Induce, Edit, Retrieve: Language-grounded Multimodal Schemata for instructional Video Retrieval

Yue Yang  Joongwon Kim  Artemis Panagopoulou

Mark Yatskar  Chris Callison-Burch

Department of Computer and Information Science
University of Pennsylvania
Motivation

• Schema: a set of rules people use to perform everyday tasks.
• Schema can be generalized.
• When facing new tasks, people use prior knowledge. (Chen et al., 2004)
IER Overview

• Our Induce, Edit, Retrieve (IER) system:
  • **Induce:**
    • Input: a task name, a set of related videos.
    • Output: a set of sentences as the schema.
  • **Edit:**
    • Given an unseen task.
    • Use language models to modify schema.
  • **Retrieve:**
    • Improve video retrieval using schema.

Figure 1. An example from our IER system, which first induces a schema for *Bake Chicken* using a set of videos. Then it edits the steps in the schema to adapt to the unseen task *Bake Fish* (the tokens that have been edited are highlighted). Finally, IER relies on the edited schema to help retrieve videos for *Bake Fish*. 
Schema Induction

Objective: Construct schemata on tasks.
Task: Bake Chicken

Video 1

... YouTube Videos (Learning Data)

Video n

Clip-Step Alignment

WikiHow to do anything

1M human written steps

- Wash the chicken thoroughly.
- Marinate the chicken.
- Season the drumsticks.
- Bake the blackened chicken in the oven or on the grill.

Schema of Bake Chicken

- Wash the chicken thoroughly
- Season your drumsticks
- Marinate the chicken
- Insert a roasting thermometer into the thigh
- Sprinkle the ginger on the chicken
- Bake the blackened chicken in the oven
- ... ...

- Align wikiHow steps with videos.
- MIL-NCE as the video-text model.
- We induced 21,299 schemata using 1.2 M videos from Howto100m.
Schema Editing

- Given an **unseen** task without videos, edit **existing** schema.
- Find the most similar task in the schema library:

  - **Textual Similarity** = cosine similarity of SBERT embeddings
  - **Visual Similarity** (Google Image Search)
  - **Task Similarity** = max(Textual Similarity, Visual Similarity)
Schema Editing

• Editing Module 1: **Object Replacement**
  • Every task has a main object, e.g., “chicken” of “Bake Chicken”.
  • Use POS tagger to find the 1st occurred noun as main object.
  • Replace the objects in all steps.

<table>
<thead>
<tr>
<th>Object Replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cook Ham $0.86 \rightarrow$ Cook Lamb</td>
</tr>
<tr>
<td>Put the <strong>ham</strong> in the oven.</td>
</tr>
<tr>
<td>Put the <strong>lamb</strong> in the oven.</td>
</tr>
<tr>
<td>Clean a Guitar $0.84 \rightarrow$ Build a Violin</td>
</tr>
<tr>
<td>Use a polish for particularly dirty <strong>guitars</strong>.</td>
</tr>
<tr>
<td>Use a polish for particularly dirty <strong>violins</strong>.</td>
</tr>
<tr>
<td>Trap a Rat $0.84 \rightarrow$ Trap a Rabbit</td>
</tr>
<tr>
<td>Bait and set snap <strong>rat</strong> traps.</td>
</tr>
<tr>
<td>Bait and set snap <strong>rabbit</strong> traps.</td>
</tr>
</tbody>
</table>
Schema Editing

• Editing Module 2: **Step Deletion**
  • Delete the steps no longer suitable for the new target task.
  • “Insert a roasting thermometer into the thigh” of “Bake Chicken” ❌ “Bake Fish”
  • Sentence BERT pre-trained on QA pairs.
    • Compute the score of (task, step).
    • if (source task, step) >> (target task, step), delete, otherwise include

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Task 2</th>
<th>Score</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transplant a Young Tree</td>
<td>Remove a Tree</td>
<td>0.89</td>
<td>Delete Fill your pot with a balanced fertilizer.</td>
</tr>
<tr>
<td>Fix a Toilet</td>
<td>Remove a Toilet</td>
<td>0.85</td>
<td>Delete Test out the new flapper.</td>
</tr>
<tr>
<td>Brush a Cat</td>
<td>Brush a Long Haired Dog</td>
<td>0.87</td>
<td>Include Comb and groom your pet.</td>
</tr>
</tbody>
</table>
Schema Editing

- Editing Module 3: **Token Replacement**
  - Use *masked language model* to replace the token with the lowest probability.
  - “Season the *drumstick*” in “Bake Chicken”
  - Mask the token “Season the <mask>”.
  - Use a prompt: How to [TASK]? [STEP]
    - How to Bake Fish? “Season the <mask>”.
  - Predict a new token from vocabulary.
    - <mask> --> fish, “Season the fish”

<table>
<thead>
<tr>
<th><strong>Token Replacement</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare Fish ( \xrightarrow{0.82} ) Prepare Crabs</td>
</tr>
<tr>
<td>Cut the <em>fins</em> from the <em>fish</em> using <em>kitchen shears</em>.</td>
</tr>
<tr>
<td>Cut the <em>shells</em> from the <em>crabs</em> using <em>steel scissors</em>.</td>
</tr>
<tr>
<td>Make Healthy Donuts ( \xrightarrow{0.88} ) Bake Healthy Cookies</td>
</tr>
<tr>
<td>Slice your <em>donuts</em> into <em>disks</em>.</td>
</tr>
<tr>
<td>Slice your <em>cookies</em> into <em>squares</em>.</td>
</tr>
<tr>
<td>Wash Your Bike ( \xrightarrow{0.84} ) Wash a Motorcycle</td>
</tr>
<tr>
<td>Clean the <em>bike</em> <em>chain</em> with a <em>degreaser</em>.</td>
</tr>
<tr>
<td>Clean the <em>motorcycle</em> thoroughly with a <em>towel</em>.</td>
</tr>
</tbody>
</table>
Schema-Guided Video Retrieval

- Query: Task Name (short) Retrieve long multi-minute videos.
- Global Matching (use task name only)
- Step Aggregation (use schemata to expand task name)

Use the task name "Bake Fish" as Query

With Schema
- Wash the fish
- Sprinkle the sauce on the fish
- Preheat the oven
- Bake the blackened fish in the oven
Experiment-Datasets

- **Howto-GEN** (a new split of Howto100M)
  - 3365 tasks, 2,184 unique main objects.
  - Random select 500 tasks for training, 500 for validation, 2365 for test.

- **COIN** (**CO**mprehensive **IN**structional video analysis)
  - 180 tasks, 11,827 videos.
  - Unseen tasks, e.g., “Blow Sugar”, “Make Youtiao”, etc.

- **Youcook2**
  - 89 recipes – tasks, 2,000 long videos.
Experiment - Baselines

- **Generation Models:**
  - T5, GPT-2-large.
  - GPT-3: Zero-shot generation (How to [Task Name]? Give me several steps.)

- **Goal-Oriented Script Construction**
  - Given the input task name, retrieve the set of desired steps from wikiHow.

- **Oracle Schemata (human written, upper bound)**
  - Howto-GEN (from wikiHow)
  - COIN/Youcook2 (Human annotation)
Results:
Results:

Figure 3. Retrieval performance by video length (in the number of clips). We group the test videos of Youcook2 by the number of clips per video and compute the mean rank for each group.

IER helps more for longer videos
Results:

Figure 4. Retrieval performance by task similarity. We sort the test tasks of Howto-GEN based on their task similarity ($G$) and compute their mean rank for every batch of 400 tasks.

Editing helps more when task similarity is low.
Results:

<table>
<thead>
<tr>
<th>Method</th>
<th>P@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>Med r↓</th>
<th>MRR↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>full</td>
<td>54.4</td>
<td>37.3</td>
<td>50.1</td>
<td>10.0</td>
<td>.231</td>
</tr>
<tr>
<td>– mask</td>
<td>53.7</td>
<td>36.3</td>
<td>49.3</td>
<td>11.0</td>
<td>.229</td>
</tr>
<tr>
<td>– deletion</td>
<td>53.6</td>
<td>36.9</td>
<td>49.8</td>
<td>11.0</td>
<td>.230</td>
</tr>
<tr>
<td>– replacement</td>
<td>51.5</td>
<td>34.9</td>
<td>47.3</td>
<td>12.0</td>
<td>.220</td>
</tr>
<tr>
<td>– all</td>
<td>45.5</td>
<td>31.0</td>
<td>43.1</td>
<td>15.0</td>
<td>.199</td>
</tr>
</tbody>
</table>

All three editing modules are beneficial.
Results:

<table>
<thead>
<tr>
<th>Model</th>
<th>P@1↑</th>
<th>R@5↑</th>
<th>R@10↑</th>
<th>Med r↓</th>
<th>MRR↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIL-NCE</td>
<td>48.3</td>
<td>37.1</td>
<td>52.8</td>
<td>9.5</td>
<td>.227</td>
</tr>
<tr>
<td>+schema</td>
<td>57.2</td>
<td>42.2</td>
<td>57.8</td>
<td>7.0</td>
<td>.256</td>
</tr>
<tr>
<td>CLIP [38]</td>
<td>58.9</td>
<td>44.9</td>
<td>58.8</td>
<td>6.0</td>
<td>.264</td>
</tr>
<tr>
<td>+schema</td>
<td>65.0</td>
<td>47.4</td>
<td>60.8</td>
<td>5.5</td>
<td>.282</td>
</tr>
</tbody>
</table>

Table 5. Retrieval performance on COIN using MIL-NCE and CLIP as the matching functions. +schema represents using schema induced by IER (MIL-NCE as matching function) for retrieval.

Schemata can be reused by different video-text model.
Conclusion

• We propose a schema induction and generalization system that improves instructional video retrieval performance.
• We demonstrate that the induced schemata benefit video retrieval on unseen tasks, and our IER system outperforms other methods.
• In the future, we plan to investigate the structure of our schemata.
Thank you!