LiveSeg: Unsupervised Multimodal Temporal Segmentation of Long Livestream Videos Supplementary Materials

Jielin Qiu[1,2], Franck Dernoncourt[1], Trung Bui[1], Zhaowen Wang[1], Ding Zhao[2], Hailin Jin[1]
[1]Adobe Research, [2]Carnegie Mellon University
{jielinq,dingzhao}@andrew.cmu.edu, {dernonco,zhawang,bui,hljin}@adobe.com

1. More Details of Unsupervised Video Summarization Experiment

We used the same key-fragment-based approach for evaluation [18], where the similarity between a machine-generated and a user-defined ground-truth summary is represented by expressing their overlap using the F-Score. This protocol can be directly applied on the user summaries of the SumMe dataset, while its application on TVSum requires to transform the original frame-level annotations into key-fragment-based summaries.

Finally, for a given video and a machine-generated summary, this protocol matches the latter against all the available user summaries for this video and computes a set of F-Scores. For TVSum the final outcome occurs by averaging the computed F-Scores, while for SumMe this output corresponds to the maximum value among the computed F-Scores [8].

2. More Details about WD, GWD, and CCA

2.0.1 Wasserstein Distance

Wasserstein Distance (WD) is introduced in Optimal Transport (OT), which is a natural type of divergence for registration problems as it accounts for the underlying geometry of the space, and has been used for multimodal data matching and alignment tasks [4, 17, 11, 5]. In Euclidean settings, OT introduces WD $\mathcal{W}(\mu,\nu)$, which measures the minimum effort required to "displace" points across measures μ and ν, where μ and ν are values observed in the empirical distribution. In our setting, we compute the temporal-pairwise Wasserstein Distance on both visual features and language features, considering each feature vector representing each frame or transcript embedding, which are $(\mu,\nu) = (V_{2i}, V_{2(i+1)})$ and $(\mu,\nu) = (L_{2j}, L_{2(j+1)})$ for $i, j \in t-1$.

For simplicity without loss of generality, assume $\mu \in P(\mathbb{X})$ and $\nu \in P(\mathbb{Y})$ denote the two discrete distributions, formulated as $\mu = \sum_{i=1}^{n} u_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} v_j \delta_{y_j}$, with δ_x as the Dirac function centered on x. $\Pi(\mu,\nu)$ denotes all the joint distributions $\gamma(x,y)$, with marginals $\mu(x)$ and $\nu(y)$. The weight vectors $u = \{u_i\}_{i=1}^{n} \in \Delta_n$ and $v = \{v_i\}_{i=1}^{m} \in \Delta_m$ belong to the $n-$ and $m-$dimensional simplex, respectively. The WD between the two discrete distributions μ and ν is defined as:

$$\begin{aligned} \mathcal{WD}(\mu,\nu) &= \inf_{\gamma \in \Pi(\mu,\nu)} \mathbb{E}_{(x,y)\sim\gamma}[c(x,y)] \\ &= \min_{\mathbf{T}\in\Pi(\mathbf{u},\mathbf{v})} \sum_{i=1}^{n}\sum_{j=1}^{m} T_{ij} \cdot c\left(x_i, y_j\right) \end{aligned} \tag{1}$$

where $\Pi(u,v) = \left\{T \in \mathbb{R}_{+}^{n\times m} \mid T\mathbf{1}_m = u, T^{\top}\mathbf{1}_n = v\right\}$, $\mathbf{1}_n$ denotes an $n-$dimensional all-one vector, and $c\left(x_i, y_j\right)$ is the cost function evaluating the distance between x_i and y_j. The temporal-pairwise WD on both visual and language features encodes the temporal difference and consistency within the same domain.

2.0.2 Gromov Wasserstein Distance

Classic OT requires defining a cost function across domains, which can be challenging to implement when the domains are in different dimensions [14]. Gromov Wasserstein Distance (GWD) [12] extends OT by comparing distances between samples rather than directly comparing the samples themselves.

Assume there are metric measure spaces $(\mathcal{X}, d_x, \mu)$ and $(\mathcal{Y}, d_y, \nu)$, where d_x and d_y are distances on $\mathcal{X}$ and $\mathcal{Y}$, respectively. We compute pairwise distance matrices D^x and D^y as well as the tensor $L \in \mathbb{R}^{n_x\times n_x\times n_y\times n_y}$, where $L_{ijkl} = L(D^x_{ik}, D^y_{jl})$ measures the distance between pairwise distances in the two domains. Intuitively, $L(d_x(x_1,x_2),\ d_y(y_1,y_2))$ captures how transporting x_1 onto y_1 and x_2 onto y_2 would distort the original distances between x_1 and x_2 and between y_1 and y_2. The discrete Gromov-Wasserstein problem is then defined by:

$$\mathcal{GWD}(p,q) = \min_{\Gamma\in\Pi(p,q)} \sum_{i,j,k,l} L_{ijkl}\Gamma_{ij}\Gamma_{kl} \tag{2}$$

where $(p, q) = (V_{2k}, L_{2k})$ is the visual-language feature pairs. For each tuple (x_i, x_k, y_j, y_l), we compute the cost of altering the pairwise distances between x_i and x_k when splitting their masses to y_j and y_l by weighting them by Γ_{ij} and Γ_{kl}, respectively. The computed GWD across domain is to capture the relationship and dependencies between visual and language domains.

2.0.3 CCA and DCCA

Canonical Correlation Analysis (CCA) is a method for exploring the relationships between two multivariate sets of variables, which can learn linear transformation of two vectors in order to maximize the correlation between them, which is used in many multimodal problems [1, 7]. In our problem, we apply CCA to capture the cross-domain relationship. For visual features V_{2l} and language features L_{2l}, where $l \in t$. We assume $(V_{2l}, L_{2l}) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ has covariances $(\Sigma_{11}, \Sigma_{22})$ and cross-covariance Σ_{12}. CCA finds pairs of linear projections of the two views, $(w_1' V_{2l}, w_2' L_{2l})$ that are maximally correlated:

$$\begin{aligned}(w_1^*, w_2^*) &= \underset{w_1, w_2}{\operatorname{argmax}} \operatorname{corr}\left(w_1' V_{2l}, w_2' L_{2l}\right) \\ &= \underset{w_1, w_2}{\operatorname{argmax}} \frac{w_1' \Sigma_{12} w_2}{\sqrt{w_1' \Sigma_{11} w_1 w_2' \Sigma_{22} w_2}}\end{aligned} \tag{3}$$

Since the objective is invariant to scaling of ω_1 and ω_2, the projections are constrained to have unit variance:

$$(w_1^*, w_2^*) = \operatorname{argmax}_{w_1' \Sigma_{11} w_1 = w_2' \Sigma_{22} w_2 = 1} w_1' \Sigma_{12} w_2 \tag{4}$$

To obtain V_2 and L_2, Deep CCA (DCCA) is applied in the framework for nonlinear feature transformation. If we assign θ_1 and θ_2 to represent the parameters for $f(V_1)$ and $g(L_1)$, respectively, where V_1 and L_1 represents the low-level visual and language features, then the transformation aims at:

$$(\theta_1^*, \theta_2^*) = \underset{(\theta_1, \theta_2)}{\operatorname{argmax}} \operatorname{corr}\left(f\left(V_1; \theta_1\right), g\left(L_1; \theta_2\right)\right) \tag{5}$$

The parameters are trained to optimize this quantity using gradient-based optimization by taking the correlation as the negative loss with backpropagation to update the nonlinear transformation model [1].

3. Ablation Experimental Results of Different Parameters

In this section, we provide more experimental results with different parameters, including α, γ, β_0, α_0, ν_0, κ_0, l_s, init state, and Nmax. The results are shown in Tables 1,2,3,4,5,6,7,8,9, respectively.

Table 1. Comparison of performance with different α.

α	Precision	Recall	F1-score
1	0.673	0.697	0.685
2	0.669	0.694	0.681
3	0.667	0.686	0.676
4	0.666	0.687	0.676
6	0.670	0.693	0.681
8	0.665	0.698	0.681
10	0.666	0.690	0.678

Table 2. Comparison of performance with different γ.

γ	Precision	Recall	F1-score
1	0.673	0.697	0.685
2	0.649	0.684	0.665
3	0.651	0.686	0.668
4	0.673	0.697	0.685
6	0.672	0.691	0.681
8	0.672	0.690	0.681
10	0.669	0.693	0.681

Table 3. Comparison of performance with different β_0.

β_0	Precision	Recall	F1-score
1	0.673	0.697	0.685
2	0.671	0.698	0.684
4	0.664	0.693	0.678
6	0.658	0.685	0.671
8	0.660	0.691	0.675
10	0.660	0.692	0.676

Table 4. Comparison of performance with different α_0.

α_0	Precision	Recall	F1-score
20	0.649	0.689	0.668
60	0.651	0.685	0.668
100	0.655	0.691	0.673
160	0.658	0.684	0.671
180	0.660	0.682	0.671
200	0.663	0.686	0.674
300	0.668	0.690	0.679
400	0.666	0.661	0. 663
500	0.673	0.697	0.685
600	0.671	0.693	0.682

Table 5. Comparison of performance with different ν_0.

ν_0	Precision	Recall	F1-score
1	0.672	0.680	0.676
2	0.673	0.685	0.679
4	0.670	0.678	0.674
6	0.669	0.686	0.677
10	0.671	0.690	0.680
20	0.671	0.693	0.682
100	0.673	0.697	0.685
200	0.670	0.694	0.682
300	0.671	0.695	0.683
500	0.671	0.690	0.680

Table 6. Comparison of performance with different κ_0.

κ_0	Precision	Recall	F1-score
0.25	0.670	0.693	0.681
0.5	0.673	0.697	0.685
0.75	0.668	0.668	0.668
1	0.672	0.697	0.684
1.5	0.671	0.697	0.684
2	0.671	0.696	0.683
2.5	0.668	0.687	0.677

Table 7. Comparison of performance with different l_s.

l_s	Precision	Recall	F1-score
0.5	0.662	0.685	0.673
1	0.673	0.697	0.685
5	0.671	0.696	0.683
10	0.670	0.690	0.680

Table 8. Comparison of performance with different init state.

init state	Precision	Recall	F1-score
1	0.671	0.695	0.683
2	0.672	0.696	0.684
3	0.673	0.697	0.685
4	0.672	0.691	0.681
5	0.672	0.693	0.682
6	0.673	0.697	0.685
8	0.670	0.688	0.679
10	0.672	0.693	0. 682

Table 9. Comparison of performance with different Nmax.

Nmax	Precision	Recall	F1-score
10	0.672	0.696	0.684
50	0.670	0.697	0.683
90	0.673	0.697	0.685
100	0.668	0.692	0.682
150	0.669	0.687	0.678
200	0.673	0.688	0.680

4. HSMM

Hidden Markov Model (HMM) is a statistical model which follows the Markov process to identify the hidden states from a set of observations, which has been widely used in sequential data problems [3, 13, 6], but the state duration distributions are restricted to a geometric form and the number of hidden states must be set a priori [16, 10, 9]. To overcome this, Hidden Semi-Markov Model (HSMM) was proposed [16], where there is a distribution placed over the duration of every state, tweaking the idea into a semi-Markov one. However, the number of hidden states in HMM and HSMM is unknown beforehand, and their patterns are subject to a specific distribution defined over a measure space. HMM with Hierarchical Dirichlet Process (HDP) extension can be used for inferring arbitrarily large state complexity from sequential and time-series data [2, 15]. However, the HDP-HMM's strict Markovian constraints are undesirable in many settings. To overcome the issues, Johnson et al. introduced explicit-duration Hierarchical Dirichlet Process Hidden semi-Markov Model (HDP-HSMM) and provided new methods for sampling inference in the finite Bayesian HSMM [10].

5. The Architecture of DCCA Model

The architecture of the DCCA model and the parameters used in the main paper is shown in Table 10, respectively, where one view contains the visual raw features, and the other view contains the language raw features. The outputs are transformed high-level visual features and language features.

Table 10. DCCA model parameters in the experiments.

Modality	Model Parameters
Visual	[1024, 512, 256]
Language	[1024, 512, 256]

6. Example of Livestream Video

Fig. 1 shows some examples of the videos collected in our dataset.

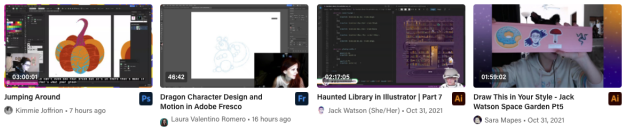

Figure 1. Example of livestream videos.

7. Example of Livestream Transcripts

Some transcript examples are shown in Table 11, which shows the noisy characteristic nature of Livestream video transcripts.

Table 11. Example of livestream transcripts.

Sentence	Offset	Transcript
1	79	Good morning, good morning. My name is Kara Sykes and I am in artist here.
2	91	My light is very bright this morning.
3	94	Sometimes you can turn it down.
12	137	I got more sleep than we have been getting so I was like I'm going Live Today.
20	166	Let's open up Photoshop Screen, but it's going to be we're gonna be working in illustrator.
31	204	Let's go ahead and create.
32	208	I've got a sketch, but I'm actually going to work just without it, but what I want to do here is create some lines.
112	568	Doing letters you never do letters, and I say, I know, but I really wanted to let her his name, so that's what I'm doing currently.
146	698	Now when I work for the area that I want to create, but let's just let's do this OK, I have my do not disturb on because at night we keep it off just so that doesn't wake up.
160	825	Tell you what these type people who create custom type you are amazing.

References

[1] Galen Andrew, R. Arora, Jeff A. Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, 2013.

[2] Matthew J. Beal, Zoubin Ghahramani, and Carl E. Rasmussen. The infinite hidden markov model. In *NIPS*, 2001.

[3] Yan Chang, Weiqing Yang, and Ding Zhao. Energy efficiency and emission testing for connected and automated vehicles using real-world driving data. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2058–2063, 2018.

[4] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. *ArXiv*, abs/2006.14744, 2020.

[5] Pinar Demetci, Rebecca Santorella, Björn Sandstede, William Stafford Noble, and Ritambhara Singh. Gromov-wasserstein optimal transport to align single-cell multi-omics data. *bioRxiv*, 2020.

[6] Jie Dong, Chi Zhang, and Kai xiang Peng. A new multimode process monitoring method based on a hierarchical dirichlet process—hidden semi-markov model with application to the hot steel strip mill process. *Control Engineering Practice*, 110:104767, 2021.

[7] Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision*, 106:210–233, 2013.

[8] Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3090–3098, 2015.

[9] Matthew J. Johnson and Alan S. Willsky. The hierarchical dirichlet process hidden semi-markov model. *ArXiv*, abs/1203.3485, 2010.

[10] Matthew J. Johnson and Alan S. Willsky. Bayesian nonparametric hidden semi-markov models. *J. Mach. Learn. Res.*, 14:673–701, 2013.

[11] John Lee, Max Dabagia, Eva L. Dyer, and Christopher J. Rozell. Hierarchical optimal transport for multimodal distribution alignment. *ArXiv*, abs/1906.11768, 2019.

[12] Gabriel Peyré, Marco Cuturi, and Justin M. Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *ICML*, 2016.

[13] Xuqiang Qiao, Ling Zheng, Yinong Li, Yuqing Ren, Zhida Zhang, Ziwei Zhang, and Lihong Qiu. Characterization of the driving style by state–action semantic plane based on the bayesian nonparametric approach. *Applied Sciences*, 2021.

[14] Ivegen Redko, Titouan Vayer, Rémi Flamary, and Nicolas Courty. Co-optimal transport. *ArXiv*, abs/2002.03731, 2020.

[15] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566 – 1581, 2006.

[16] Shunzheng Yu. Hidden semi-markov models. *Artif. Intell.*, 174:215–243, 2010.

[17] Siyang Yuan, Ke Bai, Liqun Chen, Yizhe Zhang, Chenyang Tao, Chunyuan Li, Guoyin Wang, Ricardo Henao, and Lawrence Carin. Advancing weakly supervised cross-domain alignment with optimal transport. *ArXiv*, abs/2008.06597, 2020.

[18] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, 2016.