Intro and Recap Detection for Movies and TV Series

Xiang Hao, Kripa Chettiar, Ben Cheung, Vernon Germano and Raffay Hamid
Amazon Prime Video
{xianghao,ksivakum,cheungb,germanov,raffay}@amazon.com

Abstract

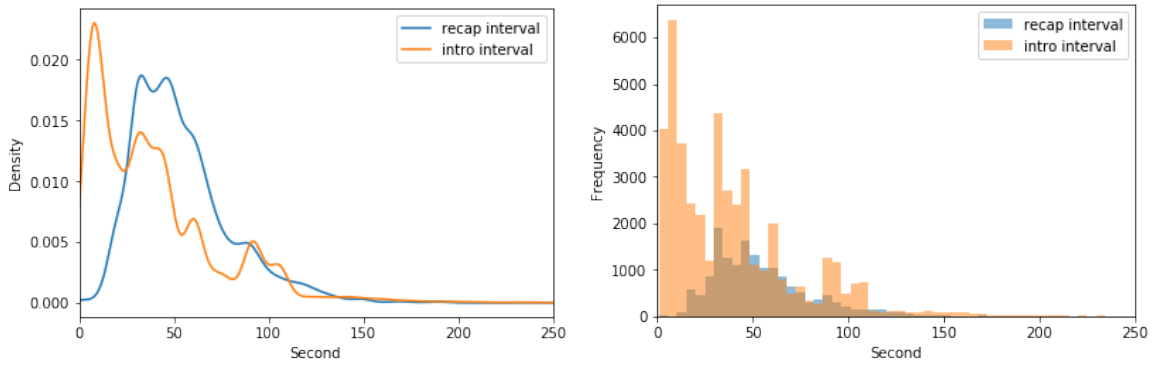
In this supplementary material, we first provide details on the distributions of labeled intro and recap timestamps and then provide more setup details for experiments and comparison of architectures used in the ablation study. After that, we show the performance of our proposed model for different tolerance values on the test data-set. Lastly, we show qualitative detection examples to visualize the results of our proposed method.

1. Data Distribution

Table 1 shows the mean and standard deviation (STD) of labeled timestamps. The standard deviations of labeled timestamps are 141.6, 136.8, 16.2 and 33.7 seconds for *intro*-start, *intro*-end, *recap*-start and *recap*-end respectively. Figure 1 shows the density plot and histogram of *intro* and *recap* intervals. We can see that the *intro* intervals have much more variation compared to *recap* which are clustered at around 40 seconds.

	Recap Start	Recap End	Intro Start	Intro End
Count	15,934	15,934	45,027	45,027
Mean	6.7s	61.0s	139.3s	179.0s
STD	16.2s	33.7s	141.6s	136.8s

Table 1: Mean and STD of intro/recap timestamps



(a) (b)

Figure 1: (a): Density plot of *recap*/*intro* intervals ; (b): Histogram of *recap*/*intro* intervals

2. Timestamps and Tags Conversion in Network Training

Timestamp to Sequence Tags for Training. To feed the sequence labels to our model for training, we need to convert the timestamps to a sequence tags, *i.e.*, we need to tag each of the time intervals with a label (*intro*, *recap* or actual content) inferred from the timestamps. In all our experiments, since we choose one second interval as our temporal granularity, we use ceiling operation for the start timestamp and floor operation for the end timestamp respectively to make sure that our labels are strictly within the timestamps. For example, if the *recap* starts from 3.2s till 10.5s and the *intro* starts from 15.3s till 50.2s in a title, we will pass a vector of length 600 with most of elements as 0 except that the $4^{th} - 10^{th}$ elements as 1 and the $16^{th} - 50^{th}$ elements as 2, where 0, 1 and 2 indicates content, *recap* and *intro*.

Sequence Tags to Timestamp for Metrics. Our trained model predicts the sequence labels for a video title. In order to compare it with the labeled timestamps for evaluating the evaluation metrics, we need to parse the timestamps from the predicted sequence labels. To do this, we take the start and end index from the consecutive *intro* or *recap* sequences respectively as the predicted start and end timestamps. In the vast majority of the cases, there is at most one consecutive sequence for *intro* or *recap*. When there are multiple ones, we simply take the longest consecutive sequence based on the intuition that shorter sequences for *intro* and *recap* are probably noises and less likely to occur. Among all *intro* and *recap* sequences predicted by our proposed model architecture with early fusion, 0.2% and 0% of them have more than one consecutive sequence for *intro* and *recap* respectively.

3. Comparison of Architectures Used in Ablation Study

As mentioned in Section 4.4, in the ablation study we set up the architecture comparison to be incremental where only one change is made for each pair of consecutive architectures being considered in order to distill the effectiveness of using the B-LSTM and CRF components. Table 2 describes the structure of each of the architectures and the incremental change compared to the previous architecture.

#	Architecture Name	Description
1	1-layer-LSTM	LSTM with only one layer + dense layer. Use cross entropy loss
2	LSTM	Add one more LSTM layer compared to architecture 1
3	B-LSTM	Replace both LSTM layers in architecture 2 with B-LSTM layers
4	B-LSTM + Viterbi	Same as architecture 3 but apply Viterbi decoding during inference
5	B-LSTM + CRF	Replace dense layer with CRF layer compared to architecture 3

Table 2: Architecture comparison

4. Model Performance for Different Tolerances

Table 3 shows the performance of our proposed model with early fusion for different tolerance values on the test data-set. Compared to metrics with 1-second tolerance on the validation data-set, we can see that the model also achieve high accuracy with the 70.85% F_1 score for *intro* and 70.57% F_1 score for *recap* on the test data. In addition, we can see that all metrics including precision, recall, and F_1 score increase as the tolerance value increases. For example, when the tolerance is increased from 1 second to 3 seconds, all metrics increase by more than 8%.

Tolerance	Intro			Recap		
	Precision	Recall	F_1	Precision	Recall	F_1
1 second	73.22%	68.64%	70.85%	73.46%	67.89%	70.57%
2 seconds	80.88%	75.82%	78.26%	81.85%	75.65%	78.63%
3 seconds	82.72%	77.54%	80.05%	83.97%	77.60%	80.66%

Table 3: Bi-LSTM+ CRF model performance using early fusion for different tolerances on test data-set

5. Qualitative Results

Figure 2 shows three examples to visualize the result of our proposed method. Ground truth is shown in blue. Correct detections within 1-second tolerance are marked in green, and incorrect detections are marked in orange. The start and the end timestamps of segments are indicated in seconds at the boundaries of the segments. In the first example, the video-title only has *intro*, and we can find the model is able to detect the start timestamp of *intro* accurately and detect the end timestamp of *intro* within 1-second tolerance. In the second example, the video-title has both *intro* and *recap*, and our method can detect both the start and the end timestamps of *recap* accurately and is able to detect the *intro* within 1-second tolerance. As shown in the third case, while our method is able to detect the end timestamp of *intro* within 1-second tolerance, the detected start timestamps is off by 4 seconds, which results in an incorrect detection.

Figure 2: Qualitative results of detection from our method. Correct detections within 1-second tolerance are marked in green, and incorrect detections are marked in orange. The start and the end timestamps of segments are indicated in seconds at the boundaries of the segments.