Florida International University – University of Miami: TRECVID 2019

Ad-hoc Video Search (AVS) Task

Yudong Tao[1], Tianyi Wang[2], Diana Machado[2], Raul Garcia[2], Yuexuan Tu[1], Maria Presa Reyes[2], Yeda Chen[1], Haiman Tian[2], Mei-Ling Shyu[1], Shu-Ching Chen[2]

[1]University of Mimai, Coral Gables, FL, USA
[2]Florida International University, Miami, FL, USA

Agenda

UNIVERSITY OF MIAMI COLLEGE of ENGINEERING

FIU Computing & Information Sciences

1. **Submission Details**
2. **Introduction**
3. **Proposed Framework**
   - Concept Bank
   - Incorporating Object Detection
   - Just-In-Time Concept Learning
   - Query Parsing
4. **Experimental Results**
   - Evaluation
   - Performance
5. **Conclusion**

Submission Details

- **Class**: F (Fully automatic runs)
- **Training Type**: E (Used only training data collected automatically, using only the official query textual description)
- **Team ID**: FIU-UM (Florida International University – University of Miami)
- **Year**: 2019

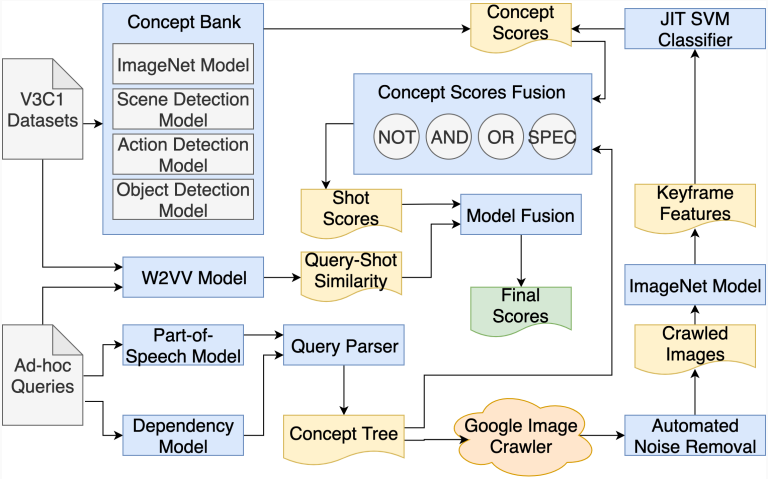
Introduction TRECVID 2019 AVS Task

UNIVERSITY OF MIAMI COLLEGE of ENGINEERING

FIU Computing & Information Sciences

- **Test Collection**: V3C1 dataset with 7475 Internet Archive videos (1.3 TB, around 1000 total hours and 1.08 million shots)
- **Mean Video Duration**: 8 minutes and 2 seconds
- **Queries**: 30 new queries (some new challenges)
  - Complex Scene: 639 “Find shots for inside views of a small airplane flying”
  - Ambiguous Objects: 627 “Find shots of a person holding a tool and cutting something”
  - Objects with various appearance: 617 “Find shots of one or more picnic tables outdoors” and 625 “Find shots of a person wearing a backpack”
- **Results**: A maximum of 1000 possible shots from the test collection for each query

Proposed Framework



The designed framework for the TRECVID 2019 AVS task

Concept Bank Summary

The concept bank contains all the datasets and the corresponding deep learning models we used in our system

Model Name	Database	# of concepts	Concept type(s)
InceptionResNetV2	ImageNet	1000	Object
ResNet50	Places	365	Scene
VGG16	Hybrid (Places, ImageNet)	1365	Object, Scene
Mask R-CNN	COCO	80	Object
ResNet50	Moments in Time	339	Action
TRN	Something-Something-v2	174	Action
Kinetics-I3D	Kinetics	400	Action

Concept Bank Usage

- Many concepts are not available in concept bank
- Used concepts:
  - ImageNet: “coral reef” “truck” and “backpack”
  - Coco: “backpack”, “umbrella”, “bicycle”, “car”, and “truck”
  - Moment: “cutting”, “dancing”, “driving”, “hugging”, “opening”, “flying”, “racing”, “riding”, “running”, “singing”, “smoking”, “standing”, and “walking”
  - Kinetics: “driving car”, “hugging”, “singing”, and “smoking”
  - Places, Something-Something: None (Several available for progress topics)
- Using concept name to match can be misleading:
  - expected “drone flying”, dataset “bird/airplane flying”
  - expected “opening door”, dataset “opening boxes”

Incorporating Object Detection

- Count the number of objects;
- Detect small objects;
- Object detection model significantly benefits query 625 “Find shots of a person wearing a backpack” due to the small object
- Object detection model helps explicitly determine object count (two progress topics 607 & 608)

Confidence Score of the Object Count

- $P_{O,N}(I)$: the confidence score object O appearing N times in the image I;
- n: the number of object O in the image I detected by the model;
- $P^i_O(I)$: the i-th highest confidence score among all the detected objects O in image I;

$$P_{O,N}(I) = \begin{cases} 0 & n < N \\ \prod_{i=1}^{N} P^i_O(I) & n = N \\ \prod_{i=1}^{N} P^i_O(I) \cdot \prod_{i=N+1}^{n} (1 - P^i_O(I)) & n > N \end{cases}$$

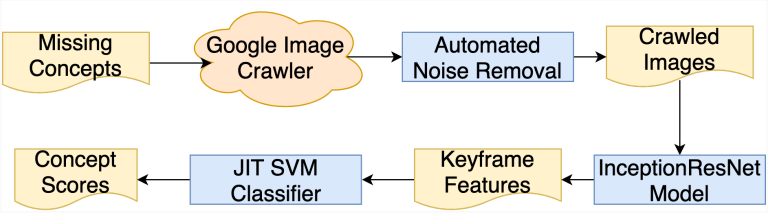
K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.

Just-In-Time Concept Learning

UNIVERSITY OF MIAMI COLLEGE of ENGINEERING

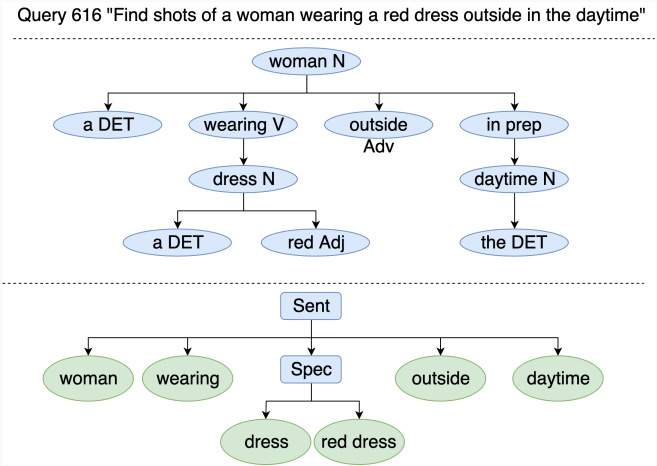
FIU Computing & Information Sciences

- Automatically crawls images in Google image search engine for the missing concepts;
- For each new concept, around 10,000 images are crawled;
- Filters the outliers in the search engine results with auto-encoder;
- InceptionResNet-v2 model is used to extract features;
- Trains the SVM classifier to detect the concepts.



Query Parsing Concept Tree

1. Process query using pre-trained Part-Of-Speech (POS) and Dependency (DET) parser
2. Convert the Dependency Tree into Concept Tree incorporating POS



Query Parsing Node Types

- **Concept:** the basic leaf nodes. It represents a specific semantic concept.
- **Numbered Concept:** an alternative leaf node. It represents that the concept is modified by a number.
- **Not Node:** a non-leaf node with only one child, which represents that the query includes a concept with complementary meaning of its child.
- **And Node:** a non-leaf node with two or more children, which represents that the query has its semantic meaning of all its children appearing concurrently.
- **Or Node:** a non-leaf node with two or more children. The query has its semantic meaning that any of its children exists in the video.
- **Spec Node:** a non-leaf node with exactly two children. One is the modifier and the other is the central concept.
- **Sent Node:** an unique non-leaf node which is essentially an “And Node” while it has at most five children, namely subject, action, object, place, and time, respectively.

Query Parsing Score Fusion - NOT/AND/OR

- **Not Node:** The score of this node is computed by $1 - s_{child}$, where s_{child} is the score of its child.
- **And Node:** The score of this node is computed by the geometric mean of all the children of the node.
- **Or Node:** The score of this node is determined as the maximum of the scores among all its children.
- S_i: The score of the *i*-th concept;
- w_i: The weights of the *i*-th concept, determined by the concept rarity;
- $\mathcal{N}$: Number of the concepts;

"NOT" Operation	"AND" Operation	"OR" Operation
$\text{Score}_{\text{not}} = 1 - S_{child}$	$\text{Score}_{\text{and}} = \prod_{i=1}^{\mathcal{N}} S_i^{w_i}$	$\text{Score}_{\text{or}} = \max_{i=1,\ldots,\mathcal{N}} S_i$

- **Spec Node:** The score of this node is computed in one of the two ways: the weighted arithmetic or geometric mean of the central concept and the modifier;
- $w_c \in [0, 1]$ is the weight of central concept;
- s_c is the score of its central concept;
- s_m is the score of its modifier.

"SPEC" Operation (arithmetic)

$$\text{Score}_{\text{spec}} = w_c \times S_c + (1 - w_c) \times s_m$$

"SPEC" Operation (geometric)

$$\text{Score}_{\text{spec}} = S_c^{w_c} \times S_m^{(1-w_c)}$$

Model Fusion

- **W2VV Model:** We leverage existing zero-shot video-text matching model, Word2VisualVector model trained on MSR-VTT and Flickr30k datasets, to generate similarity scores.
- **Fusion by threshold:** We compute the tf-idf measures of each concepts in training dataset of W2VV models and decide to rely on one of the model based on a empirical learned threshold;
- **Fusion by average:** Use the average of normalized scores from both models;
- **Score Normalization:** the normalized score is computed by the z-score normalization for each model,

$$\tilde{s} = \frac{s - \mu}{\sigma}$$

where s is the original model scores, μ and σ is the mean and standard deviation of model scores over all video shots in V3C1 dataset.

Evaluation

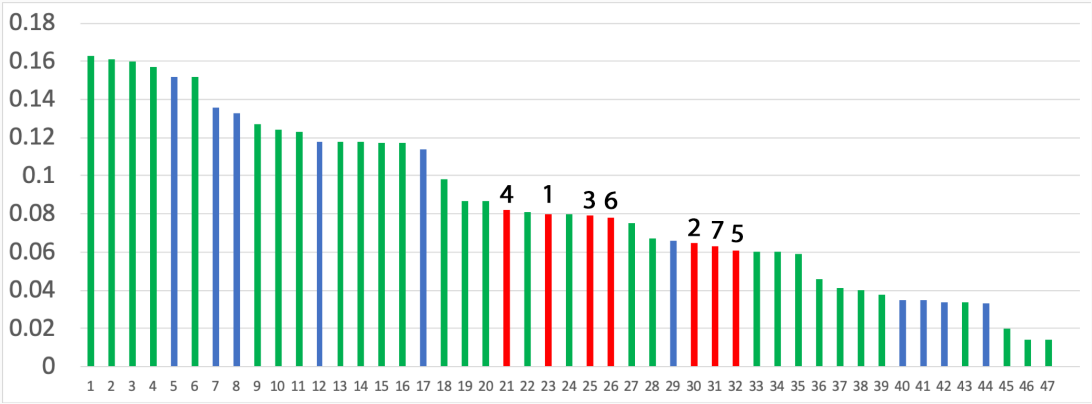
- **Metrics**: Mean extended inferred average precision (mean xinfAP);
- **Sampling**: All the top-250 results and 11% of the remaining results;
- As in the past years, the detailed measures are generated by the *sample_eval* software provided by NIST.

Submission Details

Table 1. Configuration of all the submitted runs

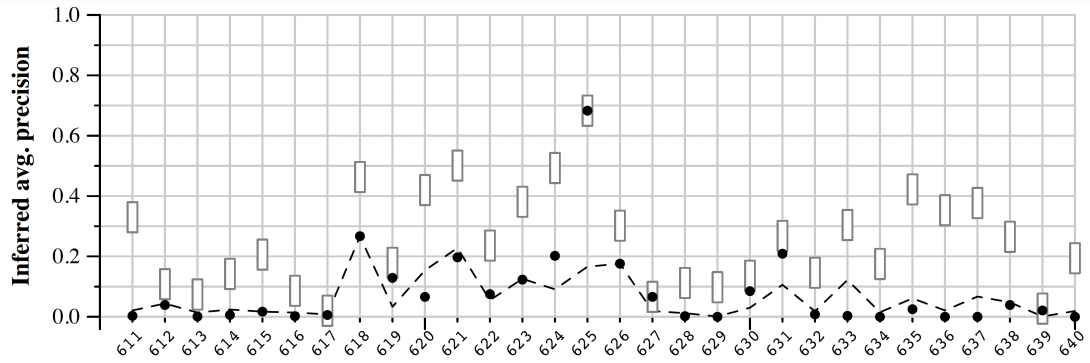
Run Name	Weighted	Concept Fusion	W2VV	Model Fusion
run1	no	arithmetic	yes	average
run2	yes	geometric	yes	threshold
run3	yes	geometric	yes	average
run4	yes	geometric	no	N/A
run5	no	geometric	yes	threshold
run6	no	geometric	no	N/A
novel run	use specific only	geometric	no	N/A

Performance overall xinfAP



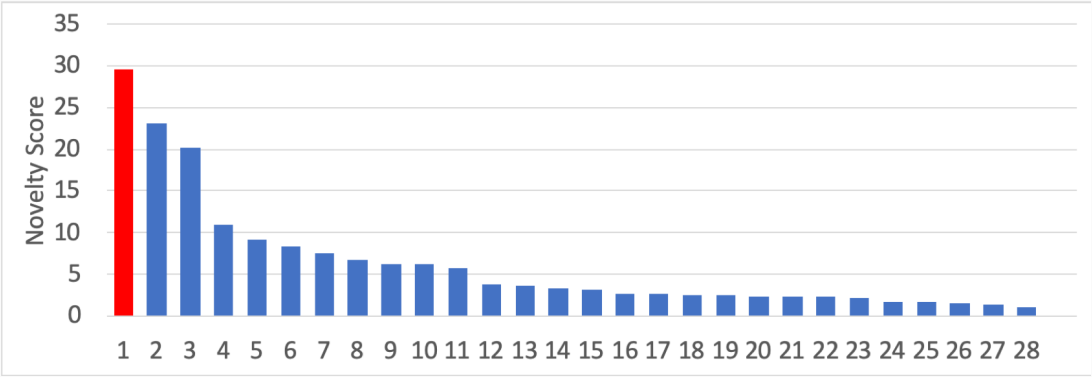
Comparison of FIU UM runs (red) with other runs for all the submitted fully automated (green), manually-assisted (blue), and relevance-feedback (orange) results.

Performance per-query xinfAP



Detailed scores of run4

Performance novelty scores



Novelty score of submitted novel run

Conclusion

- Develop methods to summarize training dataset in textual or embedding data
- Most of the pre-trained model suffer various resolution and object size.
- Better filter algorithm should be developed since when a very specific concept is submitted to search engine, many noisy images are included

Thanks!

Any questions?