How to search among live video streams?

User-broadcast live video means that standard whole-video or multi-modal retrieval approaches can not be used. Instead, videos must be described by their current visual content.

You can't know what people will search for...

How to find it if you've never seen it?

*e.g. "UFO Invasion Belize"

We adapt an approach from Zero-Shot Classification:

A semantic embedding relates visual concepts to query (Norouzi, ICLR'14). Videos are thusly:

$$\text{score}(q, x_t) = s(q)^T \phi(x_t)$$

Where $s$ gives the cosine similarity between the query and the concepts, and $\phi$ is some sparse representation.

Representation should represent current content...

Information from the first frame may no longer be useful.

We explore Memory Pooling and Memory Welling, which seek to boost temporal relevance.

Memory Pooling

Mean or max pooling over temporal memory window

Memory Welling

Wells fill with new detections and leak with time:

$$w(x_t) = \max\left(\frac{m - 1}{m}w(x_{t-1}) + \frac{1}{m}x_t - \beta, 0\right)$$

$\beta$ is the leakiness parameter, encouraging reliability and inducing sparsity within the representation.

Instantaneous Retrieval

"Give me a ranked list of relevant videos"

Much like traditional retrieval, you want to return videos that are relevant at the time of the search.

Evaluation

Mean Average Precision across both classes and time.

$$\frac{1}{\sum_t y_t} \sum_t \text{AP}_t y_t$$

Results

Shown for ActivityNet dataset, split across classes

- Memory methods outperform baselines.
- Memory Welling exceeds Pooling approach at lower cost

Continuous Retrieval

"Keep showing me relevant content."

A user wants to watch an unbroken series of relevant content. If the stream is no longer relevant, the algorithm should switch streams.

Evaluation

Count every time the algorithm stays on a single relevant stream or switches from an irrelevant to relevant stream.

$$\frac{x_t + r_x}{\sum_t y_t}$$

Results

Shown for 30-minute long concatenations of ActivityNet.

- Avg-to-Time fails due to nature of data
- Memory methods outperform baselines