Context Correlation Using Probabilistic Semantics

Setareh Rafatirad
George Mason University
Email: srafatir@gmu.edu

Kathryn Laskey
George Mason University
Email: klaskey@gmu.edu

Paulo Costa
George Mason University
Email: pcosta@gmu.edu

***Abstract*—We present an approach for recognizing high-level geo-temporal phenomena – referred as events/occurrences– from in-depth discovery of information, using geo-tagged photos, formal event models, and various context cues like weather, space, time, and people. Due to the relative availability of information, our approach automatically obtains a probabilistic measure of occurrence likelihood for the recognized geo-temporal phenomena. This measure, however, is not only used to find the best event among the merely possible candidates – witnessing the data (including photos), but it can also provide informative cues to human operators in the environments where uncertainty is involved in the existing knowledge.**

I. INTRODUCTION

Sensors have become one of the biggest contributors of BIG DATA datasets. Numerous datasets have been already generated in real-time with rich content, about various information. Mobile wireless devices with multiple sensors like camera and GPS, and internet connectivity, can continuously capture photos and record camera parameters, GPS location, and time. The availability of various web services like *MapMyRide* [1], and *Wunderground* [2], provides semantics like ride, and geo-temporal weather status logs, using the captured sensory data. Given that context data exists in massive volumes, an information management paradigm is needed to correlate the information and infer higher level semantics. We propose a technique that automatically correlates various information, and creates a context-aware event graph by combining event models with contextual information related to photos, sensor logs, heterogeneous data sources, and web services. Our technique automatically computes the occurrence-likelihood for the event nodes in the output graph – referred as plausibility measure that provides informative cues to human operators in uncertain environments to make better decisions. Note that this work provides a holistic view of the high-level events witnessed by a dataset; further cause-effect decision-making using the output of this stage in out of the scope of this paper.

Events, in general, are structured and their subevents have relatively more expressive power [13]. In this work, an event model (or event ontology) provides a multi-granular conceptual description, i.e., it provides conceptual hierarchy in multiple levels using containment event-event relationships e.g., *subevent-of*, and *subClassOf*. In addition, event types can have multiple instances; instance events are contextual, and they should be augmented with context cues (like place, time, weather). This makes instance events more expressive than event types. Augmenting an instance event with context cues adapts a concept to multiple contextual descriptions (e.g., event type *visit-landmark* may have two instances; one instance associated with *World War II Memorial* and the other to *Washington Monument*). Consider the following example: A person takes a photograph at an airport less than 1 hour after his flight arrives. To explain this photograph, we first need the background knowledge about the events that generally occur in the domain of a trip. These semantics can only come from an event-ontology that provides the vocabulary for event/entity and event relationships related to a domain. An event-ontology 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 Extension*. It constructs a more robust and refined version of an event-ontology either fully or semi-automatically. Secondly, given the uncertain nature of sensory data (like GPS that is not always accurate), the event type witnessed by the available context data is not decisive; in the above example, the event might either be *rent a car*, or *baggage claim* that are two possible conclusions — 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 (observations) to the most related conclusion out of all possible ones (explanations) is performed through a ranking algorithm that incorporates the plausibility measure; this ranking process is used in *Event Ontology Extension*.

Problem Formulation: Every input photo has context information (timestamp, location, and camera parameters) and a user. Each photo belongs to a photo stream P of an event with a domain event model $O(V, E)$ –handcrafted by a group of domain experts– whose nodes V are event/entity classes, and edges E represent the relationships between the nodes. There is a bucket B of external data sources represented with a schema. The sources can be queried using the metadata of the input photographs and other available information, including the 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 photos in P?

Solution: We propose an Event Ontology Extension technique described as follows: select a relevant domain event model through the information related to both P and the user. Using P, B, O, and the user information, infer S – that consists of the best relevant subevent categories to P– where $S \subseteq V$. A member of S is the most plausible event category for a group of contextually-similar photos. For a group of similar photos c_j, a function f calculates the plausibility measure m^p_{ij} for every competing event candidate s_i: $f(s_i, c_j) = m^p_{ij}$; this measure indicates how much s_i is relevant to c_j such that

[1] http://www.mapmyride.com/
[2] http://www.wunderground.com/

$c_j \subset P$. Using the information from B, extend S with one or more augmented instances of S, and obtain expressive event tags T. An event tag $t_i^e \in T$ is 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. If t_i^e is an assignable tag to any photo, and $t_i^e \notin O$, we intend to extend O by adding t_i^e to O such that the constraints governing O are preserved. The output is an extension to O that is referred as O_r (see fig 1). 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. Fig 3 depicts the processing components of our proposed approach. We used semantics such as spatiotemporal attributes/constraints of events, subevent structure, and spatiotemporal proximity. In contrast to 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 external data sources including public data sources (like public event/weather directories, local business databases), and digital media archives (like photographs). With this knowledge expansion, new infrastructures are constructed to serve relevant data to communities. 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)* A sufficiently flexible structure to express context attributes for events such that the attributes are not hardwired to events, but rather they are discovered on the fly. This feature does not limit our approach to a single data set; *2)* leveraging context data across multiple sources could facilitate building a consistent, unambiguous knowledge base.

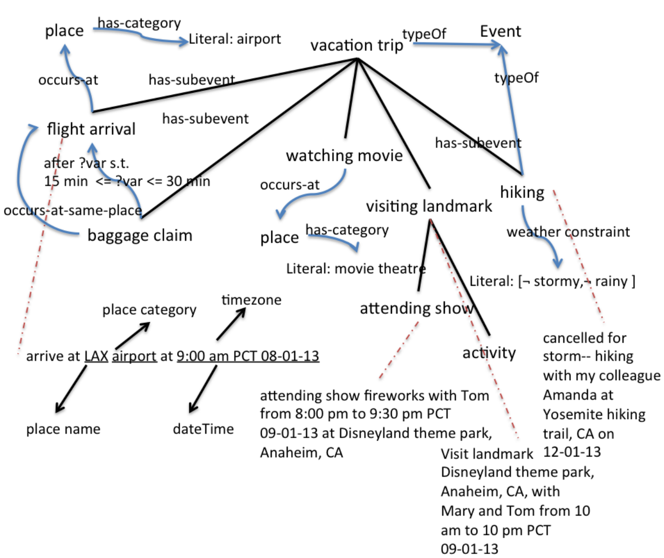

Fig. 1. An example of an event model being extended with contextually propagated instances.

Some of the main challenges of this work are: *a)* collecting and correlating information from various sources – we need a general mechanism that automatically queries sources and represents the output; *b)* a validation mechanism to ensure the coherency of the obtained data; *c)* currently, publicly available benchmark data sets such as those offered by TRECVid do not suit the purpose of this research (they deal with low level events i.e.,activities). However, higher-level events have relatively more contextual characteristics; *d)* according to the useful properties of photos, relevant event categories in the model must be discovered. This paper is organized as follows: in section II, we review the prior art that use context and event models for annotating photographs; in section III and IV, we explain our solution strategy; this is followed by section V that demonstrates our experiments, and section VI which is the conclusion.

II. State Of Art

The important role of context is emphasized in [9]. Context information and ontological event models are used in conjunction by [16], [6]. 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]; this work, does not support subevent structure. Liu et al. reports a framework that converts each event description from existing event directories (like Last.fm) into an event ontology that is a minimal core model for any general event [11]. This approach is not flexible to describe domain events (like *trip*) and their subevent structure. Paniagua et al. propose an approach that builds an event hierarchy 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 [12]. 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 structure that is connected by causal, temporal, spatial and subevent relations. If these aspects are carefully modeled, they can be used to create a descriptive knowledge base for interpreting multimedia data. In [14], a 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, the quality of annotation degrades.

III. Event Ontology Extension

Photo's incomplete information can be improved if combined with the information related to a group of similar photos. In this work, two images are similar if they belong to the same event type. Partitioning a photo stream of an event based on the context of its digital photographs can create separate subevent boundaries for its photos [5]. An event is a spatiotemporal entity [7]. In addition, optical camera parameters (CP) in photos provide useful information related to the environment (like *outdoor*) at which an event occurs [15]. We used a clustering that partitions photos hierarchically based on their timestamp, location, and CP. We used single linkage clustering and Euclidean distance in our clustering technique. However, one can use other approaches and refine the results. We present the observations (i.e., photos/clusters) with a set of descriptors – 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) 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 [8] 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 [10], [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 $\Box_{[t_1,t_2]}(e_1 \rightarrow \Diamond_{[t_2,t_2+1800]} e_2 \wedge \mathcal{D}(e_2) \leq 1800)$ 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 Model: A domain event ontology provides specialized taxonomy for a certain domain like *trip*, see fig 2. 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 2, 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*[3], *Distance*[4], and *HoursOfOperation*[5]. 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 [15]. If a descriptor value is 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 $g(p_j, d)$ 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.

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

Descriptor Consistency: Consistency among a set of descriptors is a mandatory condition to infer the best possible conclusion from it. In this work, consistency must exist 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

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

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.

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

(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.

C. 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[6]. A 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 continues until no more source with matching attributes is left in the bucket B.

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

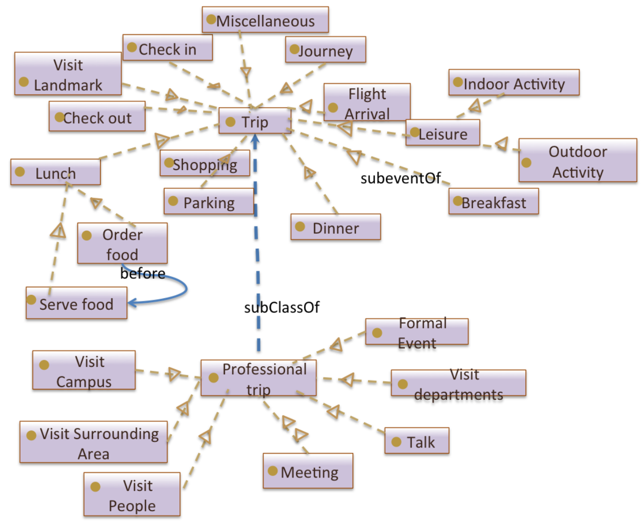

Fig. 2. An event ontology for the domain *professional trip*.

D. EVENT INFERENCE

From a set of consistent cluster descriptors (*observations*), we developed a *context discovery* 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. 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*, and *after* — 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.

To find the most plausible subevent category, we introduce *Measure of Plausibility* (m_{ij}^p) to rank event candidates. This measure is computed using two parameters (1) granularity score (w_g), and (2) plausibility score (w_{AX}). w_g is equivalent to the level of the event in the subevent hierarchy in the domain event ontology. To compute w_{AX}, we used 'plausibility-weight' (w^+) and 'implausibility-weight' (w^-) which are two fields of a cluster descriptor. 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 III-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_{ij}^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, $e_i.m_{ij}^p = e_i.w_{AX}$. The candidate with the highest m_{ij}^p is the most plausible subevent category.

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

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.

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.

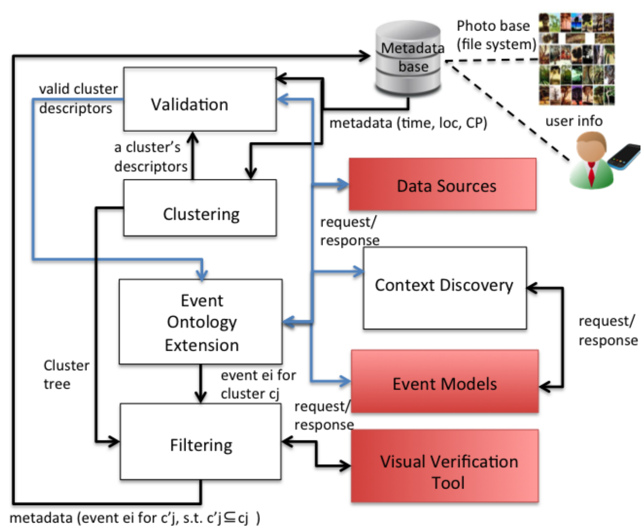

Fig. 3. The Big Picture. Photos and their metadata are stored in *photo-base* and *metadata-base* respectively. Using *user info*, including events' type, time, and space in a user's calendar, a photo stream is queried, and its metadata is passed to *Clustering*. In *Validation*, a set of consistent descriptors is obtained from the cluster that best represents an individual photo — the component *event inference* uses these descriptors in addition to a domain model that is selected according to *user info*. *Event Ontology Extension* propagates the most relevant subevent categories (to the input photo stream) with the information discovered from *Data Sources*, then extends the event structure (ontology) with the applicable propagated event instances (i.e., tags). The tags are validated (using data sources), and added to the event ontology – the extended event ontology is used in *filtering* that queries *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*.

Our extension algorithm uses a similar strategy as what we used in subsection III-C. The difference is, 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 for each seed-event. 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*. 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).

IV. FILTERING

Filtering is a two-step process; (1) redundant and irrelevant clusters are pruned from the hierarchical cluster structure produced by the *clustering* component, see fig 4-step-1. (2) filter redundant photos from the matched cluster, see fig 4-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[7] 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.

V. EXPERIMENTAL EVALUATIONS

We focused on 3 domain scenarios vacation, professional trip, and wedding. We crawled Flickr, Picasaweb, and our lab data sets. We observed that many people store their personal photos according to events; accordingly, we collected the data sets based on time, space, and event types (like travel, conference, meeting, workshop, vacation, and wedding). We developed some crawlers to download about 700 albums of the day's featured photos; we crawled photo albums created since the year 2010 since most of the older collections did not contain geo-tagged photos. After 4 months, we collected 570 albums (about 60K photos) which had the required EXIF information containing location, timestamp, and optical camera

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

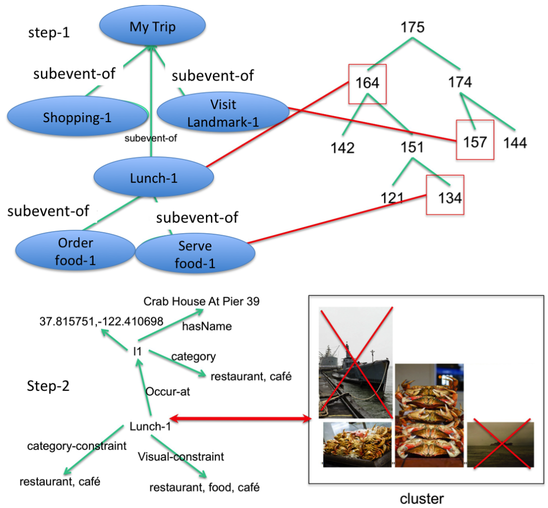

Fig. 4. Filtering Operation.

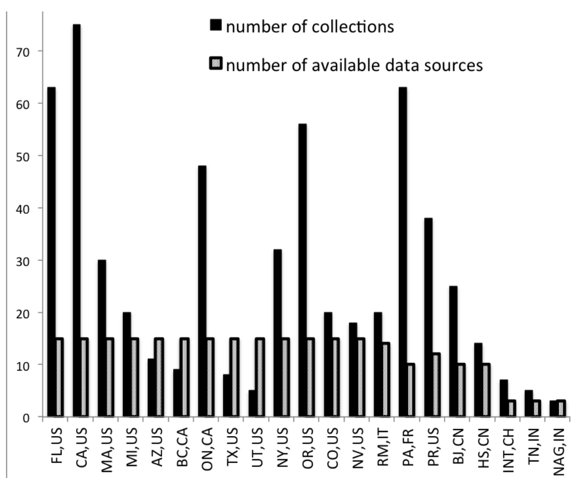

Fig. 5. Data set geographical distribution. The black bars show the number of albums in each geographic region, and the gray bars show the number of data sources that supported the corresponding geographic region.

parameters. We ignored the albums a) smaller than 20 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 annotations, ending up with a 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 (see fig 5). We noticed that data sources do not equally support all the geographic regions; e.g., 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.

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

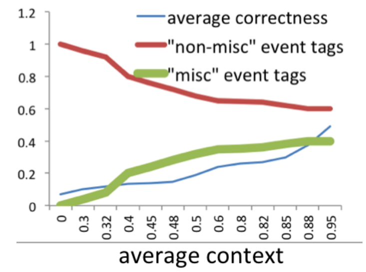

Fig. 6. Role of context in improving the correctness of event tags.

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 models. 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 5, where w_j is the score for the j^{th} predicted tag.

$$\overline{correctness} = \frac{\sum_{j=1}^{L} w_j}{L} \; ; \; \overline{context} = 1 - \overline{Err} \qquad (5)$$

The metric $\overline{context}$ in equation 5 is used to measure the average context provided by data sources for annotating a photo stream; 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)* 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 $\overline{Err}$. The implication of our result in fig 6 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 (Paniagua, 2012) 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, 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. Also, 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 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.

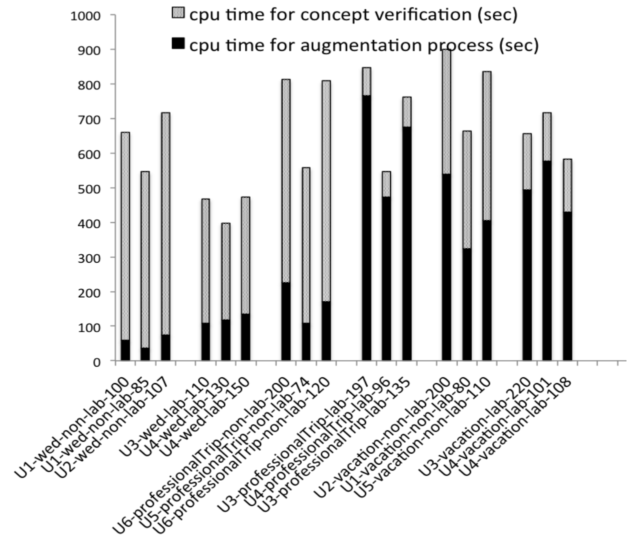

Fig. 7. CPU-Time for experimental data sets of the 5 most active users. Each data set is represented by its owner, domain type, source, and size. The domain *wed* implies *wedding* domain.

CPU-Performance

The running time for our proposed approach, and visual concept verification is shown in fig 7, which illustrates the results for data sets of two sources i.e., lab, and non-lab (including Flickr, and Picasaweb), and three event domains.

Cross-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 extension 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 Extension* process exists fast. This observation confirms the findings of fig 6.

Cross-Source Comparison: Within each domain, we compared the cpu-performance among lab and non-lab data sets; Event Ontology Extension exits relatively faster for non-lab data sets. The justification for this observation is that we could obtain user-related context like facebook events/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 that are not publicly available on photo sharing websites. In *professionalTrip* scenario, this impact is smaller than *wedding*, and larger than *vacation*; the missing data is due to 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.

VI. CONCLUSIONS

Our proposed technique addresses a broad range of research challenges to achieve a powerful event-based system that can adapt to different scenarios and applications like

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

TABLE I. RESULTS FOR AUTOMATIC PHOTO ANNOTATION FOR THE DATA SETS OWNED BY THE 5 MOST ACTIVE USERS.

those in intelligence community, multimedia applications, and emergency response. 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] 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.

[7] 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.

[8] A. Gupta and R. Jain. Managing event information: Modeling, retrieval, and applications. Synthesis Lectures on Data Management, 2011.

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

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

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

[12] 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.

[13] 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.

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

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

[16] 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.