Learning to Segment Actions from Visual and Language Instructions via Differentiable Weak Sequence Alignment

Yuhan Shen [1] e-mail: shen.yuh@northeastern.edu

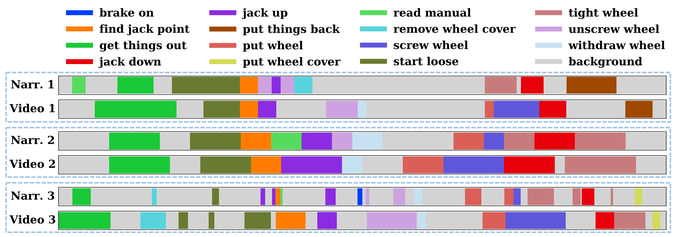
Lu Wang [2] e-mail: wangluxy@umich.edu

Ehsan Elhamifar [1] e-mail: e.elhamifar@norrtheastern.edu

[1] **Northeastern University, Boston, USA** [2] **University of Michigan, Ann Arbor, USA**

Motivation

Unsupervised Action Segmentation in instructional/procedureal videos: localize **task-relevant actions (key-steps)**.



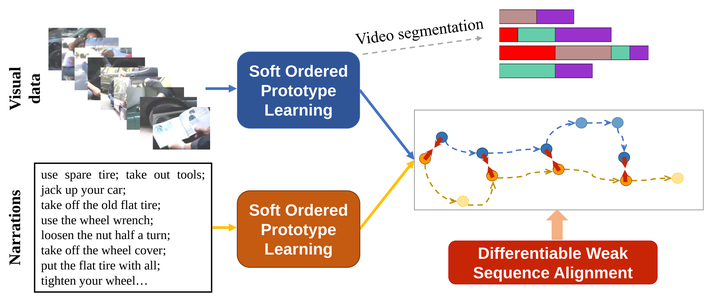
- Visual demonstrations often have **narrations** describing actions.
- The key-steps in videos and narrations are **weakly-aligned**.
  - One may describe one or few key-steps before or after performing them.
  - Some key-steps may be missing in either modality.

Prior Work

- Visual-only unsupervised methods: consider visual similarity for segmentation.
  - **Limitation**: *cannot use the rich information from narrations*.
- Visual-textual unsupervised methods: use both visual data and narrations.
  - **Limitations**: *i) assume temporal alignment between two modalities, which is often violated; ii) mainly rely on one modality; iii) use pre-computed features, cannot perform feature learning*.

Contributions

- Address **task-relevant action (key-step) localization** and **multimodal feature learning** in instructional videos using visual and language data.
- Develop a **Soft Ordered Prototype Learning** method to recover the sequences of visual and linguistic prototypes, representing key-steps.
- Develop a **Differentiable Weak Sequence Alignment** loss that finds the weak one-to-at-most-one alignment between modalities and enables feature learning.
- **Outperform the state-of-the-art** unsupervised methods by about **4.7%** in F1 score on two datasets.



Soft Ordered Prototype Learning

- **Assumption**: key-steps in different videos occur roughly around the same time.
- **Objective**:

$$\min_{\{\boldsymbol{c}_k,\tau_k\}} \sum_n \sum_i -\beta \log\Big(\sum_k e^{-d_{nik}/\beta}\Big), \text{ where } d_{nik} \triangleq \big\|\boldsymbol{y}_{n,i} - \boldsymbol{c}_k\big\|^2 + \gamma\,(\frac{t_{n,i}}{T_n} - \tau_k)^2 \quad (1)$$

- A **soft clustering** method that considers both **feature** vectors and **time-stamps**.

Algorithm 1: Soft Ordered Prototype Learning (SOPL)

Input : $\{(\boldsymbol{y}_{n,i}, t_{n,i})\}_{n,i}, K, \beta \geq 0$

Initialize prototypes, $\{\boldsymbol{c}_k, \tau_k\}_{k=1}^K$

for $iter \leftarrow 1$ **to** $p = 5$ **do**

  Compute $\{d_{nik}\}$ via (1) and soft assignments

  $s_{nik} = \frac{\exp(-d_{nik}/\beta)}{\sum_{j=1}^K \exp(-d_{nij}/\beta)}$

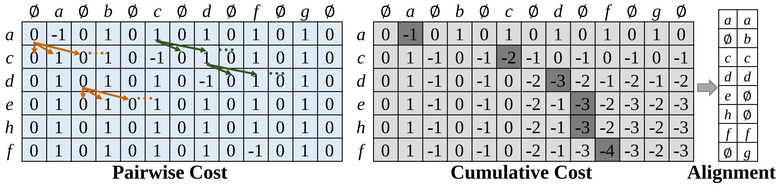
  Update prototypes $[\boldsymbol{c}_k, \tau_k] = \frac{\sum_n \sum_i s_{nik}[\boldsymbol{y}_{n,i},\, t_{n,i}/T_n]}{\sum_n \sum_i s_{nik}}$

Output : Feature and time prototypes $\{\boldsymbol{c}_k, \tau_k\}_{k=1}^K$

Differentiable Weak Sequence Alignment

- Allow **weak sequence alignment** and **multimodal feature learning**.
- **Weak sequence alignment**: find one-to-one alignment while allowing some items to be unmatched.

  Consider two symbolic sequences: $\mathcal{O} = (a, c, d, e, h, f)$ and $\mathcal{O}' = (a, b, c, d, f, g)$.
  - Step 1: insert empty slots in either sequence;
  - Step 2: compute pairwise alignment cost;
  - Step 3: dynamic program to update cumulative cost matrix.



- **Differentiable Weak Sequence Alignment**:

$$\boldsymbol{\Delta}(\mathcal{O}, \mathcal{O}') \triangleq \Big[e^{\delta_{i,j}} / \sum_{j'} e^{\delta_{i,j'}}\Big], \quad \delta_{i,j} \triangleq \begin{cases} \ell(\boldsymbol{o}_i, \boldsymbol{o}'_j) & , j:\ \text{even}, \\ \delta_e & , j:\ \text{odd}, \end{cases} \quad (2)$$

$$\min\nolimits_\beta\{\alpha_1, \alpha_2, \ldots\} = -\beta \log \sum_k e^{-\alpha_k/\beta} \quad (3)$$

Algorithm 2: Forward Propagation

input : Cost matrix $\boldsymbol{\Delta}$; soft-min parameter $\beta \geq 0$

$d_{1,j} \leftarrow \delta_{1,j}, j \in \{1, 2, \ldots, 2q'+1\}$

for $i \leftarrow 2$ **to** q **do**

  for $j \leftarrow 1$ **to** $2q'+1$ **do**

    if j *is odd* **then**

      $d_{i,j} \leftarrow \delta_{i,j} + \min_\beta\{d_{i-1,1}, \ldots, d_{i-1,j}\}$

    else

      $d_{i,j} \leftarrow \delta_{i,j} + \min_\beta\{d_{i-1,1}, \ldots, d_{i-1,j-1}\}$

$\mathcal{L} \leftarrow \min_\beta\{d_{q,1}, \ldots, d_{q,2q'+1}\}$

output : DWSA Loss $= \mathcal{L}$

Algorithm 3: Backward Propagation

input : Matching cost $\boldsymbol{\Delta} \in \mathbb{R}^{q \times 2q'+1}$; Cumulative cost $\mathcal{D}$; soft-min parameter $\beta \geq 0$

$g_{q,j} \leftarrow \frac{e^{-d_{q,j}/\beta}}{\sum_{r=1}^{2q'+1} e^{-d_{q,r}/\beta}}, j \in \{1, \ldots, 2q'+1\}$

for $i \leftarrow q-1$ **to** 1 **do**

  for $j \leftarrow 2q'+1$ **to** 1 **do**

    if j *is odd* **then**

      $g_{i,j} \leftarrow \sum_{r \geq j} g_{i+1,r} e^{(-d_{i,j}+d_{i+1,r}-\delta_{i+1,r})/\beta}$

    else

      $g_{i,j} \leftarrow \sum_{r > j} g_{i+1,r} e^{(-d_{i,j}+d_{i+1,r}-\delta_{i+1,r})/\beta}$

Set $\boldsymbol{G} = [g_{i,j}]$

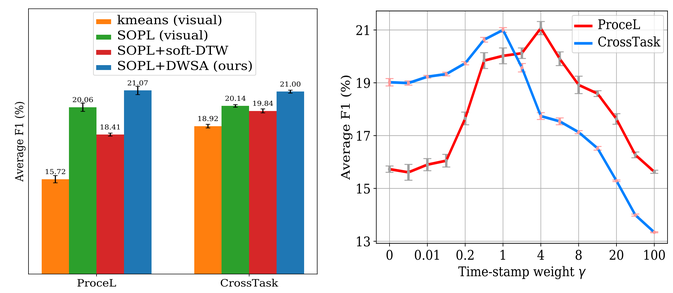
output : $\nabla_{\mathcal{O}}\text{DWSA}(\mathcal{O}, \mathcal{O}') = \big(\frac{\partial \boldsymbol{\Delta}(\mathcal{O}, \mathcal{O}')}{\partial \mathcal{O}}\big)^T \boldsymbol{G}$

Experiments

- **Datasets:** ProceL (Elhamifar et al. ICCV'19) and CrossTask (Zhukov et al. CVPR'19).
- **Evaluation Metrics:** Frame-wise F1, precision, recall, MoF (Mean over Frame).
- **Quantitative Results:** improve F1 score by about **4.7%** on both datasets.

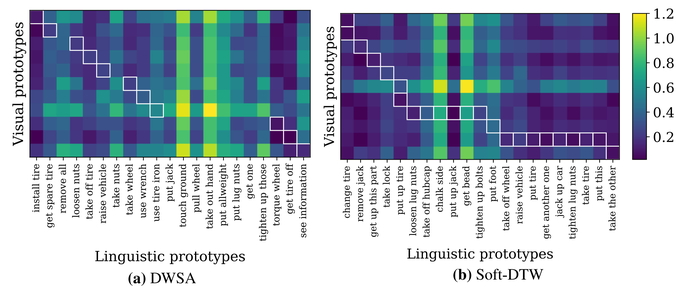
	ProceL			CrossTask		
	F1 (%)	Recall (%)	Precision (%)	F1 (%)	Recall (%)	Precision (%)
Uniform	10.28	9.36	12.41	9.03	9.75	8.69
Alayrac et al.	5.54	3.73	12.25	4.46	3.43	6.80
Kukleva et al.	16.39	30.19	11.69	15.27	35.90	9.82
Elhamifar et al.	14.00	26.70	9.49	16.30	**41.60**	10.14
Ours	**21.07**±0.25	**31.78**±0.37	**16.51**±0.09	**21.00**±0.09	35.46±0.14	**15.21**±0.07

- **Ablation Study:** Variants of our method (left), effect of time-stamp weight γ (right).



- Alignment between visual and linguistic prototypes before/after learning via DWSA:

- Learned visual and linguistic prototype distances using DWSA and Soft-DTW:



- Localization results of different methods: