrence-likelihood values for the event nodes in the output graph; we refer to this value as plausibility measure in this work. Not all information for inferring events is hardwired to photographs; hence, it must be discovered. Personal photos have become rich sources of information about the events occurring in a user's life. Events themselves are also key cues to recall personal photos [16] and, therefore, they can be used to create searchable description metadata for them. Events, in general, are structured and their subevents have relatively more expressive power [20]. For instance, the event *Giving a Talk* is more expressive than its superevent, *Professional Trip*. In addition, instance events are contextual and should be augmented with context cues (like place, time, weather, participants). This makes instance events more expressive than event types. For example, the instance event *Giving a Talk at UCF at most two hours before meeting with Ted on a windy day* is far more expressive than the event type *Giving a Talk*. We define flexible expressiveness as follows: *a)* multi-granular conceptual description, which provides conceptual hierarchy in multiple levels using containment event relationships e.g. *subevent-of*, *subClassOf*; *b)* multi-context adaptation of conceptual description, which adapts a concept to multiple contextual descriptions (e.g., event type *visit-landmark* may have two instances; one instance associated with *Forbidden City* and the other to *Great Wall of China*). Currently, photos are not searchable based on expressive subevent tags because manual annotation is a labor-intensive task, and there is no standard mechanism to create and assign these tags to photos automatically and reliably. Consider the following example: A person takes a photograph at an airport less than 1 hour after his flight arrives. To explain this observation (i.e. the photograph), we first need the background knowledge about the events that generally occur in the domain of a trip. The corresponding semantics can only come from a domain event-ontology that provides the vocabulary for event/entity and event relationships related to the domain. In general, ontology is a powerful logical framework that is the glue that bonds human understanding of the real world and models of the real world in machines. An event-ontology could support flexible expressiveness. It allows explicit specification of models that could be modified using context information to provide very flexible models for high-level semantics of events. We refer to this modification as *Event Ontology Augmentation*. It constructs a more robust and refined version of an event-ontology either fully or semi-automatically. Secondly, given the uncertain metadata of a photo (like GPS that is not always accurate), the event type that the photo witnesses is not decisive; it might either be *rent a car*, or *baggage claim* that are two possible conclusions. Because the photo has incomplete information, the derived conclusions are not decisive, but merely possible — sometimes no single obvious explanation is available, but rather, several competing explanations exist and we must select the best one. In this work, reasoning from a set of incomplete information (or observations) to the most related conclusion out of all possible conclusions (or explanations) is performed through a ranking algorithm that incorporates the plausibility measure; this ranking process is used in *Event Ontology Augmentation*.

Problem Formulation: We assume that every input photo has context information (specifically, timestamp, location, and camera parameters) and a user/creator. Each photo belongs to a photo stream P of an event with a basic domain event-

[1] http://technology.inquirer.net/18188/facts-and-figures-about-facebook-2

ontology $O(V, E)$ whose nodes (V) are event/entity classes, and edges (E) are event/entity relationships, handcrafted by a group of domain experts. We assume that there is a bucket (B) in which external data sources are represented with a schema. The sources in B can be queried using the metadata of the input photographs and information about the associated user. Given P, B, O, and information associated to the user, how does one find the finest possible event tag that can be assigned to a photo or a group of similar photographs in P?

Solution Strategy: We propose Event Ontology Augmentation technique described as follows: select a relevant domain event ontology $O(V, E)$ through the information related to both the user and P. Using P, B, O, and the user information, infer S that consists of the best relevant subevent categories to P where $S \subseteq V$. An event category in S is the most plausible one among other competing candidates that have failed to be selected. For each competing candidate s_i, a plausibility measure m_i^p is calculated using function f to rank s_i and indicate how much it is relevant to c_j such that $c_j \subset P$ and c_j is a group of similar photographs: $f(s_i, c_j) = m_i^p$. Next, augment S using the information from B to obtain expressive event tags T. We define an event tag $t_i^e \in T$ as a subevent of an event that either exists in O, or can be derived from O such that t_i^e is the finest subevent tag that can be assigned to a group of similar photos. Also, if t_i^e is an assignable tag to any photo, and t_i^e does not exist in O, we intend to augment O by adding t_i^e to O using the shortest composition path such that the constraints governing O are preserved. Simply put, the final step is adding T to O by preserving the rules that govern O if $T \notin O$. The output is an extension to O that is referred as O_r. We argue that O_r (see fig 1) can be used for an event recognition task in photo annotation applications. The key insight in our proposed approach is to infer event characteristics from the image metadata (timestamp, location, and camera parameters), information about the user, ontological event model, mobile device connectivity, web services, and external data sources. We argue that attribute values related to an inferred event need to be obtained, refined, and validated as much as possible to create very expressive and reliable metadata for digital photographs and facilitate image search and retrieval. Fig 6 depicts the processing components of our proposed approach in the context of personal photo annotation.

Several event semantics are utilized in this work like spatiotemporal attributes/constraints of events, subevent structure, and spatiotemporal proximity. Unlike machine learning approaches that are limited to the training data set and require an extensive amount of annotation, we propose a technique in which existing knowledge sources are modified and expanded with context information in personal data sources (like Google Calendar, and social interactions), public data sources (like public event and weather directories, geographical and local business databases), and digital media archives (like personal photographs). With this knowledge expansion, new infrastructures are constructed to serve relevant data to communities. An example result set of our proposed work is depicted in fig 2. Event tags are propagated with event title, place information (like city, category, place name), time, weather, etc. Our proposed technique provides two unique key benefits as follows: *1)* An ontological event model is sufficiently flexible to express context attributes for events such that these attributes are not hardwired to the events, but rather they are discovered on the fly. This is a very important feature since it does not limit our approach to a single data set; *2)* leveraging

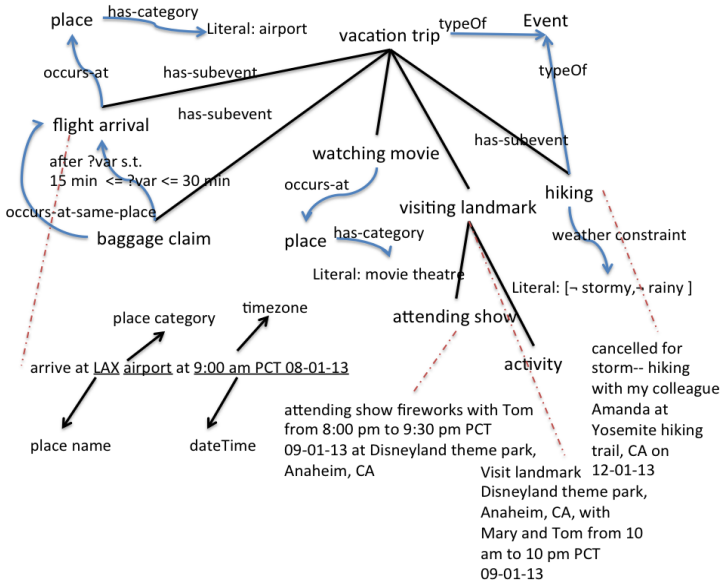

Fig. 1. An Example of Augmented Event Ontology.

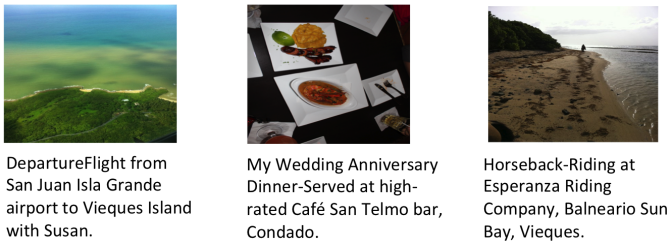

Fig. 2. Personal Photo Annotation using Event Ontology Augmentation.

context data across multiple sources could facilitate building a consistent, unambiguous knowledge base.

Event ontology augmentation has several challenges: *a)* we need a language that is characterized by formal semantics and can model different types of entity properties and relationships related to an event domain. OWL is widely used for developing ontologies. However, this language is limited in terms of its capability of describing the semantics of events. A major challenge is to create an extension of OWL and provide the grammar for that extension; *b)* collecting and combining information from multiple sources is a daunting task. It needs a general mechanism to automatically query sources and represent the output. It also needs a validation mechanism to ensure the coherency of the obtained data; *c)* currently, publicly available benchmark data sets such as those offered by TRECVid [6] do not suit the purpose of this research. Many deal with activities that are detected using low-level features in media content such as surveillance video. However, higher-level events have relatively more contextual characteristics; *d)* according to the useful properties of photoset, relevant event categories in the model must be discovered. So, those properties need to be identified. This paper is organized as follows: in section II, we review the prior art that use context and event models for annotating photographs; this is followed by section III in which we explain the our clustering method; in section IV and V, we explain our solution strategy; this is followed by section VI that demonstrates our experiments, and

section VII which is the conclusion.

II. RELATED WORK

The important role of context in image retrieval is emphasized in [6] and [13]. Context information and ontological event models are used in conjunction by [24], [23], [7]. Cao et al. present an approach for event recognition in image collections using image timestamp, location, and a compact ontology of events and scenes [4]. In this work, event tags do not address the subevents of an event. Liu et al. reports a framework that converts each event description from existing event directories (such as Last.fm, Eventful, and Upcoming APIs) into an event ontology that is a minimal core model for any general event [15]. This approach is not flexible to describe domain events (like 'trip') and their structure (like 'subevent' structure). Paniagua et al. propose an approach that builds a hierarchy of events using the contextual information of a photo based on moving away from routine locations, and string analysis of English album titles (annotated by people) for public web albums in Picasaweb. [18]. The limitations of this approach are: 1) human-induced tags are noisy, and 2) subevent relationship is more than just spatiotemporal containment. For instance, albeit a 'car accident' may occur in the spatiotemporal extent of a 'trip', it is not part of the subevent-structure of the 'trip'. According to [3], events form a hierarchical narrative-like structure that is connected by causal, temporal, spatial and subevent relations. If these aspects are carefully modeled for events, they can be used to create a descriptive knowledge base for interpreting multimedia data. The importance of building event hierarchies is also addressed in [20] where the main focus is on the issues of event composition using the subeventOf relationship between events. In [21], an image annotation mechanism is proposed that exploits context sources in conjunction with subevent-structure of an event — this structure is modeled in a domain event ontology. The limitation of this approach is no matter how much an event category is relevant to a group of photos in a photo stream, it is used in photo annotation. As a result of this operation, the quality of annotation degrades.

III. CLUSTERING

A photo has incomplete information that can be improved if combined with the information related to a group of similar photos and help to derive merely possible conclusions (i.e. event categories). In this paper, two images are similar if they belong to the same type of event. Partitioning a photo stream based on the context of its digital photographs can create separate event boundaries for the photos related to one event [5], [10], [12], [17], [19]. An event is a temporal entity. However, using time as the only dimension in clustering means ignoring other context semantics about events. Much better results can be obtained when both time and location information is used [9]. Gong et al. propose a framework for photo stream from single user that applies hierarchical mixture of Gaussian models based on context information including time, location, and optical camera parameters (such as ISO, Focal Length, Flash, Scene Capture Type, Metering Mode, and Subject Distance) [8]. In photos, optical camera parameters provide useful information related to the environment at which an event occurs, like 'indoor', 'outdoor', and 'night' [22].

In this section, we propose an agglomerative clustering that partitions a photo stream hierarchically according to the context information of photos, specifically timestamp, location, and optical camera parameters (referred as 'OCP'). Agglomerative clustering has several advantages: (a) it is fully unsupervised, (b) it is applicable to any attribute types, and (c) clusters can be formed flexibly at multiple levels (from coarser to finer). In general, larger events like 'trip' are often described using spatiotemporal characteristics whereas the subevent structure is limited by space and time. However, the depth of a spatiotemporal agglomerative clustering dendrogram can be extended using OCP to refine the precision of the clusters. Our clustering approach is described as follows: primarily, a photo stream is partitioned using timestamp, gps-latitude, and gps-longitude; the blue cluster structure in Fig 3, referred as *ST-cluster tree*, shows the output for this stage of the clustering. Next, for each ST-cluster in the blue structure, its content is partitioned based on OCP to create *ST-OCP cluster tree*. The orange structure in fig 3 shows the output of this stage. Although the orange hierarchy extends the blue one, it is important to know that these two structures are orthogonal to each other. We refer to this approach as *ST-OCP Agglomerative Clustering*. We did an experiment for which we asked 20 people (including the owner of photos, the people in the photos, and third party judges) to relatively assign a number to the result of each clustering experiment between the range of 0 to 6 based on the event boundaries produced by our clustering approach. This experiment was conducted on 30 different photo streams captured in different cities inside US. Our technique did a better job compared to the other agglomerative clustering approaches in terms of providing coarser and finer precision for event and subevent boundaries. We compared the dendrograms of ST-Clustering (location and time), ST-OCP-Clustering (our approach), OCP-Clustering, and STOCP-Clustering (in which location and time and OCP attributes are used together in the distance function). The arrangement of clusters depends on the image attributes that are used in the clustering. The photos are sorted in chronological order. Image content features are not used in these cluster arrangements. The equation '$OCP - Clustering \prec S - Clustering \prec T - Clustering \prec STOCP - Clustering \prec ST - Clustering \prec ST - OCP - Clustering$' shows that the arrangement of clusters improved from left to right in our experiment — *S-Clustering* and *T-Clustering*, respectively, mean that agglomerative clustering is conducted using the location, and the timestamp attributes of photos. An example comparison of event boundaries is shows in fig 4. We used single linkage clustering and Euclidean distance in our clustering technique. However, one can use other approaches and refine the results. We also used the standard deviation of the context space as the input for *linkage* function to find noisy fields. We observed that a considerable noise is created when a field does not have significant variations, for instance, when all photos have their Flash attribute set off. Because such noise distorts the arrangement of clusters, we discarded the fields with such noisy characteristics before the clustering.

IV. EVENT ONTOLOGY AUGMENTATION

Our goal is to derive the best possible subevent category from a set of incomplete observations. We present the observations with a set of descriptors. Each descriptor is a formula for a photo or a cluster — here, a cluster consists of a group of contextually similar photos. In this section, we show that it is feasible to go from a set of descriptors D to the best subevent category, when the following conditions are satisfied: (a) the descriptors in D are consistent among themselves, (b)

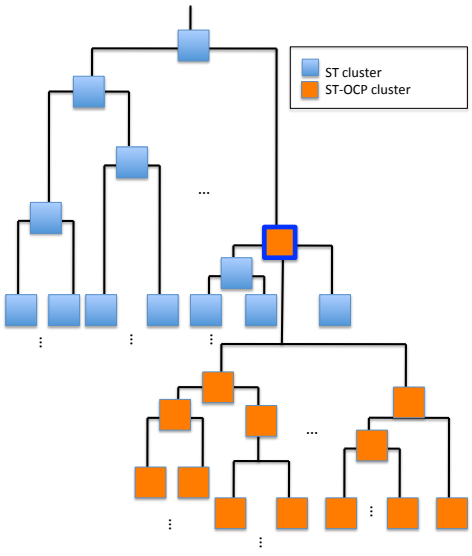

Fig. 3. ST-OCP Agglomerative Clustering.

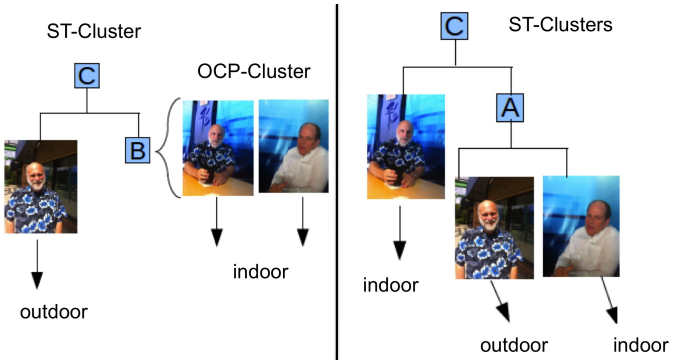

Fig. 4. Comparing ST-OCP Agglomerative Clustering with ST-Clustering. The clusters formed by ST-OCP clustering (on the left) provide relatively better event boundaries.

the descriptors in D satisfy subevent categories, (c) axioms of a subevent category are consistently formulated in an event ontology, and (d) the inferred subevent categories are sound and complete.

A. Event Model

We use a basic derivation of E* model [11] as our core event model, to specify the general relationships between events and entities. Specifically, we utilized the relationships *subeventOf*, which specifies the event structure and event containment. The expression e_1 $subeventOf$ e_2 indicates that e_1 occurs within the spatiotemporal bounds of e_2, and e_1 is part of the regular structure of e_2. Additionally, we used the spatiotemporal relationships like *occurs-during* and *occurs-at* to specify the space and time properties of an event. The time and space model that we used in this work is mostly derived from E* model. The relationship *participant* is used to describe the presence of a person in an event. We use the relationships *co-occurring-with*, and *co-located-with*, *spatially-near*, *temporal-overlap* , *before*, and *after* to describe the spatiotemporal neighborhood of an event. The relationship *same-as* between two events, makes them equivalent entities. Also, we used several other relationships to describe additional constraints about events (e.g., e_1 has-ambient-constraint A, and A has-value *indoor*). Moreover, to express a certain group of temporal constraints, we utilized some of Linear Temporal Logic, Metric Temporal Logic, and Real-Time Temporal Logic formulas [14], [2]. These formulas are a combination of the classical operators $\wedge$ (conjunction) , $\vee$ (disjunction) , implication ($\rightarrow$) , Allen's calculus [1], $\Box$ operator, $\Diamond$ operator, linear constraints, and distance functions; they are used to model complex relative temporal properties. For instance constraint $\Box_{[t_1,t_2]}(e_1 \rightarrow \Diamond_{[t_2,t_2+1800]}e_2 \wedge \tilde{\mathcal{D}}(e_2) \leq 1800)$ states that e_2 eventually happens within 1800 seconds after e_1 and that e_2 lasts less than or equal to 1800 seconds. We developed a language $\mathcal{L}$ with a syntax and grammar as an extension to OWL to embrace complex temporal formulas. Further, we extended the language to support a combination of classical propositional operators, linear spatial constraints, and spatial distance functions which can not be expressed in OWL; equation $f_{eucDist}(e_1, e_2, @ \leq 100)$ shows a relative spatial constraint in $\mathcal{L}$, which states the event e_1 occurs at most 100 meters away from the place at which event e_2 occurs.

Domain Event Ontology: A domain event ontology provides specialized taxonomy for a certain domain like *trip*, see fig 5. The *Miscellaneous* subevent category in this model is used to annotate the photos that are not matched with any other category. The general vocabulary in a core event model is reused in a domain event ontology. For instance, *Parking* in fig 5, is a *subClassOf* of *Occurrent* (or event) concept in the core event ontology. Also, relationships like *subeventOf* are reused from the core event ontology. We assume that domain event ontologies are handcrafted by a group of domain experts.

B. Descriptor Representation Model

We represent a descriptor using the schema in script $\{type_d : value_d, confidence_d : val\}$, in which $type_d$, $value_d$, and val indicate the type, value,and certainty (between 0 and 1) of the descriptor, respectively. For instance, the descriptor $\{Flash : `off`, confidence : 1.0\}$ for a photo, states that the flash was off when the photo was captured with 100% certainty. Photo and cluster descriptors follow the same representation model, however the rules for computing the value of $confidence_d$ are different. We will describe these rules in the following paragraphs. The descriptor model of a cluster includes two fields in addition to that of a photo: plausibility-weight ≥ 0 , and implausibility-weight < 0. Later, we will explain the usage of these fields. All descriptors are either *direct* or *derived*. For photo descriptors, by convention, we assume that a direct descriptor is straightly extracted from the EXIF metadata of a photo, and its confidence is 1, as in the above example. The direct descriptors that we used in this paper are related to time, location, and optical parameters of photos like *GPSLatitude* ,*GPSLongitude* , *Orientation*, *Timestamp*, and *ExposureTime*. For a derived descriptor like $\{sceneType : `indoor`, confidence : 0.6\}$, the descriptor value 'indoor' is computed using direct descriptors like *Flash*, through a sequence of computations that extract information from a bucket of data sources. Some of these descriptors are *PlaceCategory*[2], *Distance*[3], and *HoursOfOperation*[4]. The confidence score is obtained from the processing unit used to compute the descriptor value — we developed several information retrieval algorithms for this purpose,in addition to the existing tools in our lab [22]. If a descriptor value is

[2]The category of the nearest local business to the coordinates of a photo.
[3]The distance of a local business to the coordinates of a photo.
[4]The hours during which a local business is open.

directly extracted from an external data source, $confidence_d$ is equal to 1. Direct descriptors of a cluster must represent all photos contained in it; some of these descriptors represent *boundingbox*, *time-interval*, and *size* of the cluster. The confidence value for direct descriptors is equal to 1, for instance, in the descriptor $\{size : 5, confidence_d : 1.0\}$ that indicates the number of photos in a cluster, $confidence_d$ is equal to 1.

Given a photo p_i in a photo stream P, and the cluster c that groups p_i with the most similar photos in P, a processing unit produces the descriptors of c using the descriptors of the photos in c, and more importantly, this process is guided by the descriptors of p_i. Every photo in c must support every *derived* descriptor of p_i; such cluster is referred as a *sound cluster* for p_i, and the *derived* descriptors for c are represented by the distinct union of the *derived* descriptors of the photos in c. For a derived cluster descriptor d, the value of $confidence_d$ is calculated using the formula in equation 1, in which $|c|$ is the size of the cluster, p_j is every photo in c that is represented by d, and $f(p_j, d)$ gives the confidence value of d in p_j. To find a sound cluster for a photo, the hierarchical structure that is produced by the *clustering* unit, is traversed using depth-first search — the halting condition for this navigation, if no sound cluster was found, is when current cluster is a leaf node.

$$confidence_d = \frac{1}{|c|} \times \sum f(p_j, d) \quad (1)$$

Descriptor Consistency: As we mentioned earlier, consistency among a set of descriptors is a mandatory condition to infer the best possible conclusion from it. We make sure that consistency exists among the descriptors of a photo as well as the descriptors of a cluster, using entailment rules described below. *(a)* $v_i \rightarrow v_k$: if v_i implies v_k, then the rules for v_k must also be applied to v_i. This is referred as *transitive entailment rule*. For instance, suppose a photo/cluster has the following description, $'outdoorSeating : true'$; $'sceneType : outdoor'$; $'weatherCondition : storm'$, which implies that the nearest local business (e.g. restaurant) to the photo/cluster, offers *outdoorSeating*, and the weather was stormy when the photo(s) were captured. Given the sequence of rules below,

$$outdoorSeating \wedge outdoor \rightarrow fineWeather,$$
$$fineWeather \rightarrow \neg storm$$

rule 2 is entailed that indicates an inconsistency among the descriptors of a photo/cluster.

$$outdoorSeating \wedge outdoor \rightarrow \neg storm \quad (2)$$

(b) $v_i \rightarrow func_{remove}(v_k)$: v_i implies removing the descriptor v_k. This is referred as a *deterministic entailment rule*.
(c) $v_i \wedge v_k \rightarrow truth\ value$: rules of this type are referred as *non-deterministic entailment rules* in which the inconsistency is expressed by a false truth value e.g. $closeShot \wedge landscape \rightarrow false$. In that case, further decisions on keeping,modifying, or discarding either of the descriptors v_i or v_k will be based on the confidence value assigned to each descriptor — this operation is referred as *update*, which is executed when an inconsistency occurs between two candidate descriptors. The following rules are used by this process: (a) for two descriptors with the same type, the descriptor with lower confidence score is discarded, (b) for two descriptors with different types, the one with lower confidence score gets modified until the descriptors are consistent. The modification is defined as either *negation* or *expansion* within the search space. In case of negation, e.g. $\neg outdoor \rightarrow indoor$, the confidence value for *indoor* descriptor is calculated by subtracting the confidence value of *outdoor* descriptor from 1. An example of expansion is increasing a window size to discover more local businesses near a location. To avoid falling inside an infinite loop, we limit the count of negation, and the size of search space during expansion, by a threshold. We assign *null* to the descriptor that has already reached a threshold and is still inconsistent. *null* is universally consistent with any descriptor. The vocabulary that is used to model the descriptors for a photo/cluster is taken from the vocabulary that is specified in the core event model.

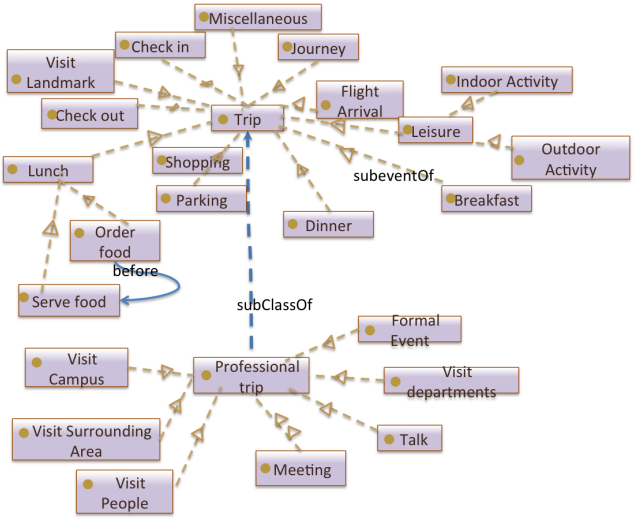

Fig. 5. An event ontology for the domain *professional trip*.

C. Bucket of Data Sources

We represent each data source with a declarative schema, by using the vocabulary of the core event model. This schema indicates the type of source output. In addition, it specifies what type of the input attributes a source needs, to deliver the output. Data sources are queried using the SPARQL language[5]. The following script shows an example of a SPARQL query that is formed to query a source; var_1 is a query variable (output that must be delivered by the source); $attr_1$ is the input attribute of the source; $class_w$ indicates a class type, and rel_a indicates a relationship. The class types and relationships used in such queries are constructed using the vocabulary of the core event model.

$$SELECT\ ?var_1\ FROM\ \ < Source\ URI > WHERE\ \{$$
$$attr_1\ \ core : typeOf\ \ class_w;\ var_1\ \ core : typeOf\ \ class_u;$$
$$?var_1\ \ core : rel_a\ \ ?x;\ ?x\ \ core : rel_b\ \ ?y;$$
$$?y\ \ core : rel_d\ \ attr_1.\ \}$$

The above query is constructed automatically using the schema of data sources, and the available information. Simply put, a source is selected if its input attributes match the available information I. At every iteration, I is incrementally updated with new data that is delivered by a source. The next source is selected if its input attributes are included in I. This process

[5]http://www.w3.org/TR/rdf-sparql-query/

continues until no more source with matching attributes is left in the bucket B.

D. Event Inference

From a set of consistent cluster descriptors, referred as *observations*, we developed an algorithm to infer the most plausible subevent category described in a domain event ontology. This algorithm, uses the domain event model, which is a graph; we represent this graph with the notation $O(V, E)$ in which V includes event classes, and E includes event relationships. Traversing the event graph O starts with the root of hierarchical subevent structure specified in the domain event ontology. The algorithm visits event candidates in E through some of the relationships in E like *subeventOf*, *co-occurring-with*, *co-located-with*, *spatially-near*, *temporal-overlap*, *before*, *after*, and *same-as* — these relationships help to reach other event candidates that are in the spatiotemporal neighborhood of an event. An expandable list, referred as L_v, is constructed from E, to maintain the visited event/subevent nodes during an iteration i — if an event is added to L_v, it cannot be processed again during the extent of i. At the end of each iteration, L_v is cleared. In every iteration, the best subevent category is inferred through a ranking process, from a set of consistent observations. We introduce *Measure of Plausibility* (m_i^p) which is used to rank event candidates, and help to find the most plausible subevent category. We compute m_i^p using two parameters (a) granularity score, and (b) plausibility score. The granularity score (w_g) is equivalent to the level of the event in the subevent hierarchy in the domain event ontology. To compute the plausibility score (w_{AX}), we used 'plausibility-weight' (w^+) and 'implausibility-weight' (w^-) which are two fields of a cluster descriptor (mentioned earlier). The value of w^+ is equal to the confidence value assigned to a descriptor, and the value of w^- is equal to $-w^+$. If a descriptor could not be mapped to any event constraint, w_{AX} remains unchanged. If a descriptor with $w^+ = \alpha$ satisfies an event constraint, then w^+ is added to w_{AX}, otherwise, w^- is added to w_{AX} (i.e., $w_{AX} = w_{AX} - \alpha$). The only exception is for the cluster descriptors *time-interval* and *boundingbox*; if either one of these descriptors satisfies an explanation, then $w^+ = 1$; in the opposite case, $w^- \leq -100$ — when a cluster has no overlap with the spatiotemporal extent of an event s_i, $w^- \leq -100$ makes s_i the least plausible candidate in the ranking. According to the formula in IV-D, w_{AX} also depends on the fraction of satisfied event constraints; N is the total number of constraints for an event candidate.

$$w_{AX} = \frac{1}{N} \sum w_{AX}^j \quad , 1 \leq j \leq N \tag{3}$$

Finally, we use the following instructions to compare two event candidates e_1 and e_2: when e_1 is subsumed by e_2, m_i^p for each event candidate is normalized using the formula in equation 4, in which $e_i \equiv e_1$ and $e_j \equiv e_2$, otherwise, $e_i.m_i^p = e_i.w_{AX}$. The candidate with the highest m_i^p is the most plausible subevent category.

$$e_i.m_i^p = \frac{e_i.w_{AX}}{\max(e_i.w_{AX}, e_j.w_{AX})} + \frac{e_i.w_g}{\max(e_i.w_g, e_j.w_g)} \tag{4}$$

When a subevent category is inferred from a set of observations, it will not be considered again as a candidate for the next set of observations. Event inference halts if no more subevent category is left to be inferred from the domain event ontology.

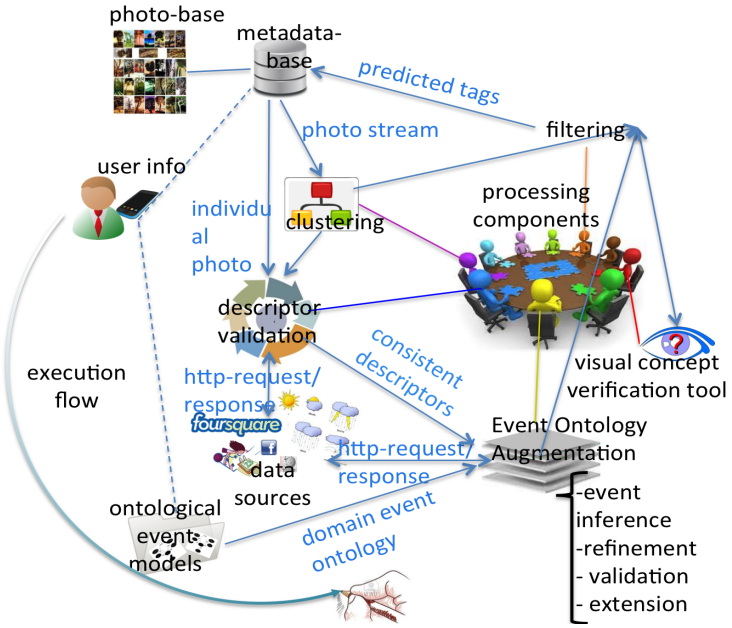

Fig. 6. The Big Picture. Photos and their metadata are stored in *photo-base* and *metadata-base* respectively. Using *user info*, including events' type, time, and space in a user's calendar, a photo stream is queried, and its metadata is passed to *clustering*. In *descriptor validation*, a set of consistent descriptors is obtained from the cluster that best represents an individual photo — the component *event inference* uses these descriptors in addition to a domain event ontology that is selected according to *user info*. *Event Ontology Augmentation* derives the most relevant subevent categories to the input photo stream, and refines the derived categories by propagating their instances with the information extracted from *data sources*. The subevent tags are then validated using external sources. These tags are added to the event ontology (extension) — the extended event ontology is used in *filtering* that integrates *visual concept verification tool*. In this stage, first, irrelevant cluster branches are pruned. Next, for each matched cluster, less relevant photos to a subevent tag are filtered. The output is a set of photos labeled with some tags; these tags are then stored as new metadata for the photos. The remaining photos are tagged as *miscellaneous*.

E. Refinement, Validation, Extension

The inferred subevent categories E' are refined with the context data extracted from data sources in the bucket B, through the refinement process. First, let us elaborate this process by introducing the notion of *seed event*, which is an instance of an inferred category in E', which is not yet augmented with information. An augmented seed-event is an expressive event tag. The seed-event is continuously refined with information from multiple sources.

Our algorithm uses a similar strategy to what we described earlier in subsection IV-C. The only difference is that the attributes of a data source at each iteration is supplemented by the user information and the attributes of a seed-event (I) that is represented with the same schema that is described in the event ontology. Given a sequence of input attributes, if a data source returns an output-array of size K, then our algorithm creates K new instances of events with the same type as in the seed-event, and augments them with the information in the output-array. The augmented seed-events are added to I for the next iteration; I is constantly updated until all the event categories in E' are augmented, and/or there is no more data source (in the bucket B) to query. To avoid falling into an infinite loop of querying data sources, we set the following condition: a data source cannot be queried more than once

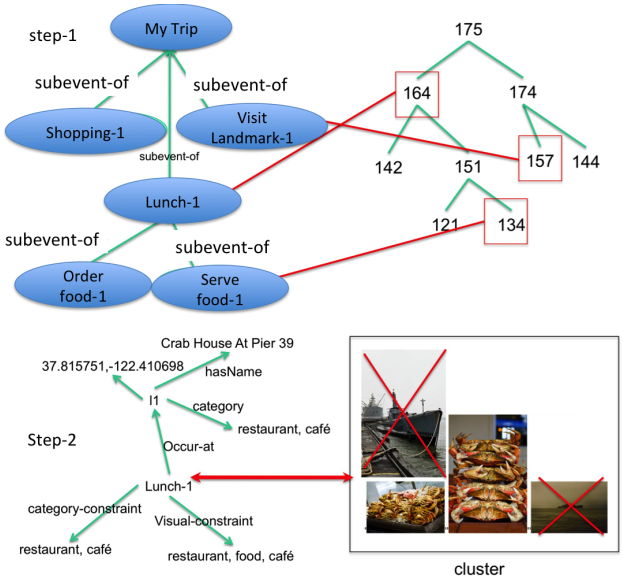

Fig. 7. Filtering Operation.

for each seed-event. Moreover, we defined some queries manually that are expressed through the relative spatiotemporal relationships in the event ontology, and the augmented seed-events; these queries are used to augment the seed-events with relative spatiotemporal properties. When a seed-event gets augmented with information, our technique validates the event tag by using the event constraints, augmented event attributes, and a sequence of entailment rules that specify the *cancel* status for an event. For instance, if the weather attribute for an event is *heavy rain*, and the weather constraint *fine weather* is defined for an event, then the status of the event tag becomes *canceled*; another example is when the place of occurrence related to the event tag is closed during its time of occurrence. After the validation, event tags are added to the domain event ontology by extending event classes through *typeOf* relationship. This step produces an augmented event ontology that is the extended version of the prior model, see fig 1.

V. Filtering

Filtering is a two-step process; during the first step, redundant and irrelevant clusters are pruned from the hierarchical cluster structure which was produced by the *clustering* component, see fig 7-step-1. Equation 5 describes the *prune-rule*, and *match-rule* that we use in this step. *traverse-rule* in equation 7 is used to visit cluster nodes— c implies cluster.

$$\neg Inside_{ST}(tag_e\ ,\ c) \rightarrow Prune(c). \quad (5)$$

$$Inside_{ST}(c\ ,\ tag_e) \rightarrow Match(c, tag_e). \quad (6)$$

$$Inside_{ST}(tag_e\ , c) \wedge hasChild(c) \rightarrow Traverse(c.child). \quad (7)$$

The second step filters redundant photos from the matched cluster, see fig 7-step-2. This is accomplished by applying the context and visual constraints of the expressive tag that is matched to the cluster. We used a concept verification tool[6] to verify the visual constraints of events using image features. This tool uses pyramids of color histogram and GIST features. Filtering operation is deeply guided by the expressive tags. During this operation, subevent relations are used for navigating the augmented event model. Expressive event tags are stored in *metadata-base*, as the new metadata for photographs.

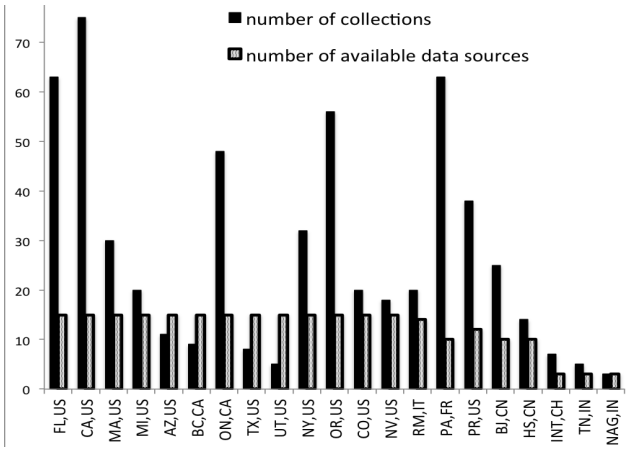

Fig. 8. Data set geographical distribution. The black bars show the number of albums in each geographic region, and the gray bars show the number of data sources that supported the corresponding geographic region.

VI. Experiments and Evaluations

We focused on the three domain scenarios vacation, professional trip, and wedding. First, we explain our experimental data set below.

1) Experimental Data set: We crawled Flickr, Picasaweb, and our lab data sets. Based on the assumption that people store their personal photos according to events,we collected the data sets based on time, space, and event types (like travel, conference, meeting, workshop, vacation, and wedding). We developed a Java-based crawler that uses Flickr's photo search api to download photos. We also used the public service ScraperWiki [7] to develop a crawler to download personal albums from Picasaweb. The crawlers were used to download about 700 albums of the day's featured photos. In addition, we crawled photo albums uploaded since the year 2010; the reason was that most of the older collections did not contain geo-tagged photos. After 4 months, we collected 84,021 albums (about 6M photos) from which only 570 albums (about 60K photos) had the required EXIF information containing location, timestamp, and optical camera parameters. We ignored the albums a) smaller than 30 photos, b) with non-English annotations. The average number of photos per album was 105. We used the albums from the most active users based on the amount of user annotation; we ended up with a diverse collection of 20 users with heterogeneous photo albums in terms of time period and geographical sparseness. The geographic sparseness of albums ranged from being across continents, to cities of the same country/state. Some of the users return to prior locations, and some do not. Fig 8 sketches the geographic distribution of our data set. We noticed that data sources do not equally support all the geographic regions; for instance,only a small number of data sources supported the data sets captured inside India. The photos for vacation/professional-trip domains have higher temporal and geographical sparseness compared to photos related to wedding domain. The number of albums for vacation domain exceeds the other two.

[6]http://socrates.ics.uci.edu/Pictorria/public/demo

[7]https://scraperwiki.com

2) Experimental Set-up: We picked the 4 most active users (based on the amount of user annotation) from our non-lab, downloaded data set, and 2 most active users from our lab data set (based on the number of collections they own). As ground-truth for the lab data set, we asked the owners to annotate the photos using their personal experiences, and an event model that best describes the data set, while providing them with three domain event ontologies (wedding, professional trip, and vacation). For the non-lab data set, the ground truth provides a manual and subjective event labeling done by the very owner of the data set being unaware of the experiments. Because of the subjective nature of the non-lab data set, the event types that were not contained in the event domain ontology are replaced with event type *miscellaneous* that is an event type in every domain event ontology in this work. For each experiment, we compute standard information retrieval measures (precision, recall, and F1-measure), for the event types used in tags. In addition to that, we introduce a measure of correctness for event tags. The score is obtained based on multiple context cues. For instance, label *meeting with Tom Johnson at RA Sushi Japanese Restaurant in Broadway, San Diego, during time interval "blah" in a sunny day, in an outdoor environment*, specifies type of the event, its granularity in the subevent hierarchy, place, time, and environment condition. We developed an algorithm that evaluates each cue with a number in the range of 0 to 1 as follows: *1)* event type: wrong = 0, correct = 1, somehow correct = $\frac{L_p}{L_{TP}}$ such that L_p is the subevent-granularity level for a predicted tag and L_{TP} is the subevent granularity level for the true-positive tag (the predicted tag is the direct or indirect superevent of the true-positive tag i.e., $\frac{L_p}{L_{TP}} \leq 1$); *2)* place: includes place name, category and geographical region. If the place name is correct, score 1 is assigned and the other attributes will not be checked. Otherwise, 0 is assigned; for the category and/or geographical region if correct, score 1 is assigned, and 0 otherwise. The average of these values represent the score for place; *3)* for weather, optical, and visual constraint: wrong=0, correct =1, unsure = 0.5; *4)* time interval: if the predicted event tag occurs anytime during the true-positive event tag, 1 is the score, otherwise 0. The average of the above scores represents the correctness measure for a predicted event tag. We introduce *average correctness* of annotation that is calculated using the formula in equation 8, in which w_j is the score for the j^{th} predicted event tag.

$$\overline{correctness} = \frac{\sum_{j=1}^{L} w_j}{L} \tag{8}$$

$$\overline{context} = 1 - \overline{Err} \tag{9}$$

We also introduced the metric $\overline{context}$ in equation 9 to measure the average context provided by data sources for annotating a photo stream. In this equation, parameter $\overline{Err}$ is the average error related to the information provided by data sources used for annotating a photo stream ($0 \leq \overline{Err} \leq 1$); the following guidelines are applied automatically, to measure this value: *(a)* if the information in a data source is related to the domain of a photo stream, but it is irrelevant to the context of the photo stream, assign error-score 1. For instance, data source *TripAdvisor* returns zero results related to *Things-To-Do* for the country at which a photo stream is created. Also, if a photo stream for a vacation trip does not include any picture taken in any landmark location, *TripAdvisor* does not provide any coverage; *(b)* assign error-score 0 if the type of a source is relevant as well as its data (i.e. non-empty results); *(c)* if the data from a relevant source is insufficient for a photo stream, assign error-score 0.5. For instance, only a subset of business venues in a region are listed in data source *Yelp*; as a result, the data source returns information for less than 30% of the photo stream; *(d)* finally, for a data source, multiply the error-score by a fraction in which the numerator is the number of photos tagged using this data source, and the denominator is the size of the photo stream. Do this for all the sources and obtain the weighted average of the error-scores. The result is the value for $\overline{Err}$. The implication of our result in fig 9 is as follows: while the correctness of event tags (for a photo stream of an event) peaks with the increase in $\overline{context}$, relatively, smaller percentage of photos are tagged using *non-miscellaneous* events, and larger percentage of photos are tagged using *miscellaneous* event. This means if the suitable event type for a group of photos does not exist in an event ontology, the photos are not tagged with an irrelevant *non-miscellaneous* event; instead, they are tagged with *miscellaneous* event which means *other*. The right side of the figure indicates that even though the number of miscellaneous and non-miscellaneous event tags does not change, the correctness is still increasing; this means that the tags get more expressive since more context cues are attached to them. The quality of annotations is increased when more context information is available. This shows that event ontology by itself is not as effective as augmented event ontology. We demonstrate three classes of experiments in table I. This table shows the average values (between 0 to 1) for the measure metrics discussed earlier (precision, recall, F1, $\overline{correctness}$). We use the work proposed in [18] as a baseline. It is based on space and time to detect event boundaries in conjunction with using English album descriptions. This baseline approach, with F1-measure about 0.6 and correctness of almost 0.56, shows promising results, and illustrates that time and space are important parameters to detect event boundaries. On the other hand, the baseline approach is limited to using only spatiotemporal containment for detecting subevent hierarchy, it does not support other types of relationships among events (like co-occurring events, relative temporal relationships) and other semantic knowledge about the structure of events. In addition, it requires human-induced tags which are noisy. For the second set of experiments, we use an event domain ontology without augmenting it with context information. This approach gives worse results since the context information is disregarded during detecting event boundaries. It provides the

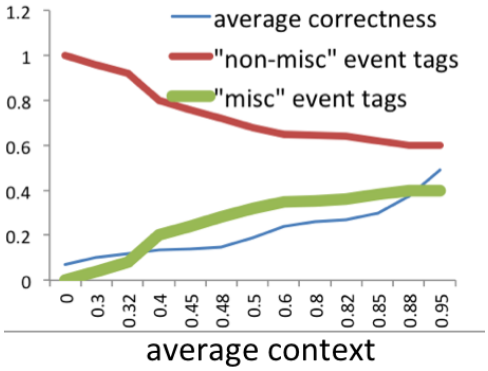

Fig. 9. Roll of context in improving the correctness of event tags.

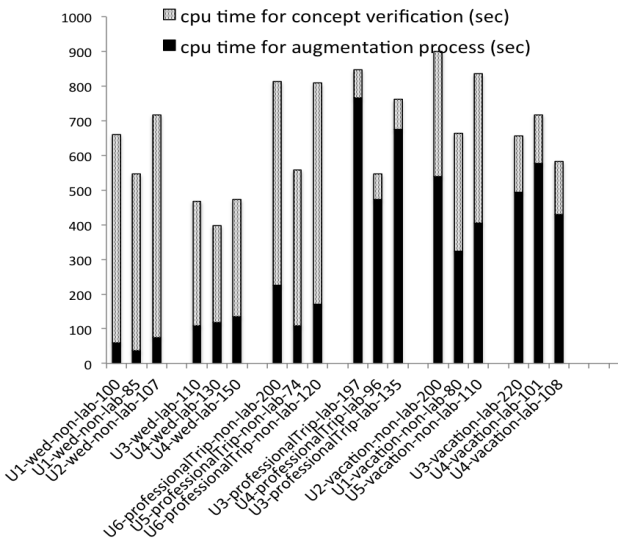

Fig. 10. CPU-Time for experimental data sets of the 6 most active users. Each data set is represented by its owner, domain type, source, and size. The domain *wed* implies *wedding* domain.

F1-measure of almost 0.32 and correctness of 0.13. Our last experiment leverages our proposed approach, and achieves F1-measure of about 0.85, and correctness of 0.82. Compared to our baseline approach, we obtain about 26% improvement in the quality of tags which is a very promising result.

3) CPU-Performance: We investigate the running time for event ontology augmentation, and visual concept verification in fig 10, through a two-stage process described below. Fig 10 illustrates the results for data sets of two sources i.e., lab, and non-lab (including Flickr, and Picasaweb), and three event domains.

Stage 1: Intra-domain comparison : In general, we found smaller number of context sources for wedding data sets compared to the other two domains; as a result, the event ontology augmentation process exits relatively faster, and the running time for the concept verification process increases. We observed the correctness of event tags degrades when event ontology augmentation process exists fast. This observation confirms the findings of fig 9.

Stage 2: Intra-source comparison: Within each domain, we compared the cpu-performance among lab and non-lab data sets. We noticed that the augmentation process exits relatively faster for non-lab data sets. The justification for this observation is that we could obtain user-related context like facebook events and check-ins from our lab users (U3, U4), but such information was missing in the case of non-lab data sets. This absence of information impacts wedding data sets the most, since the context information in the *wedding* scenario largely includes personal information such as guest list, and wedding schedule; such information is not publicly available on public photo sharing websites. In *professionalTrip* scenario, this impact is smaller than *wedding*, and larger than *vacation*; the missing personal information originates from the lack of context information related to personal meetings, and conference schedules. In *vacation* scenario, data sources are mostly public; only a small portion of context information comes from the user-related context such as flight information,and facebook check-ins; therefore, we did not find a significant change in the cpu-time between lab and non-lab data sets in the *vacation* domain.

Users		U1	U2	U3	U4	U5	U6
baseline[18]	prec	0.65	0.58	0.39	0.53	0.74	0.61
	recall	0.89	0.4	0.61	0.64	0.8	0.43
	f1	0.75	0.47	0.48	0.6	0.77	0.5
	corr	0.63	0.62	0.52	0.62	0.28	0.69
event ontology	prec	0.41	0.17	0.3	0.48	0.12	0.53
	recall	0.4	0.2	0.5	0.43	0.24	0.3
	f1	0.4	0.18	0.37	0.45	0.16	0.38
	corr	0.2	0.08	0.12	0.2	0.03	0.19
proposed	prec	0.74	0.83	0.95	0.92	0.88	0.79
	recall	0.91	0.93	0.88	0.7	0.97	0.82
	f1	0.81	0.88	0.91	0.79	0.92	0.8
	corr	0.8	0.75	0.85	0.79	0.9	0.88

TABLE I
EXPERIMENTAL RESULTS FOR AUTOMATIC PHOTO ANNOTATION FOR THE DATA SETS OWNED BY THE 6 MOST ACTIVE USERS.

VII. CONCLUSIONS

Our proposed technique addresses a broad range of both basic and applied research challenges to achieve a powerful event-based system that can adapt to different scenarios and applications such as those in intelligence community, multimedia applications, and emergency response. Facebook has recently launched the graph search feature to let the users search their content using high-level linguistic descriptions[8]; our proposed technique facilitates graph search by annotating photos with structured events in an automated fashion. Our experiments showed promising results when event models were combined with context-data from various sources. This is the starting step for combining complex models with big data.

REFERENCES

[1] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. In Journal of Logic and Computation, 1994.
[2] R. Alur and T. A. Henzinger. Logics and models of real time: A survey. In J. W. de Bakker, Cornelis Huizing, Willem P. de Roever, and Grzegorz Rozenberg, editors, REX Workshop, Springer, 1991.
[3] N. Brown. On the prevalence of event clusters in autobiographical memory. Social Cognition, 2005.
[4] L. Cao, J. Luo, H. Kautz, and T. Huang. Annotating collections of photos using hierarchical event and scene models. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. IEEE.
[5] M. Cooper, J. Foote, A. Girgensohn, and L. Wilcox. Temporal event clustering for digital photo collections. ACM Transactions on Multimedia Computing, Communications, and Applications, 2005.
[6] R. Datta, D. Joshi, J. Li, and J. Wang. Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys (CSUR), 2008.
[7] A. Fialho, R. Troncy, L. Hardman, C. Saathoff, and A. Scherp. What's on this evening? designing user support for event-based annotation and exploration of media. In 1st International Workshop on EVENTS-Recognising and tracking events on the Web and in real life, 2010.
[8] B. Gong and R. Jain. Hierarchical photo stream segmentation using context. Proceedings of SPIE, Multimedia content Access: Algorithms and System, 2008.
[9] B. Gong, U. Westermann, S. Agaram, and R. Jain. Event discovery in multimedia reconnaissance data using spatio-temporal clustering. In Proc. of the AAAI Workshop on Event Extraction and Synthesis, 2006.
[10] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as essence for photo browsing through personal digital libraries. In Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, pages 326–335. ACM, 2002.
[11] A. Gupta and R. Jain. Managing event information: Modeling, retrieval, and applications. Synthesis Lectures on Data Management, 2011.

[8]https://www.facebook.com/about/graphsearch

[12] S. Harada, M. Naaman, Y. Song, Q. Wang, and A. Paepcke. Lost in memories: interacting with photo collections on pdas. In Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries. ACM, 2004.

[13] R. Jain and P. Sinha. Content without context is meaningless. In Proceedings of the international conference on Multimedia. ACM, 2010.

[14] R. Koymans. Specifying real-time properties with metric temporal logic. In Real-Time Syst.,2(4), 1990.

[15] X. Liu, R. Troncy, and B. Huet. Finding media illustrating events. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval. ACM, 2011.

[16] M. Naaman, S. Harada, Q. Wang, H. Garcia-Molina, and A. Paepcke. Context data in geo-referenced digital photo collections. In Proceedings of the 12th annual ACM international conference on Multimedia. ACM, 2004.

[17] M. Naaman, Y. Song, A. Paepcke, and H. Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In Digital Libraries. Proceedings of the 2004 Joint ACM/IEEE Conference on. IEEE.

[18] J. Paniagua, I. Tankoyeu, J. Stöttinger, and F. Giunchiglia. Indexing media by personal events. In Proceedings of the 2nd ACM International Conference on Multimedia Retrieval. ACM, 2012.

[19] A. Pigeau and M. Gelgon. Building and tracking hierarchical geographical & temporal partitions for image collection management on mobile devices. In Proceedings of the 13th annual ACM international conference on Multimedia. ACM, 2005.

[20] S. Rafatirad, A. Gupta, and R. Jain. Event composition operators: Eco. In Proceedings of the 1st ACM international workshop on Events in multimedia. ACM, 2009.

[21] S. Rafatirad and R. Jain. Contextual augmentation of ontology for recognizing sub-events. In Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on. IEEE, 2011.

[22] P. Sinha and R. Jain. Classification and annotation of digital photos using optical context data. In CIVR, 2008.

[23] W. Viana, J. Bringel Filho, J. Gensel, M. Villanova-Oliver, and H. Martin. Photomap: from location and time to context-aware photo annotations. Journal of Location Based Services, 2008.

[24] W. Viana, J. Filho, J. Gensel, M. Villanova Oliver, and H. Martin. Photomap–automatic spatiotemporal annotation for mobile photos. Web and Wireless Geographical Information Systems, 2007.