Multi-attention Networks for Temporal Localization of Video-level Labels

Team Locust (#13)




<u>Marcos V. Conde</u>

Lijun Zhang, Srinath Nizampatnam, Ahana Gangopadhyay

Introduction

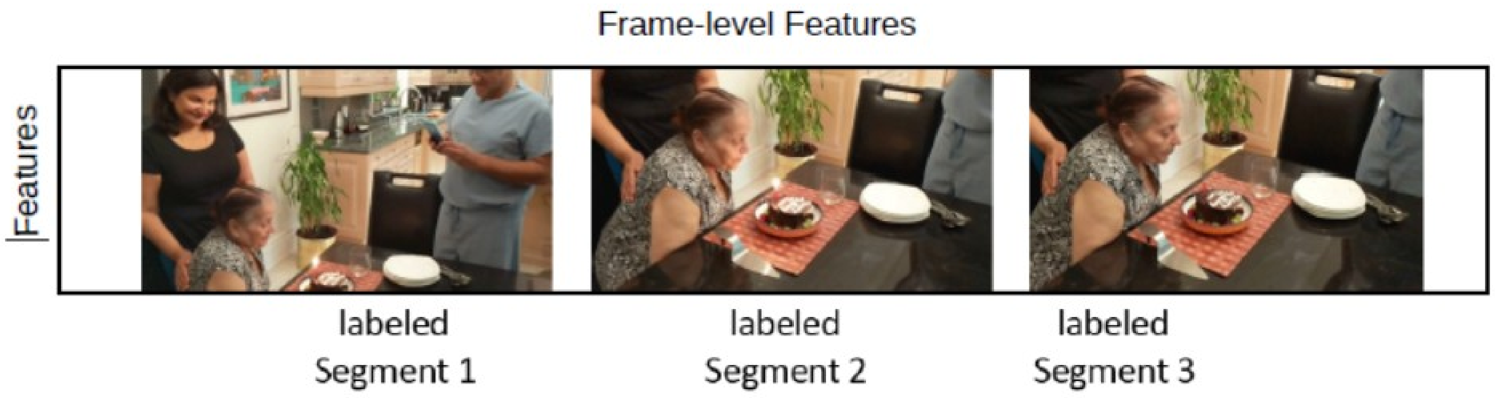
- Video-level classification vs Segment-level classification
- In the **Youtube-8M Segment Dataset**, multiple 5-second segments are sampled and then labeled by human raters
- **Temporally localizing** the presences of objects/actions can help us to *identify relevant moments in a video* and thus better understand its content.
- Large training dataset with only noisy video-level labels together and relatively smaller segment-level validation dataset.

First Approach. Previous methods

- Video-level classifier:
  - Logistic regression, Mixture of Experts(MoE)
- Frame-level classifier:
  - Neural network methods:
    - CNN, RNN
  - Pooling via clustering methods:
    - NetVlad, Deep Bag of Frames (DBoF)
- Context gating

The idea: Detect Important Frames

- The core idea is to use multiple attention weights to emphasize critical frames from different high-level topics in the video.
- We propose to use an **attention-based network** to selectively emphasize important frames within each video.



An example of the detected action "blowing out candles"

Problem Formulation. MIL.

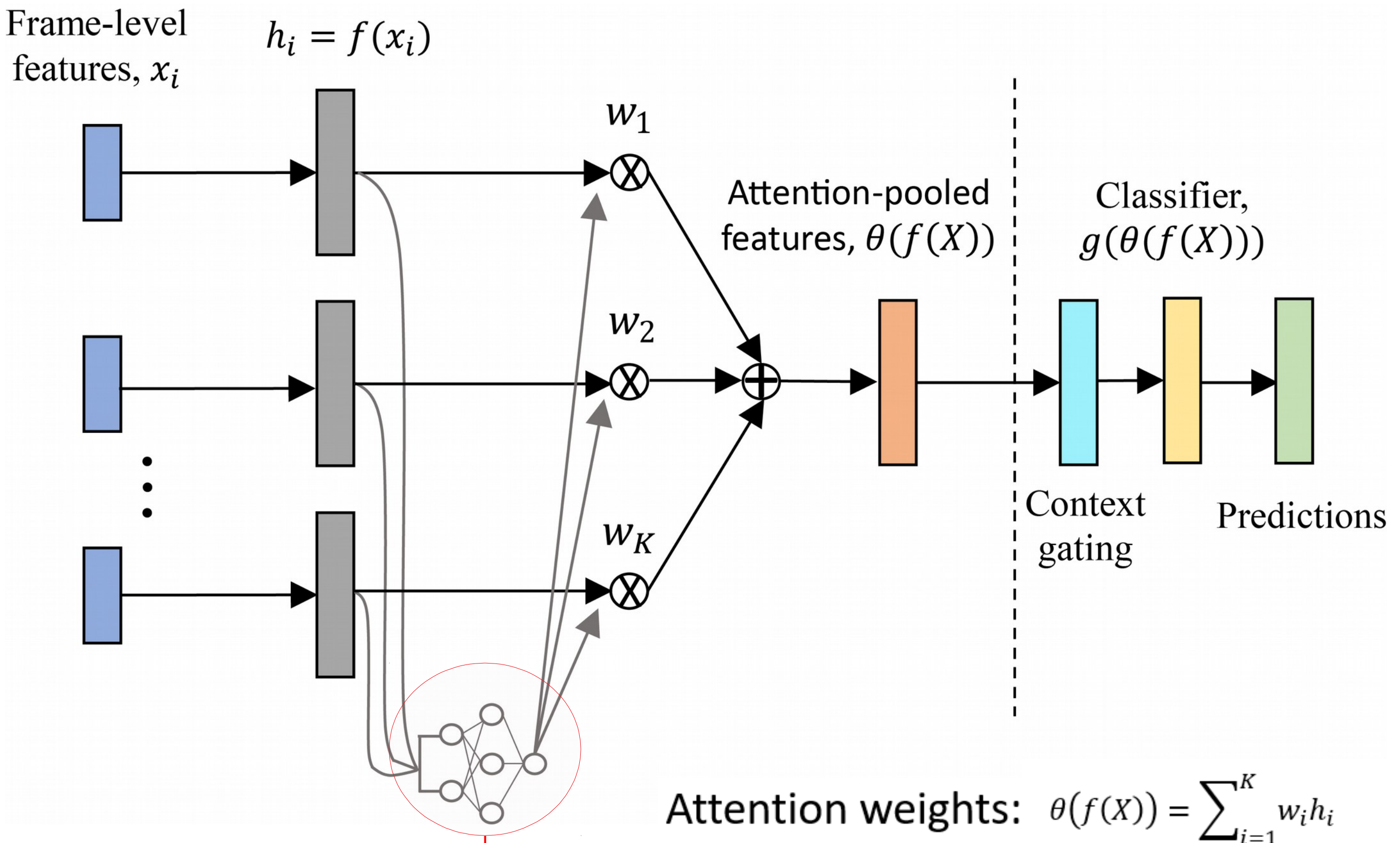
- Multi-instance Learning (MIL). General framework:

$$S(X) = g\left(\theta\big(f(X)\big)\right)$$ Video, Features, pooling, classification.

- Deals with problem of incomplete labels at training set.
- The most common models can be categorized as embedding-based MIL methods.
  - Frame-level logistic model: $\theta\big(f(X)\big) = \frac{1}{K}\sum_{i=1}^{K} f(x_i)$ Pooled features are classified by log model.
  - Deep bag of frames model: $\theta\big(f(X)\big) = \max_{i=1,\dots,K} f(x_i)$ Max pooling to perform the aggregation.

We propose a learnable weighted average of frames as the pooling method.

Attention layers



Attention weights: $\theta(f(X)) = \sum_{i=1}^{K} w_i h_i$

Attention layers: $$w_i = \frac{\exp\{a^T(tanh(Vh_i^T) \odot sigm(Uh_i^T))\}}{\sum_{j=1}^{K} \exp\{a^T(tanh(Vh_j^T) \odot sigm(Uh_j^T))\}}$$

Multi-attention layers

- Multiple sets of parameters for attention network

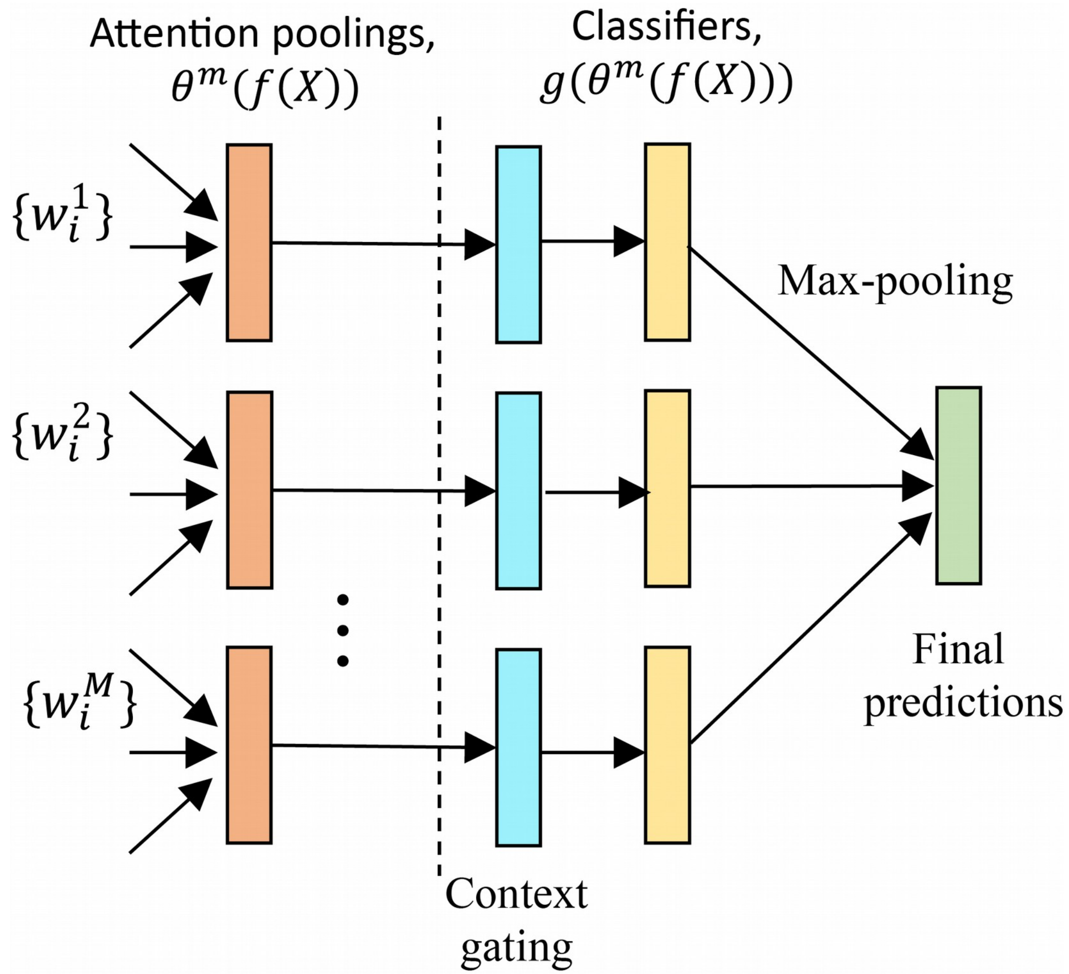
$$\theta^m(f(X)) = \sum_{i=1}^{K} w_i^m h_i$$

- Each pooled feature was then fed into video-level classifier separately:

$$S^m(X) = g\left(\theta^m(f(X))\right)$$

- Finally, the prediction outputs were pooled to obtain the final prediction result:

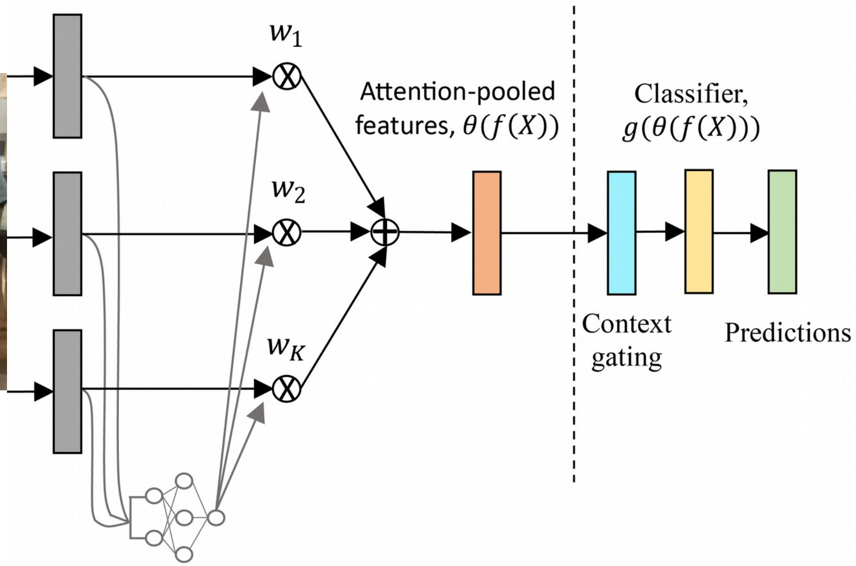
$$S(X) = \max_{m=1,\ldots,M} S^m(X)$$

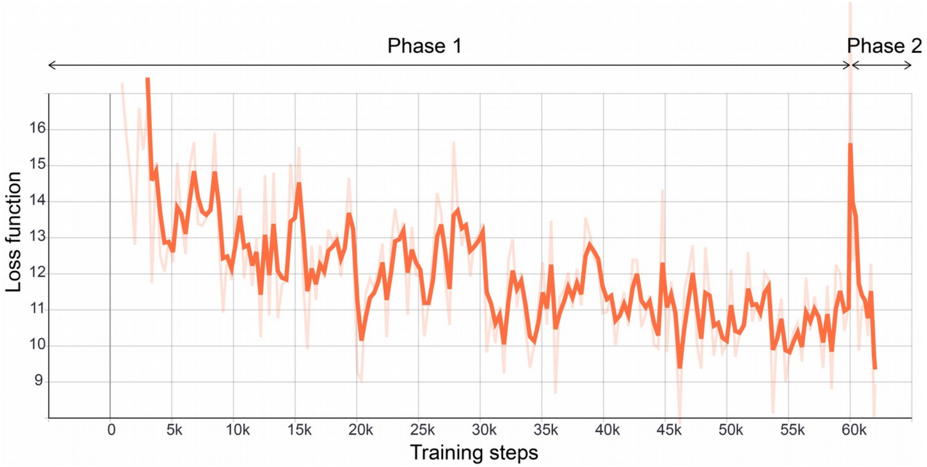


Training procedure

- Phase 1: we trained the model on the 1.4 TB regular training set (whole video). No ‘segment’ concept during phase 1.
- Phase 2: we fine-tuned the model pre-trained on the regular training set using this year segment label dataset.


The entire video





Comparing Results

Model	MAP@100,000
Attention 1 (120 samples, Sparsemax, MoE)	0.769
Attention 2 (subsampling, Softmax, MoE)	0.768
Attention 3 (120 samples, Softmax, Logistic)	0.768
Multi-attention 1 (8 sets, Logistic)	0.771
Multi-attention 2 (8 sets, MoE)	**0.772**
Multi-attention 3 (16 sets, MoE)	**0.772**

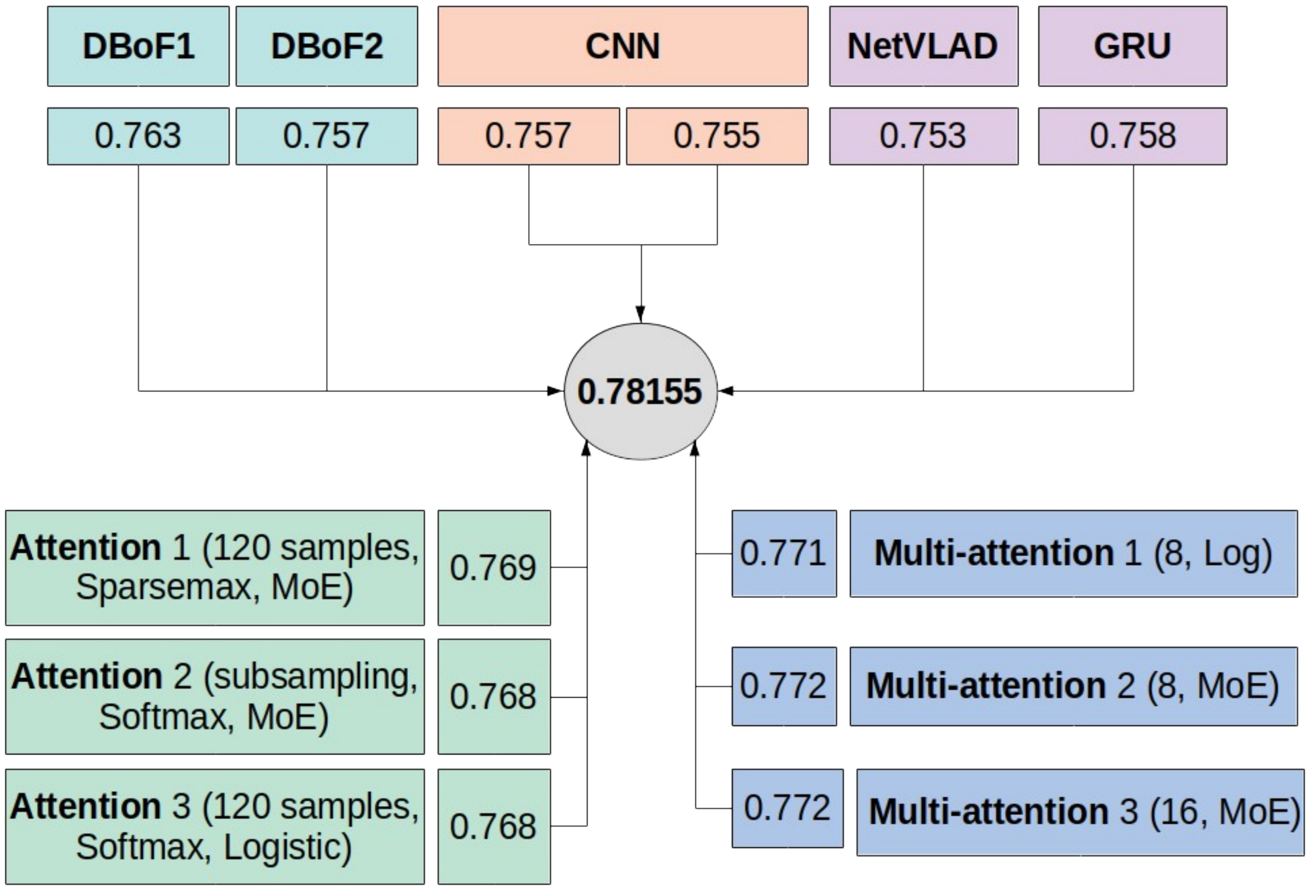
Table 1. Performance of Attention/Multi-attention models.

Model	MAP@100,000
CNN1	0.757
CNN2	0.755
DBoF1	0.763
DBoF2	0.757
NetVLAD	0.753
GRU	0.758

Table 2. Performance of other models.

Under the same training procedure (two phase training)

Final Ensemble



Future work

- **Data augmentation**: producing “virtual” segments by linear combinations of existing segment samples, reverse video, drop random segments.
- **Semi-Supervised** procedure: A typical pseudo-labeling procedure will choose the top scored segments in the test set as new training samples for the models.
- Use **the start time information as another supervisor**. We can add another loss related to segment timing information and the weights put to that segment by the attention network to the loss function.
- **Distillation** using soft labels - mixture of ground truth and teacher model predictions.

Conclusion

- **Resource efficien**t: the size of multi-attention network with MoE classifier is around **150 MB** and the size of models with logistic classifier is around 30 MB.

- All the training jobs were done in GCP **using a single P100**. For attention/multiattention models this took around **6hrs** in phase 1 and 20min in phase 2,

- The proposed model **performed better** than both standard Neural Networks and Pooling via clustering.

Acknowledgement

My Teammates:

Lijun Zhang

Srinath Nizampatnam

Ahana Gangopadhyay