2014 TRECVID MULTIMEDIA EVENT DETECTION & RECOUNTING EVALUATION PLAN

This is the evaluation plan for Multimedia Event Detection and Retrieval (MED) and Multimedia Event Recounting (MER) tracks of the 2014 TRECVID evaluation. For the purposes of these evaluations, "multimedia" is publicly available **videos** like those uploaded to Internet video sites. An "**event**" is a complex activity occurring at a specific place and time involving people interacting with other people and/or objects. An event consists of a number of human actions, processes, and activities that are loosely or tightly organized and that have temporal and semantic relationships to the overarching activity. All events are directly observable.

For MED/MER, participating teams create a prototype system that quickly finds events in a large collection of **search videos** and recounts evidence in each search video for the event. Events are described by 1) a textual event description and 2) a set of training **exemplar videos**. Teams are permitted to translate the event description into a **semantic query** that describes how **evidence** for the event should be combined to form event scores for the search videos. An **event query** combines the semantic query with an event detector formed from the exemplar videos. During an event search, the system will compute an event score and a recounting for each search video. The **recounting** 1) points to the **key evidence** found in the search video and 2) shows how the evidence was combined to form the event score. Using the event scores, the system **ranks** the search videos, from best match to worst, to be presented to the user. The system also computes a **threshold** to separate search videos containing the event from those that do not. NIST will provide a command line Input and Output Server (**I/O Server**) to provide inputs and receive outputs from the participating teams during the evaluation. A more detailed description of the prototype system and its interaction with the I/O Server is given in Section 1.

NIST will provide data collected and licensed[1] by the Linguistic Data Consortium (LDC) to all participating teams to train, test, and evaluate their prototype systems. Teams may NOT use any additional data for event training, testing, and evaluation and may NOT change the training, testing, and evaluation data in any manner.

The MER evaluation will be conducted with the help of human judges. The event query and recountings will be assessed by 1) how well the semantic query captures relevant and key evidence for the event, 2) how well the prototype system tags evidence in videos, 3) how well the key evidence is localized in videos, 4) how well does the found evidence convince the judge of the occurrence of the event in the video, and 5) how compact the recounted evidence is compared with the length of the video. NIST will provide a MER Workstation to visualize the event queries and the recountings for the judges. Participants are encouraged to improve the MER Workstation and to provide feedback to NIST. The best suggestions will be integrated into the final MER Workstation used by the judges for the evaluation. The MER evaluation process and the MER Workstation will be described in Section 2. Teams may participate in MED without participating in MER.

Teams are also encouraged to participate in a Semantic Query Editing Evaluation Pilot. For this pilot,

[1] See http://www.nist.gov/itl/iad/mig/med14.cfm for data format, licensing, and acquisition instructions.

judges will edit the participants' semantic query for a MED/MER event to create a semantic query for a new event (which will be called a pilot event). Since the judge will not have full access to the full query language for the system, the pilot event will be a sub-event for an existing MED/MER event. This pilot evaluation will be discussed in Section 3.

The MED/MER evaluations consists of several component evaluations: 1) Dry Run, 2) Pre-Specified evaluation, 3) Ad-Hoc evaluation, and 5) the Semantic Query Editing Evaluation Pilot. The definitions of the **Pre-Specified (PS) events** will be provided early in the evaluation so that teams may develop specific terms in their semantic query language that describe components of these events. The PS events will re-use events from previous MED/MER evaluations. On the other hand, for the **Ad-Hoc (AH) events**, the events will not be disclosed until immediately before they are processed by the teams' prototype systems. A schedule for all of the MED/MER evaluation tasks is detailed in Section 4.

All teams must participate in both the Pre-Specified and the Ad-Hoc (AH) evaluation that uses 10 exemplars in the event training set for either the full MED14 evaluation or MED14 evaluation subset. All other evaluations are optional. Teams are strongly encouraged to participate in the Dry Run evaluation to ensure that their system can correctly interact with the I/O Server for the tightly-timed Ad-Hoc evaluations. Teams are welcome to participate in the MER and Pilot evaluations, but these are not required.

The MED and MER evaluations are open to all who find the task of interest and who are willing to abide by the rules of the evaluation.

1 System Modules and their Interaction with the I/O Server

Each team participating in MED/MER will create a software prototype composed of four modules: 1) the Metadata Generator, 2) the Semantic Query Generator, 3) the Event Query Generator, and 4) Event Search. The I/O Server provides input to these modules and receives their outputs throughout the evaluations. I/O Server will have a command line API for manual or automatic interaction. The I/O Server will allow teams to run their modules at their locations on their computer hardware. Teams will specify which options they choose to NIST. NIST will control the ordering of the module calls and collect timing and hardware information for each module call.

Each module's interaction with the I/O Server can be represented in a functional notation:

module (inputs from I/O Server) -> outputs to be sent to I/O Server

Variables used this notation will be depicted as <variable>. Optional inputs/outputs are shown in grey text.

The following sections describe each of the four modules in more detail.

1.1 Metadata Generator (MG)

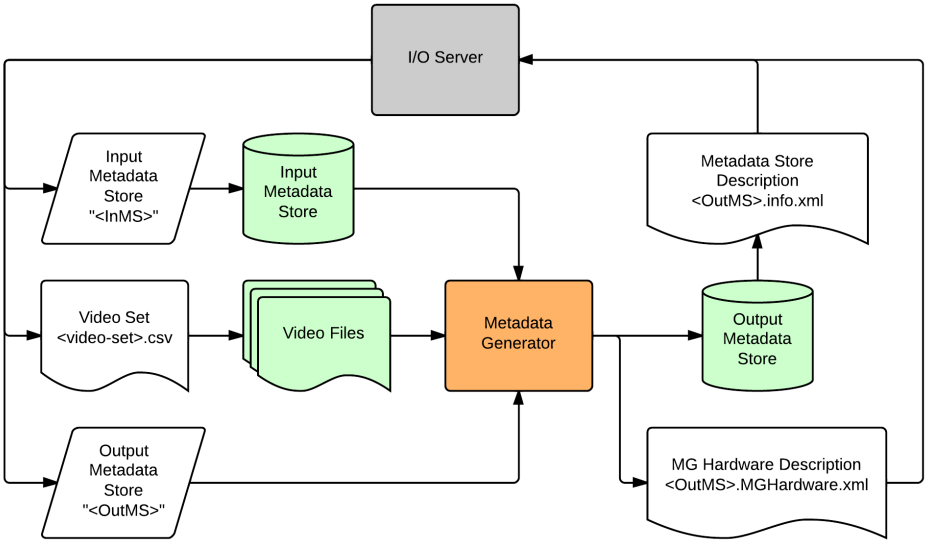

Figure 1.1.1: Metadata Generator (MG) Block Diagram

MG ("<InMS>", video_set.csv, "<OutMS>") -> <OutMS>.MGHardware.xml, <OutMS>.info.xml

A block diagram and functional notation for the Metadata Generator (MG) is shown in Figure 1.1.1. The Metadata Generator extracts metadata from videos in the input video set and adds them to the input metadata store creating a NEW output metadata store (leaving the input metadata store untouched). Since the metadata stores and video sets are too large to be passed to and from the I/O Server, video set list files (named in <video-set>.csv) and metadata store names ("<InMS>" and "OutMS") will be passed as a proxy. The module inputs the name of the input metadata store (<InMS>) that it should copy and update to create the output metadata store (named <OutMS>). If the Input metadata store name is an empty string, the output metadata store is created solely from the provided video set. Note, teams must strictly observe the input and outputs of the MG. No other inputs are to be used. **Teams must NOT manually annotate or alter the Video Files in any manner.**

The <video-set>.csv lists the videos to be added to the metadata store. It is a Comma Separated Value (CSV, exact format specified in Appendix C.2) file with the following columns:

- **Video ID**: a unique identification number for the video
- **Filename**: the unique filename of the video

NIST/LDC will provide the videos to the teams via SATA (or equivalent) disk format well before the run of the MG.

The Metadata Generator Hardware Description <OutMS>.MGHardware.xml is a required output of the Metadata Generator. It describes the hardware that was actually used in processing the metadata generation. This can include the average number of cores that were used during processing, the number of GPUs, etc. The exact format of this file is given in Appendix C.6.

The Metadata Store Description <OutMS>.info.xml is a required output of the Metadata Generator. It has the following fields (exact format specified in Appendix C.1):

- **Signal Metadata Size**: Size of the signal layer of the metadata store on disk in gigabytes (GB)

- **ASR_OCR Metadata Size**: Size of the ASR-OCR tag layer of the metatdata store on the disk in gigabytes (GB)
- **OCR Languages**: Languages for which embedded video text is transcribed via Optical Character Recognition (OCR)
- **ASR Languages**: Languages for which automated speech recognition (ASR) is translated in the video
- **Semantic Metadata Tags**: Concepts that are tagged in the video creating a semantic description of the video. In other words, the list of the concept detectors that the metadata generation will use to tag videos (not the actual tags on the video).
- **Tag Metadata Size**: Size of the signal layer of the metadata store on disk in megabytes (MB)
- **Video List**: List of video IDs represented in the Metadata Store

The Metadata Generator (MG) tags videos based on their content so that an event search can happen much more quickly. These tags identify entities in videos to be matched to event queries and provide evidence for the recountings. The metadata tags can be semantic or non-semantic. Non-semantic tags are typically feature vector codes that describe the video to the system but are unintelligible to a user. Semantic tags are designed by each team to capture key entities that can be used as evidence for event search. A metadata tag can be used directly for search and recounting in MED/MER if it contains the information shown in Table 1.1.1.

Semantic Metadata Tags

• **Video ID**: identification code of the video being tagged
• **Name**: name of the entity being tagged
• **Score**: a confidence score
• **Snippet**: the location of the entity being tagged in space and time
  • **Start**: the start time of the snippet
  ▪ **Start Upper Left**: the upper left corner of the bounding box surrounding the entity at the start time
  ▪ **Start Lower Right**: the lower right corner of the entity at the start time
  • **End**: the end time of the snippet
  ▪ **End Upper Left**: the upper left corner of the entity at the start time
  ▪ **End Lower Right**: the lower right corner of the entity snippet

Non-Semantic Metadata Tags
• **Code:** the feature vector code word that describes the video

Table 1.1.1: Minimal Metadata Tag Format

Table 1.1.2 shows the Metadata Store Names, the video set files, a description of the video sets and their approximate size that will be used as part of the 2014 MED/MER Evaluations. For the evaluations, teams will choose if they will process the MED14-EvalFull or MED14-EvalSub.

Output MS Names	Input Video Set File	Description of the Video Set	Number of Videos	Hours of Video
Event-BG	Event-BG.csv	Background event training videos used for all evaluations (Dry Run, PS, and AH)	5000	200
PS-Training	PS-Training.csv	The "positive" and "near-miss" event exemplar videos for the Dry Run and PS evaluations	2000	80
AH-Training	AH-Training.csv	The "positive" and "near-miss" event exemplar videos for the AH evaluation	1000	40
MED14-Test	MED14-Test.csv	Search videos for the Dry Run	23,000	960
MED14-EvalFull **MED14-EvalSub**	MED14-EvalFull.csv (MED14-EvalSub.csv)	Search Videos for the evaluations	200,000 (32,000)	8000 (1280)

Table 1.1.2: Video Sets for the 2014 MED/MER evaluations

1.2 Semantic Query Generator (SQG)

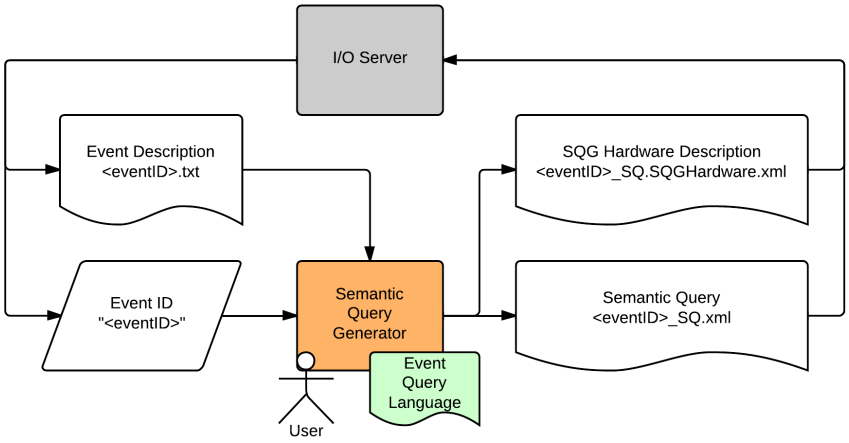

Figure 1.2.1: Semantic Query Generator Block Diagram

SQG ("<eventID>", <eventID>.txt) -> <eventID>_SQ.SQGHardware.xml, <eventID>_SQ.xml

The Semantic Query Generator (SQG), shown in Figure 1.2.1, translates an event description (<eventID>.txt) into the system's event query language building the Semantic Query (SQ) that includes the copied <eventID> string. This module may have a user help with this translation or may perform the translation automatically. Teams can choose not to use any semantics in their queries. In this case, the Semantic Query output should be an empty query (see Appendix A for the exact format). However, to participate in the MER evaluation or Semantic Query Editing Evaluation Pilot, teams will need to create a Semantic Query.

The Hardware Description <eventID>_SQ.SQGhardware.xml describes the hardware that was actually used in processing this run of Semantic Query Generation. SQG must be run on a single COTS workstation or on a single core of a cluster. The format for this file is given in Appendix C.6.

To be compatible with recounting and the MED/MER evaluations, the Semantic Queries must:

- Specify what key evidence the system will use to detect the event and how the this evidence should be combined to form the event score
- Be clear and concise and understandable by an English-reading judge.

A notional Semantic Query for the Board Trick event is shown in Table 2.2 (see Appendix A for the XML format of this example). This example is *only* for illustration and is *not* meant to specify how a team should design their semantic queries.

Semantic Board Trick [E001] = **max** (Board Trick Object**,** Board Trick Activity in a Board Trick Scene)
○ Board Trick Object = **max** {*surfboard*, *skateboard*, *snowboard*}
○ Board Trick Action in a Board Trick Scene = **min** {Board Trick Action, Board Trick Scene}
○ Board Trick Action = **max** {*person jumping*, *person flipping* }
○ Board Trick Scene = **max** {*ocean scene*, *city scene*, *snow scene*}

Table 1.2.1: Notional Semantic Query (for illustrative purposes only): The italicized text denotes metadata tags that should be found as key evidence for the event. Bold text signifies functions that combine the key evidence scores. Plain text denotes variables in the semantic query.

NIST will provide 5 event descriptions for the Dry-Run and 20 event descriptions for the Pre-Specified evaluations as shown in Table 1.2.2. For the Dry Run, teams must process <event> = (E031, …, E035). For the Pre-Specified evaluation teams must process the PS12 and PS13 event sets, i.e., <eventID> = (E021, …, E040). For the Ad-Hoc evaluation, teams must process the AH14 event set, i.e., <eventID> = (E041, …, E050). The names and descriptions of these events will be provided to the teams immediately before processing the events.

Pre-Specified Testing and Evaluation Events	
PS12: E021-E030	**PS13: E031-E040**
E021 - Bike trick	E031- Beekeeping
E022 - Cleaning an appliance	E032 – Wedding shower
E023 - Dog show	E033 – Non-motorized vehicle repair
E024 - Giving directions	E034 – Fixing a musical instrument
E025 - Marriage proposal	E035 – Horse riding competition
E026 - Renovating a home	E036 – Felling a tree
E027 - Rock climbing	E037 – Parking a vehicle
E028 - Town hall meeting	E038 – Playing fetch
E029 - Winning race without a vehicle	E039 – Tailgating
E030 - Working on a metal crafts project	E040 – Tuning a musical instrument

Table 1.2.2: Pre-Specified Evaluation Events

1.3 Event Query Generator (EQG)

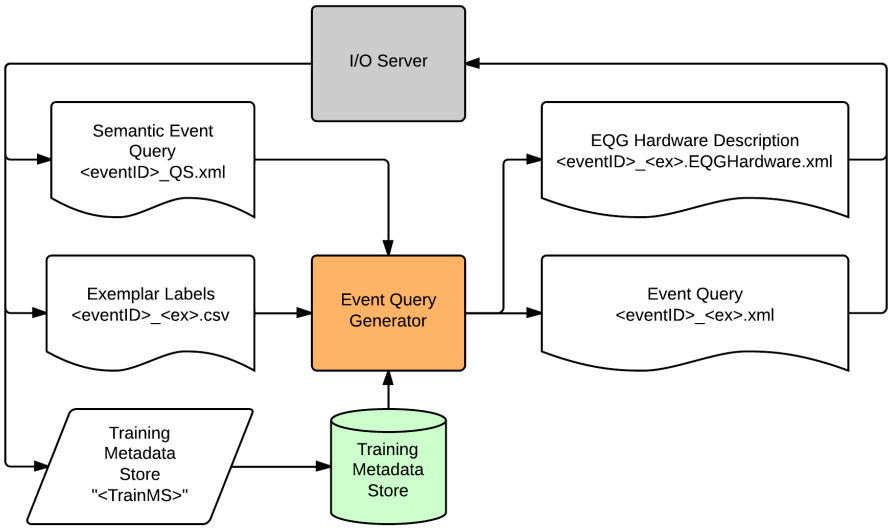

Figure 1.3.1: Event Query Generator Block Diagram

EQG (<eventID>_SQ.xml, <eventID>_<ex>.csv, "<TrainMS>") -> <eventID>_<ex>.EQGHardware.xml, <eventID>_<ex>.xml

A block diagram of the Event Query Generator (EQG) and its functional notation is shown in Figure 1.3.1. The Event Query Generator inputs the Semantic Event Query (<eventID>_SQ.xml) (created by the team's SQG) and Exemplar Labels (<eventID>_<ex>.csv) that specifies which videos (whose metadata is saved in the in the Training Metadata Store ("<TrainingMS>")) are examples of "positive", "near-miss" or "background" exemplars for the event. Teams must adhere strictly to the input/output of the EQG. No other inputs are permitted. **Only those videos specified in the Exemplar Labels list are to be used by the EQG, even though additional videos may exist in the Training Metadata Store. Teams may not edit the Exemplar Labels list. The EQG must use the metadata store and not access the original videos.**

The Exemplar List (<eventID>_<ex>.csv, exact format specified in Appendix C.3) has the following fields:

- **Event ID**: the id for the event (e.g., E021)
- **Exemplar Set**: the exemplar set (e.g., 010Ex)
- **Video ID**: the id for the video (same as the Video ID used video_set.csv input to the MG)
- **Label**: either "positive", "near-miss", or "background"

The Hardware Description <eventID>_<ex>.EQGhardware.xml describes the hardware that was actually used in processing this run of Event Query Generation. EQG must be run on a single COTS workstation or on a single core of a cluster. The format for this file is given in Appendix C.6.

The Event Query Generator creates an Event Query that combines an event detector(s) (trained from the exemplars) with the Semantic Query.

The detector(s) can be added to the semantic query as special node(s) with the following fields:

- **name**: is the detector name
- **private**: all the detector's parameters, or a reference to a set of parameters, for a private function needed to compute the event score for a video based solely on the training video metadata specified by the Exemplar Labels.

A notional example of a combined board trick query, integrating the Detector and the Semantic Query (from Table 1.2.1) is shown in Table 1.3.1. Note that this is only meant to be an illustrative example of what may be done for combining the semantic and detector queries. This is not meant to prescribe how teams should do the combination.

Board Trick [E001] = **max** (Semantic Board Trick, Detected Board Trick)
- Detected Board Trick = **private**
- Semantic Board Trick = **max** (Board Trick Object**,** Board Trick Activity in a Board Trick Scene)
  - Board Trick Object = **max** {*surfboard*, *skateboard*, *snowboard*}
  - Board Trick Action in a Board Trick Scene = **min** {Board Trick Action, Board Trick Scene}
    - Board Trick Action = **max** {*person jumping*, *person flipping* }
    - Board Trick Scene = **max** {*ocean scene*, *city scene*, *snow scene*}

Table 1.3.1: Notional Example of a Event Query (for illustrative purposes only): The italicized text denotes metadata tags. Bold text signifies functions that combine scores. Plain text denotes variables in the semantic query.

For the evaluations, NIST will provide teams with three sets of Exemplar Labels (<ex>) for each event (i.e., <eventID> = E021 … E050). Table 1.3.2 shows the numbers of positive, near-miss and background exemplars available for each Exemplar Set:

<ex>	Positive Exemplars	Near-Miss Exemplars	Background Exemplars
000Ex	0	0	5000
010Ex	10	5	5000
100Ex	100	50	5000

Table 1.3.2: Exemplar Sets for the 2014 MED/MER Evaluations

All teams must process the <ex> = 010Ex input to participate in the MED/MER evaluations. Teams may optionally process the <ex> = (000Ex and 100Ex).

1.4 Event Search (ES)

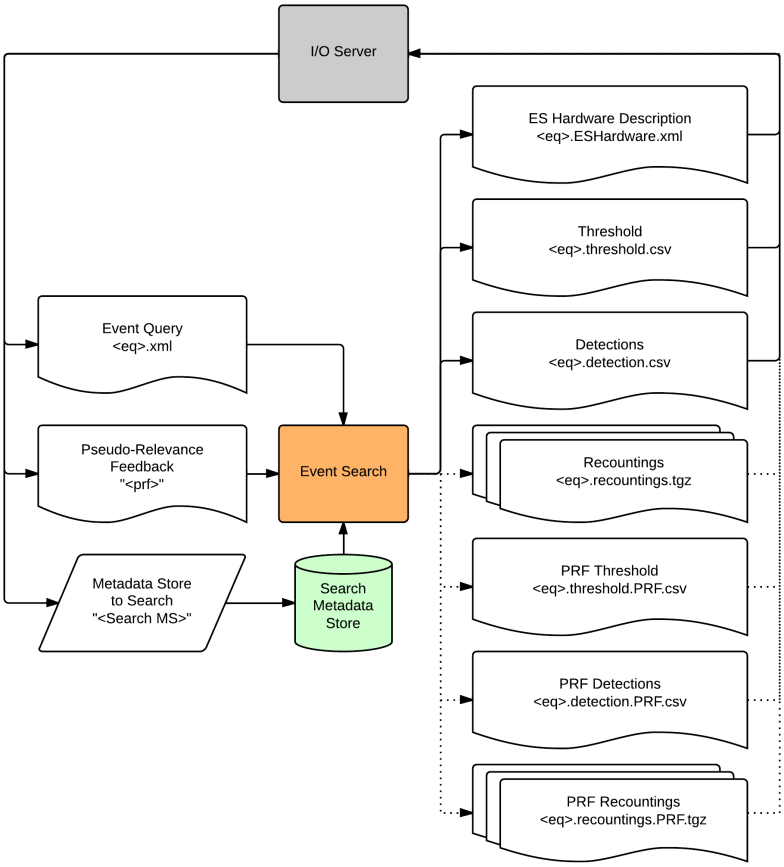

Figure 1.4.1: Event Search Block Diagram

ES (<eq>.xml, "<prf>", "<SearchMS>") -> <eq>.ESHardware.xml
<eq>.threshold.csv, <eq>.detection.csv, <eq>.recountings.tgz
<eq>_PRF.threshold.csv, <eq>_PRF.detection.csv, <eq>_PRF.recountings.tgz

The Event Search (ES) module performs the search over the metadata store (<Search MS>, e.g., "MED14-EvalFull") specified by the event query (<eq>.xml) as shown in Figure 1.4.1. The Event Search module will be run on all Event Queries that a team's system has previously produced by its SQG or EQG, in an order specified by the I/O Server. The system must produce a search results for all videos represented in the metadata store (and reported to the I/O server via the Video List in the metadata store description output "<SearchMS>".info.xml).

Search can produce up to two output sets: 1) the results from a run of the query, and 2) the results from a run of the query that includes the pseudo-relevance feedback. The PRF input tells search whether or not it should create the pseudo-relevance feedback output set. While generating the non-PRF output set, each video must be analyzed independently for the event; whereas the output set generated using PRF includes a dependent search, i.e., that one video is similar to another which the system is confident is an event positive. However the output set generated with PRF must have no user interaction.

For each output set, Event Search creates three outputs:

1) Threshold (<eq>.threshold.csv): the rank and event score thresholds
2) Detection Scores (<eq>.detection.csv): the rank and score for each trial (event query / search video pair)
3) Recounting (<eq>.recountings.tgz) (Optional): the annotation of the Event Query with scores and with key metadata tag evidence that was used to compute the score for the event. Recounting outputs are only required for teams participating in the MER evaluation.

Teams must strictly follow the inputs and outputs given in Figure 1.4.1. No other inputs are permitted. **The team should only use information in the Event Query to run the evaluation. No manual input is used in Event Search. Teams must use the metadata store and not the original videos.**

The Hardware Description <eventID>_<ex>.EShardware.xml describes the hardware that was actually used in processing this run of Event Search. ES must be run on a single COTS workstation or on a single core of a cluster. The format for this file is given in Appendix C.6.

The threshold file (<eq>.threshold.csv, exact format specified in Appendix C.4) has five columns:

- **EventID**: The Event ID processed – copied from the event query.
- **QueryType**: The query type (e.g. 010Ex, SQ)
- **PRF**: "PRF" or "noPRF"
- **DetectionThresholdScore**: A confidence score value of the system-supplied threshold.
- **DetectionThresholdRank**: A rank value of the system-supplied threshold.

The detection file (<eq>.detection.csv, exact format specified in Appendix C.5) contains one line for each trial. There are six columns in the file as follows:

1. **EventID**: The Event ID processed – copied from the event query.
2. **QueryType**: The query type (e.g. 010Ex, SQ)
3. **PRF**: "PRF" or "noPRF"
4. **VideoID**: The Video ID input and output from Metadata Generator as described in Section 1.1.
5. **Score**: A probability value between 0 (low) and 1 (high) representing the system's confidence that the event is present in the video.
6. **Rank**: a number from 1 to N (where N is the size of the video set) that ranks the video on how well it matches the event kit. 1 is the best match, N is the worst.

The recountings add information to the event query and must include (to be used for the MER Evaluation):

- **Name**: name of the entity being tagged
- **Score**: a confidence score
- **Snippet**: the location of the entity being tagged in space and time
  - **Start**: the start time of the snippet
    - **Start Upper Left**: the upper left corner of the bounding box surrounding the entity at the snippet start time
    - **Start Lower Right**: the lower right corner of the bounding box surrounding the entity at the snippet start time
  - **End**: the end time of the snippet
    - **End Upper Left**: the upper left corner of the bounding box surrounding the entity at the snippet start time

- **End Lower Right**: the lower right corner of the bounding box surrounding the entity at the snippet end time

An example recounting for the board-trick query is given in Table 1.4.2. The recounting points to key evidence (specified by the Event Query) that was used in computing the event score. If the metadata tags are as was shown in Table 1.1.1, then these pointers are just the metadata tag pointers of the matching key evidence. The red text shows how the event score was computed by the system. Details of the recounting format are provided in Appendix A.

0.95: Board Trick [E001] = **max** (0.95: Semantic Board Trick, 0.87: Detected Board Trick)

- 0.95: Detected Board Trick = **private**
  - 0.87: Semantic Board Trick = **max** (0.45: Board Trick Object**,**
    0.87: Board Trick Activity in a Board Trick Scene)
  - 0.45: Board Trick Object = **max** {0.0: *surfboard*, 0.0: *skateboard*, 0.45: *snowboard*}
  - 0.87: Board Trick Action in a Board Trick Scene = **min** {0.87: Board Trick Action,
    0.99: Board Trick Scene}
    - 0.87: Board Trick Action = **max** {0.87: *person jumping*, 0.78: *person flipping* }
    - 0.99: Board Trick Scene = **max** {0.0: *ocean scene*, 0.0: *city scene*, 0.99: *snow scene*}

Table 1.4.2: Recounting for Combined Board Trick Query (for illustrative purposes only): The recounting includes the Event Query show with 1) italicized text denoting metadata tags, 2) bold text signifying functions that combine scores and 3) plain text denoting variables in the semantic query. The recounting additions to the Event Query are shown in 1) red indicates the score computation 2) blue underlined text are pointers to the metadata tags (i.e., name, score, and video snippet pointers) that contain the evidence that was used to compute the event confidence score.

1.5 MED Performance Evaluation

For each MED evaluation (i.e., PS or AH) performance will be computed for each of the following event search runs:

<eq> = (SQ, 000Ex, 010Ex, 100Ex)

and with the additional PRF runs:

(SQ-PRF, 000Ex-PRF, 010Ex-PRF, 100Ex-PRF)

MED will be evaluated by how well Event Search retrieves and detects events in evaluation search video metadata and by the computing resources used to do so. The determination of correct detection will be at the video level: i.e. systems will provide a response for each video in the evaluation search video set. Participants must process each event independently in order to ensure each event will be scored independently.

Section 1.4 describes the output for a MED system for each event to be, a confidence score threshold value, a rank threshold and, for each video in the search set, an event confidence score between 0.0 and 1.0 and a rank from 1 to N in order of descending confidence score. Using ground truth, the system-supplied rank of each true-positive event video will converted into a rank vector, **rank**(tp) whose values is the system-supplied rank where tp = 1 to P_E and P_E is the total number of positives for the event.

Precision and Recall can be computed for each position in the rank vector. Recall, **Recall**(tp), is the index in the rank vector tp divided by the total number of positive videos P_E. Precision, **Prec**(tp), is the index in the rank vector, tp, divided by the system-supplied rank of that positive, **rank**(tp).

1.5.1 MED Performance Measures

<u>Retrieval Metric: Mean Average Precision, MAP</u>

For PS and AH event sets, the mean average precision (*MAP*) score will be computed as:

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AP\,(q)$$

where Q denotes the number of different events and *AP*(q) is the average precision for event *q*.

$$AP(q) = \frac{1}{P_E}\sum_{tp=1}^{P_E} Prec(tp) = \frac{1}{P_E}\sum_{tp=1}^{P_E} \frac{tp}{rank(tp)}$$

where P_E denotes the number of positives of event q.

<u>Detection Metric: Minimum Acceptable Recall, R_o</u>

For PS and AH event sets, the mean minimal acceptable recall, MRo will be computed as:

$$MR_0 = \frac{1}{Q}\sum_{q=1}^{Q} R_0\,(q)$$

where Q denotes the number of different events and Ro(q) is the minimal acceptable recall for event q and is computed as:

$$R_o(q) = Recall(T_q) - 12.5 * \frac{rank(T_q)}{V}$$

where Recall(T_q) and rank(T_q) is the recall and rank at the threshold T for event q, and V is the

total number of videos in the search set.

1.5.2 MED Timing Measures

Metadata Generation Processing Speed

NIST will report the time it takes each team to compute the various metadata stores, i.e., Event-BG, PS-Training, and AH-Training, and MED14Eval (i.e., MED14-EvalFull or MED14-EvalSub). NIST will also distribute the information on the computer system used for the processing. NIST will explore methods of normalizing processing time or performance results by the hardware used (returned by the functions hardware descriptions).

Event Query Generation Processing Speed & Search Processing Speed

NIST will report the SQG, EQG and ES run times for the Dry Run, PS and AH evaluations. NIST will also report the computing time used for the processing and will attempt to normalize these scores by the hardware used.

1.5.3 MED Diagnostic Measures

Diagnostic measures will be provided in software for teams to run on the test search videos. Diagnostic measures will not be provided as part of the evaluation to keep the content of the evaluation search videos blind. These diagnostic measures include:

Recall-Percent Rank Curves

Graphical performance assessment uses the recall as a function of the system's percent rank. Percent rank is computed by dividing rank by the total number of videos in the search set, V.

Precision-Recall Curves

Graphical performance assessment uses the precision as a function of the recall.

2 MER Evaluation Process and Performance Measures

Teams participating in MER must return the recounting from their Event Search modules for the 10 Exemplar Event Queries (<eventID>_010Ex.xml) for all events.

NIST will select a few events from E021-E050 to be used in the MER evaluation. For each of the selected events, NIST will also select a few Search Videos that were commonly detected as positives by the participating teams' systems. Human judges will view the same <videoID>-<eventID> recountings for the MER evaluation using the MER Workstation (similar to that used in MER 2013).

For the MER evaluation, NIST will report the following measures:

- **Event Query Quality**: The average Event Query Rating provided by the judges over all MER events. Judges will be asked to:
  - View the Event Query and the Event Descriptions (<eventID>.txt provided by NIST).
  - Rate the Event Query Quality (on a scale of 0-4): i.e., how well the Event Query captures *relevant* and *key* evidence for the event, i.e., does this seem like a concise and logical event query that would be created for the event.
- **Tag Quality**: The average Tag Quality Rating from the judges over all MER events. Judges will be asked to:
  - View the recounted key evidence, one piece at a time, and for each piece of key evidence provided rate the Tag Quality (on a scale of 0-4): i.e., how well the system's name captures the content of the recounted snippet.
- **Evidence Localization:** The average Evidence Localization Rating from the judges over all MER events. Judges will be asked to:
  - View the recounted key evidence, one at a time, and for each piece of key evidence provided rate the Evidence Localization (on a scale of 0-4): i.e., how well the system localized the evidence in time and space.
- **Key Evidence Quality**: The average Key Evidence Quality Rating provided by the judges over all MER events. Judges will be asked to:
  - View the all of the recounted evidence for the event and judge the Key Evidence Quality, (on a scale of 0-4), i.e., if the key evidence was convincing that the video contained the event.
- **Recounted Percent**: NIST will compute the time of the recounted snippets and divide by the total length of the video. Note that if no snippet start and end time are specified, NIST will assume that the snippet is the entire length of the video.

3 Semantic Query Editing Evaluation Pilot

In this evaluation judges will attempt to create a new Semantic Query from an existing Semantic Query. Since judges will not have access to the full Semantic Query Language of the participant's systems, the new events <E1xx> will be defined to be subsets of the original event <E0xx> so that judges need only to remove only metadata tags from the existing Semantic Query (<E0xx>_SQ.xml) to form the new Semantic Query (<E1xx>_SQ.xml).

For example if a judge were to edit the Board Trick Semantic Query to create a new Board Trick event that excludes surfing, then the judge would eliminate metadata tags in the Semantic Query as shown in Table 3.1.1. The text crossed out on the query represents the metadata tags that will be removed from the original Board Trick Semantic Query (E001_SQ.xml) to form a new Board Trick Semantic Query (E101_SQ.xml).

Semantic Board Trick [E001] = **max** (Board Trick Object**,** Board Trick Activity in a Board Trick Scene)
- Board Trick Object = **max** {~~*surfboard*~~, *skateboard*, *snowboard*}
- Board Trick Action in a Board Trick Scene = **min** {Board Trick Action, Board Trick Scene}
  - Board Trick Action = **max** {*person jumping*, *person flipping* }
  - Board Trick Scene = **max** {~~*ocean scene*~~, *city scene*, *snow scene*}

Table 3.1.1: Example Editing of a Semantic Query: The italicized text denotes metadata tags. Bold text signifies functions that combine scores. Plain text denotes variables in the semantic query. The crossed out text shows the metadata tags will be removed from the query.

Teams participating in the pilot evaluation will then run their Event Search module on the new Semantic Query (E1xx_SQ.xml). The teams will be scored (i.e., the MED performance measures discussed in Section 1.5) on the Event Search results on the new Semantic Query.

In addition, judges will be asked to provide a Query Editability rating (on a scale of 0-4): i.e., how easy it was to edit the team's Semantic Query.

To support this pilot MER evaluation, NIST will identify several (at least four) sub-events of the events used in the PS or AH evaluations. NIST/LDC will create a definition of the new event that will be used by the judges in editing the event. In addition, they will create ground-truth for the new event for the MED14-Eval.

NIST will report the Mean Average Precision (MAP) and the Mean Minimum Acceptable Recall (MRo) on the Event Search returns of the edited Semantic Queries. NIST will also report the average Query Editibility score across all Pilot events and judges.

4 Evaluation Schedule

MED/MER consists of several evaluations: Dry-Run, Pre-Specified (PS), Ad-Hoc (AH), MER, and Semantic Query Editing Evaluation Pilot. To compress the time line, some processing of these evaluations overlaps resulting in processing phases. The schedule is shown in Table 4.1.

MED has two evaluations: the Pre-Specified (PS) and the Ad-Hoc (AH); both are required. For the PS MED, teams must process <eventID> = (E021, …, E040) and for the AH MED, teams must process <eventID> = (E041, …, E050). For each evaluation, the team must perform search for each event using the <eq> = 010Ex option using either MED14-EvalFull or MED14-EvalSub search set.

For each of the two evaluations, teams will have the following options:

- Produce and search for the SQ, 000Ex and/or 100Ex event queries
  - If a team is participating in the Pilot evaluation, they must have the SQ option
- Produce recounting in search
  - If a team is participating in MER, they must produce recountings
- Produce an additional PRF output in Event Search

The acronyms MG (Metadata Generator), SQG (Semantic Query Generator), EQG (Event Query Generator), and ES (Event Search) are used in the following subsections to identify modules.

Start	End	Section	T&E Team Deliveries (in blue) and Phases
	Thu, May 01		I/O Server available to teams for testing
Thu, May 01	Fri, Jun 06	4.1	MG on MED14-Test for the Dry Run
Mon, Jun 09	Fri, Jun 20	4.2	Metadata Generation on Event-BG (for all evaluations)
Mon, Jun 23	Mon, Jun 30	4.3	Dry Run
Mon, Jul 07	Fri, Jul 11	4.4	Semantic Query Editing Pilot Dry Run
	Wed, Jul 09		Evaluation Search Video delivered to teams
Wed, Jul 16	Mon, Aug 18	4.5	MG MED14-Eval for PS and AH evaluations
Mon, Aug 11	Sun, Aug 17	4.6	PS Evaluation – Part A
Mon, Aug 18	Fri, Aug 22	4.7	PS Evaluation – Part B
Mon, Aug 25	Thu, Aug 28	4.8	AH Evaluation
Mon, Sep 01	Thu, Sep 11	4.9	Semantic Query Editing Evaluation Pilot

Table 4.1: Evaluation Schedule

The MG can be implemented on any computing system. However, the SQG, EQG, and ES must be implemented on a personal COTS workstation OR on a single core of a cluster processor.

Included with the output of each module call teams are required to send NIST a hardware description file specifying the hardware configuration(s) for the computation of that particular module call. Teams are also required to provide a system description prior to the completion of the evaluation. Sending the system description will be handled through the I/O Server client-side utility. System and Hardware description file specifications are defined in Appendix D and C.6 respectively.

The following sections describe which module calls will be required for each phase.

4.1 Metadata Generation on MED14-Test

Teams are strongly encouraged to participate in the Dry Run Evaluation. The MED14-Test Set serves as the Search Set for the Dry Run Evaluation and for independent testing for the teams. This phase includes one run of the Metadata Generator (MG) on the MED14-Test set, i.e.

MG (“”, MED14-Test.csv, “MED14-Test”) -> MED14-Test.MGHardware.xml, MED14-Test.info.xml

4.2 Metadata Generation on Background Event Training Videos

The same set of background event training videos (Event-BG) is used for all evaluations. This phase includes one run of the Metadata Generator (MG) on the Event-BG set, i.e.

MG (“”, Event-BG.csv, “Event-BG”) -> Event-BG.MGHardware.xml, Event-BG.info.xml

Note: Teams participating in the Dry Run have the option to recompute their Event-BG metadata up to the beginning of the PS Evaluation Part A.

4.3 Dry Run

Teams are strongly encouraged to participate in the Dry Run Evaluation which includes the following module runs (gray text shows optional inputs and outputs):

1. **MG** (“Event-BG”, PS-Training.csv, “PS-Training”) -> PS-Training.MGHardware.xml, PS-Training.info.xml
2. **SQG** (“<eventID>”, <eventID>.txt) -> <eventID>_SQ.SQGHardware.xml, <eventID>_SQ.xml (pre-computed SQ can be submitted)
3. **EQG** (<eventID>_SQ.xml, <eventID>_<ex>.csv, “PS-Training”) -> <eventID>_<ex>.EQGHardware.xml, <eventID>_<ex>.xml
4. **ES** (<eventID>_<eq>.xml, “<prf>”, <trial>.csv, “MED14Test”) -> <eventID>_<eq>.ESHardware.xml, <eventID>_<eq>.threshold.csv, <eventID>_<eq>.detections.csv, <eventID>_<eq>.recountings.tgz, <eventID>_<eq>_PRF.threshold.csv, <eventID>_<eq>_PRF.detections.csv, <eventID>_<eq>_PRF.recountings.tgz

where,

<eventID> = E031 – E035
<eq> = (SQ, <ex> = (000Ex, 010Ex, 100Ex))
<prf> = “noPRF” or “PRF”

4.4 Semantic Query Editing Pilot Dry Run

For teams participating in the Pilot Dry Run must also participate in the Dry Run. NIST will modify one of their Semantic Queries from the Dry Run to create a new query <dr-pilot>_SQ.xml. Teams will then run this new semantic query on the Dry Run test set by:

ES(<dr-pilot>_SQ.xml, “<prf>”, “MED14-Test”) -> <dr-pilot>_SQ.ESHardware.xml, <dr-pilot>_SQ.threshold.csv, <dr-pilot>_SQ.detections.csv, <dr-pilot>_SQ.recountings.tgz, <dr-pilot>_SQ_PRF.threshold.csv, <dr-pilot>_SQ_PRF.detections.csv, <dr-pilot>_SQ_PRF.recountings.tgz

where <prf> = “noPRF” or “PRF”

4.5 Metadata Generation on the MED14-Eval

This phase is required for the PS and AH evaluations. The participating team has the choice to run the Metadata Generator on either the MED14EvalFull or MED14EvalSub. The MED14-Eval is special since parts of this set are intended to be used for future evaluations. Therefore the following rules MUST be observed by all teams:

- Participants MUST delete all PROGTest Collection videos and any metadata (or any other data derived from this set) before the beginning of MED14.
- To maintain the integrity of the evaluation, these video lists should NOT be edited and the underlying videos must not be annotated or altered in any manner.
- Participants must not attempt to gain knowledge of the video sets’ properties or content by visually inspecting the video, video metadata, module outputs, or statistics developed during the processing.
- These video sets are only to be used as part of an official evaluation submission to MED14 and MER14 evaluations and must be deleted from teams’ systems after each year’s MED evaluation ends.

This phase runs:

MG (“”, <eval>.csv, “<eval>”) -> <eval>.MGHardware.xml, <eval>.info.xml

where,

<eval> = MED14-EvalFull or MED14-EvalSub

4.6 PS – Part A

These are the beginning runs for the pre-specified evaluation. Note that step 1 is optional, teams may use the “PS-Training” metadata store generated in the Dry Run phase (Section 4.3), or re-generate it. (optional steps are shown in gray):

1. **MG**(“Event-BG”, PS-Training.csv, “PS-Training”) -> PS-Training.MGHardware.xml, PS-Training.info.xml
2. **SQG**(“<eventID>”, <eventID>.txt) -> <eventID>_SQ.SQGHardware.xml, <eventID>_SQ.xml (pre-computed SQ can be submitted)
3. **EQG**(<eventID>_SQ.xml, <eventID>_<ex>.csv, “PS-Training”) -> <eventID>_<ex>.EQGHardware.xml, <eventID>_<ex>.xml,

where

<eventID> = E021 – E040

<ex> = 000Ex, 010Ex, 100Ex

4.7 PS – Part B

This phase includes is the Event Search modules runs for the PS Evaluation (optional steps are shown in gray):

ES (<eventID>_<eq>.xml, “<prf>”, “<eval>”) -> <eventID>_<eq>.ESHardware.xml, <eventID>_<eq>.threshold.csv, <eventID>_<eq>.detections.csv, <eventID>_<eq>.recountings.tgz,

<eventID>_<eq>_PRF.threshold.csv, <eventID>_<eq>_PRF.detections.csv, <eventID>_<eq>_PRF.recountings.tgz

where

<eval> = MED14-EvalFull or MED14-EvalSub
<eventID> = E021 – E040
<eq> = (SQ, <ex> = (000Ex, 010Ex, 100Ex))
<prf> = "noPRF" or "PRF"

4.8 Ad-Hoc

The Ad-Hoc MED contains the following runs (optional steps are shown in gray):

1. **MG** ("Event-BG", AH-Training.csv, "AH-Training") -> AH-Training.MGHardware.xml, AH-Training.info.xml
2. **SQG** ("<eventID>", <eventID>.txt) -> <eventID>_SQ.SQGHardware.xml, <eventID>_SQ.xml
3. **EQG**(<eventID>_SQ.xml, <eventID>_<ex>.csv, "AH-Training") -> <eventID>_<ex>.EQGHardware.xml, <eventID>_<ex>.xml,
4. **ES**(<eventID>_<eq>.xml, "<prf>", "<eval>") -> <eventID>_<eq>.ESHardware.xml, <eventID>_<eq>.threshold.csv, <eventID>_<eq>.detections.csv, <eventID>_<eq>.recountings.tgz, <eventID>_<eq>_PRF.threshold.csv, <eventID>_<eq>_PRF.detections.csv, <eventID>_<eq>_PRF.recountings.tgz

where

<eval> = MED14-EvalFull or MED14-EvalSub
<eventID> = E041 – E050
<eq> = (SQ, <ex> = (000Ex, 010Ex, 100Ex))
<prf> = "noPRF" or "PRF"

4.9 Semantic Query Editing Evaluation Pilot

For teams participating in the Pilot Semantic Query Editing Evaluation, NIST will modify several of their Semantic Queries from the PS-MED or AH-MED, i.e., <pilot_event>. Teams will then run their Event Search modules on the pilot events:

ES(<pilot_event>_SQ.xml, "<prf>", "<eval>") -> <pilot_event>.ESHardware.xml, <pilot_event>_SQ.threshold.csv, <pilot_event>_SQ.detections.csv, <pilot_event>_SQ.recountings.tgz, <pilot_event>_SQ_PRF.threshold.csv, <pilot_event>_SQ_PRF.detections.csv, <pilot_event>_SQ_PRF.recountings.tgz

where

<eval> = MED14-EvalFull or MED14-EvalSub
<prf> = "noPRF" or "PRF"

5 Evaluation Tools and Command Line Example

NIST will adapt the Detection EVAluation (DEVA) tools within the NIST Framework for Detection Evaluation (F4DE) toolkit to make use of the Scoring I/O server. Details of how to use the tools will be forthcoming.

6 References: A list of pertinent references

None

Appendix A: Query Structure

The following serves as an early draft of the query and recounting structure. Final structure TBD.

Queries are xml files, whose structure conforms to that of the following example:

```
<query eventID="E001">
        <node   id='E001', name='Board Trick', eq='MAX'>
                <detector id='D' name='Detected Board Trick'> <![CDATA[parameters]]> </detector>
                <node   id='S', name='Semantic Board Trick', eq='MAX'>
                        <node   id='S1', name='Board Trick Object', eq='MAX'>
                                <node   id='S1.1' name='surfboard'/>
                                <node   id='S1.2' name='skateboard'/>
                                <node   id='S1.3' name='snowboard'/>
                        </node>
                        <node   id='S2' name='Board Trick Activity in a Board Trick Scene' eq='MIN'>
                                <node   id='S2.1' name='Board Trick Activity' eq='MAX'>
                                        <node   id='S2.1.1' name='person jumping'/>
                                        <node   id='S2.1.2' name='person flipping'/>
                                </node>
                                <node   id='S2.2' name='Board Trick Scene' eq='MAX'>
                                        <node   id='S2.2.1' name='ocean scene'>
                                        <node   id='S2.2.2' name='city scene'>
                                        <node   id='S2.2.3' name='snow scene'>
                                </node>
                        </node>
                </node>
        </node>
</query>
```

An xml schema will be provided by NIST for reference and for use during validation. Semantic nodes can be either combination nodes or terminals, and there may multiple detector nodes.

Recountings are xml files that "fill out" the query from which they were generated. Recountings contain evidence nodes as shown in the following example:

```
< recounting eventID="E001" videoID="123456">
        <node id='E001' name='Board Trick' eq='MAX' score = 0.95>
                <detector, id='D' name='Detected Board Trick' score='0.95'> <![CDATA[parameters]]>
                </detector>
                <node   id='S' name='Semantic Board Trick' eq='MAX' score='0.87'>
                        <node   id='S1' name='Board Trick Object' eq='MAX' score='0.45'>
```

```
                    <node   id='S1.1' name='surfboard' score = 0.0/>
                    <node   id='S1.2' name='skateboard' score = 0.0/>
                    <node   id='S1.3' name='snowboard' score = 0.45>
                            <evidence start ='5' end ='10' score='0.45'
                            start_upper_left='10:20' start_lower_right='180:240'
                            end_upper_left='100:200' end_lower_right='280:340'/>
                    </node>
            </node>
            <node   id='S2' name='Board Trick Activity in a Board Trick Scene' eq='MIN'
                                                                    score='0.87'>
                    <node   id='S2.1' name='Board Trick Activity' eq='MAX' score='0.87'>
                            <node   id='S2.1.1' name='person jumping' score='0.87'>
                                    <evidence start ='35' end ='40' score='0.87'
                                    start_upper_left='15:25' start_lower_right='85:145'
                                    end_upper_left='25:35' end_lower_right='125:340'>
                            </node>
                            <node   id='S2.1.2' name='person flipping' score='0.45'>
                                    <evidence start ='105' end ='110' score='0.45'
                                    start_upper_left='110:120' start_lower_right='280:340'
                                    end_upper_left='200:300' end_lower_right='380:440'>
                            </node>
                    </node>
                    <node   id='S2.2' name='Board Trick Scene' eq='MAX' score = 0.99>
                            <node   id='S2.2.1' name='ocean scene' score='0.0'>
                            <node   id='S2.2.2' name='city scene' score='0.0'>
                            <node   id='S2.2.3' name='snow scene' score='0.99'>
                    </node>
            </node>
    </node>
</node>
</recounting>
```

Empty Semantic Queries are specified in XML by query tag (<query/>) in an xml file

Elements/attributes shown in red indicate content which is new/modified when generating a recounting based on the example query.

Appendix B: Submission Instructions

B.1 Recounting Archive Specifications

Teams participating in MER are required to include a recounting archive upon completion of Event Search tasks. Recounting archives should only include recountings using the following naming convention:

<clipID>.<eventID>.mer.xml *e.g. 012345.E001.mer.xml*

B.2 Validating Outputs

NIST will provide a set of tools to validate output that is to be sent to the MED I/O Server. The client-side tool used to interface with the I/O Server will automatically validate files before sending them.

Exact command line usage TBD.

Appendix C: File Formats

The following subsections specify the file formats of files to be received from, or sent to the I/O Server.

C.1 Metadata Store Description

The Metadata Store Description is an output from the Metadata Generator module. It is an xml file of the following format.

```
<MS_Description>
        <Signal_Metadata size="size in gigabytes (GB)">
        </Signal_Metadata>
        <ASR_OCR_Metadata size="size in megabytes (MB)">
                <OCR_Languages>
                        <Language>lang1</Language>
                        <Language>lang2</Language>
                </OCR_Languages>
                <ASR_Languages>
                        <Language>lang3</Language>
                        <Language>lang4</Language>
                </ASR_Languages>
        </ASR_OCR_Metadata>
        <Semantic_Metadata size="size in megabytes (MB)">
                <Semantic_Tags>                 # these are names of
                        <Tag>tag name 1</Tag>   # things tagged by the MG
                        <Tag>tag name 2</Tag>   # (not specific tags on the videos)
                </Semantic_Tags>
        </Semantic_Metadata>
        <Video_List>
```

```
            <VideoID>video1</VideoID>
            <VideoID>video2</VideoID>
        </Video_List>
</MS_Description>
```

C.2 Video List

A video list file is an input provided to the Metadata Generator module. It is a csv file of the following format.

VideoID, Filename
HVC012345, HVC012345.mp4
HVC123456, HVC123456.mp4
HVC234567, HVC234567.mp4

Header line should be included.

C.3 Exemplar List

An exemplar list is an input to the Event Query Generator module. It is a csv file of the following format.

EventID, ExemplarSet, VideoID, Label
E001, 010Ex, HVC012345, positive
E001, 010Ex, HVC123456, near-miss
E001, 010Ex, HVC234567, background

ExemplarSet can be 000Ex, 010Ex, 100Ex
Header line should be included

C.4 Threshold

A threshold file is an output of the Event Search module. It is a csv file of the following format.

EventID, QueryType,PRF, DetectionThresholdScore, DetectionThresholdRank
E001, SQ, noPRF,0.90,1

QueryType can be SQ, 000Ex, 010Ex, 100Ex
PRF can be noPRF, PRF
Header line should be included

C.5 Detection

A detection file is an output of the Event Search module. It is a csv file of the following format.

EventID, QueryType, PRF, VideoID, Score, Rank
E001, 010Ex, noPRF, HVC012345, 0.97, 1
E001, 010Ex, noPRF, HVC123456, 0.66, 2

E001, 010Ex, noPRF, HVC234567, 0.54, 3

QueryType can be SQ, 000Ex, 010Ex, 100Ex
PRF can be noPRF, PRF
Header line should be included

C.6 Hardware Descriptions

Teams must submit a hardware description XML file along with the output from each module call. The intent of the hardware description file is for teams to specify the hardware configurations used for the computation of a particular module call. Each hardware description should follow the example structure shown in the subsections below.

C.6.1 Metadata Generator Hardware Description

For the Metadata Generator hardware description multiple CPU/GPU specs can be defined. Multiple clusters may also be specified.

```
<MG_Hardware>
        <Cluster>
                <Comments>Include any comments here</Comments>
                <OS>(Type, version, 32 vs 64 bit)</OS>
                <CPU    model="model"
                        speed="speed in GHz"
                        num_cores="number of cores"
                        ram_per_cpu="RAM per CPU in GB"
                        count="number of CPUs of this spec"/>
                <CPU .... />
                <GPU    model="model"
                        speed="speed in MHz"
                        num_cores="number of cores"
                        memory="memory in GB"
                        count="number of GPUs of this spec"/>
                <GPU … />
                <Network_Bandwidth>Network bandwidth in Gbit/s</Network_Bandwidth>
                <Temp_Storage>Temporary Disk Usage in GB</Temp_Storage>
                <Output_Bandwidth>Network bandwidth in Gbit/s</Output_Bandwidth>
                <Output_Storage>Output Disk Usage in GB</Output_Storage>
        </Cluster>
        <Cluster>
                …
        </Cluster>
</MG_Hardware>
```

C.6.2 Other Modules Hardware Description

For the hardware descriptions of other modules, only a single workstation or cluster core may be defined.

```
<COTS_Hardware>
        <Workstation>
                <Comments>Include any comments here. </Comments>
                <OS>(Type, version, 32 vs 64 bit)</OS>
                <CPU    model="model"
                        speed="speed in GHz"
                        count="number of CPUs of this spec"
                        ram="RAM per CPU in GB"/>
                <GPU    model="model"
                        speed="speed in MHz"
                        memory="memory in MB"
                        count="number of GPUs of this spec"/>
                <Temp_Storage>Temporary Disk Usage in GB</Temp_Storage>
                <Output_Storage>Output Disk Usage in GB</Output_Storage>
        </Workstation>
        -- OR --
        <Single_Cluster_Core>
                <Comments>Include any comments here. </Comments>
                <OS>(Type, version, 32 vs 64 bit)</OS>
                <CPU    model="model"
                        speed="speed in GHz"
                        ram="RAM per CPU in GB"/>
                <Network_Bandwidth>Network bandwidth in Gbit/s</Network_Bandwidth>
                <Temp_Storage>Temporary Disk Usage in GB</Temp_Storage>
                <Output_Bandwidth>Network bandwidth in Gbit/s</Output_Bandwidth>
                <Output_Storage>Output Disk Usage in GB</Output_Storage>
        </Single_Cluster_Core>
</COTS_Hardware>
```

Appendix D: System Description

Documenting each system is vital to interpreting evaluation results. As such, each team must submit a system description as a .txt file with the information listed below.

Section 1 ***System Description:*** A brief technical description of your system, including a description of the metadata language

Section 2 ***Metadata Generator Description:*** A brief technical description of your Metadata Generation module

Section 3 ***Semantic Query Generator Description:*** A brief technical description of your Semantic Query Generation module

Section 4 ***Event Query Generator Description:*** A brief technical description of your Event Query Generation module

Section 5 ***Event Search Description:*** A brief technical description of your Event Search module

Section 6 ***Training data and knowledge sources:*** Lists the resources used for system development beyond the provided MED corpora. Note teams are allowed only to use data outside the MED corpora for the development of their modules. No external data is to be used during the execution of the modules.