

LUSE: Using LLMs for Unsupervised Step Extraction in Instructional Videos

Chuyi Shang[†], Emi Tran[†], Medhini Narasimhan, Sanjay Subramanian, Dan Klein, Trevor Darrell
UC Berkeley

{chuyishang, emitran, medhini, sanjayss, klein, trevordarrell}@berkeley.edu

Abstract

In this work, we introduce an unsupervised language-only approach to automatically segment and identify steps in instructional videos. Parsing a video into its steps has several use cases - training video action recognition models to recognize steps, creating visual summaries highlighting relevant steps, identifying mistakes in steps of the video, and retrieving/localizing steps in the video to name a few. Our framework LUSE, zero-shot prompts a Large Language Model (LLM) to extract steps from the transcript of a single instructional video. Next, the steps across several videos of the same task are consolidated to generate a general set of steps for the task, via a second pass through the LLM, and are then localized back in the transcript of each video. Existing datasets for steps rely on manual annotations which are expensive to collect and oftentimes subjective. Our fully automated approach overcomes these issues and generates competitive quality step labels, as highlighted by our qualitative examples. Furthermore, we fine-tune a state-of-the-art image captioning model on our generated steps to show that the resulting output has better qualitative step descriptions and denser coverage compared to existing manually annotated datasets.

1. Introduction

Instructional videos, such as tutorials, how-tos, and walkthroughs, are one of the most viewed types of videos on the internet. The processing of these instructional videos, such as understanding the task at hand, generating step labels, and localizing them, can be extremely useful for a variety of downstream tasks, such as video narration and mistake detection. However, a large problem with instructional video datasets is that they are very costly and labor-intensive to annotate. Datasets such as COIN, for example, required each label to be looked over by 3 different humans, and the final dataset of 11,000 videos is estimated to have taken over 600 labor hours [15]. As a result, there is often an

observed tradeoff between dataset size and annotation quality - many instructional video datasets are very small, and larger ones have noisy or poor-quality annotations. This raises the concern of scalability and annotation subjectivity, which will only become more important as more data is needed for new models.

As a result, we wish to develop a more scalable and accurate process for annotation. Our first goal is to (1) leverage pre-trained GPT language models to automatically generate steps from videos. We perform this without the need of human supervision by feeding the Automatic Speech Recognition (ASR) transcripts for each video to our LLM. Next, we group similar tasks together, and feed the sets of steps into a second LLM pass in order to achieve step generalizability and reduce variance in our generated steps. We then use Drop-Dynamic Time Warping (Drop-DTW [3]) to localize these generated steps back into the video, extracting the corresponding timestamps for each step label.

Additionally, we also fine-tuned an existing image captioning model on our generated step labels. This allows us to test the quality of our generated steps and to generate steps on any new video, relying purely on the visual features and not constrained by the availability of a transcript. Qualitatively comparing the labels generated by fine-tuning BLIP-2 with the ground-truth labels from COIN, we find that step labels auto-generated by LUSE are far more nuanced and descriptive compared to the COIN ground-truth.

2. Related Work

2.1. Language models for Vision Language Tasks

Video-Language models map videos and language onto a shared embedding space, and use it to perform a variety of tasks, such as action recognition, action retrieval, and other downstream tasks. Recently, many such models have leveraged Large-Language Models (LLMs) in their design, using frozen large language models to reduce computational costs. Some examples include BLIP-2 [6] and LaViLa [16], which use LLMs to generate captions and dense video narrations. Other models use Large Language Models to perform Visual Question Answering (VQA), such as

[†]Equal contribution

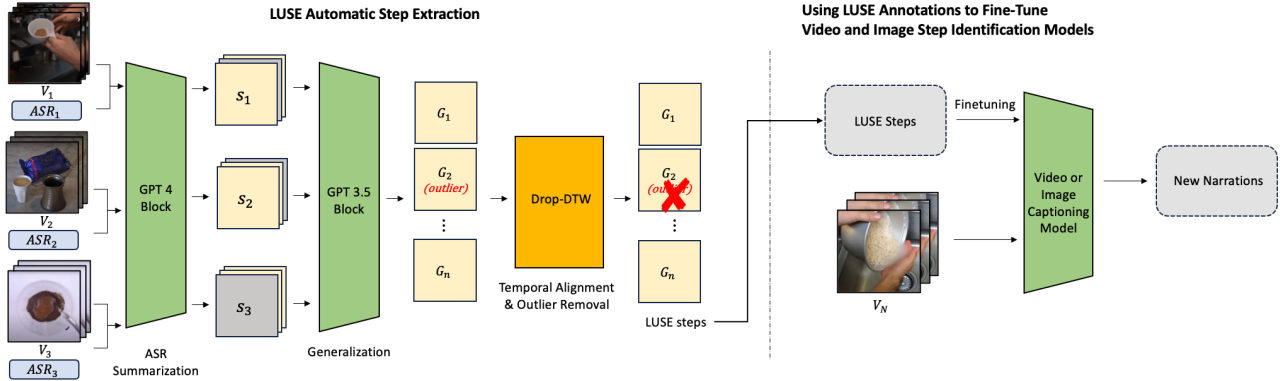


Figure 1: **LUSE** is a two-pass LLM approach for automatic step label generation for each task category. The first LLM pass extracts steps from each video using its ASR transcript, and its output includes both main steps (yellow) and noise steps (gray). A second LLM pass removes drops outliers and generalizes a new set of steps for each task category. Drop-DTW [3] is used to temporally align steps to video and remove recurring outlier steps (i.e. advertisements, ending credits, etc.) common in online instructional videos. We use LUSE steps to finetune video and image captioning models, without any narration/ASR inputs.

CodeVQA [14], which uses LLMs to generate code to answer questions about videos. Our work is most similar to previous attempts to discover steps without supervision [1, 4, 5, 12, 11], but these are often confined to small-scale datasets, which usually do not have the breadth of tasks that larger datasets have.

2.2. Instructional Video Datasets

The collection of large-scale instructional video datasets have been crucial in the development of models that learn task structure and representations from videos, in addition to the performance of various downstream tasks such as step recognition, step forecasting, and mistake step detection. Datasets such as COIN [15], CrossTask [17], and HowTo100M [9] are more general in terms of task content, while other datasets such as Assembly-101 [10] and Ikea ASM [2], contain task sets that are more domain-specific, such as furniture assembly.

3. Unsupervised Step Extraction Using LLMs

In this work, we introduce an approach to automatically detect steps in instructional videos. We adopt a two part approach - first, we use a Large Language Model (LLM) to extract steps from an instructional video transcript. In addition, we localize each step back into the video using drop-dynamic time warping (Drop-DTW) [3], extracting the start and end timestamps for each step. Finally, using our localized steps as ground-truth annotations, we fine-tune a video/image captioning network to generate steps for any given video clip. This approach allows us to generate steps without the need for human supervision. Moreover, it allows our labels and subsequent captioning/narration to not

be confined by the often rigid set of step labels present in dataset annotations.

3.1. Generating Steps From Transcript

The top tab of Fig. 1 displays our step generation process using a two-pass approach to prompt our LLM. For each video, we first take the noisy ASR transcript and pre-process it. We then pass the processed transcript ASR_i to an LLM and prompt it to identify the key steps S_i . However, the steps generated in this fashion still display large variance across different videos, even for videos of the same task category. For example, making Turkish Coffee is very different from making espresso or cold brew, causing the generated sets of steps to be different as well. Moreover, there are inherent variations in videos. For example, some videos in the *make coffee* category may have *grind the coffee beans* described verbally as a step but not show it on camera, whereas other videos do. As a result, we perform a second LLM pass to reduce some of this variance and to get a more generalizable set of steps for each type of task. We first group the videos by their task type, such as *make coffee*, *tie a tie*, or *install curtains*. We then collect the previously generated steps for each video in the group, and prompt the LLM for a generalized set of steps G_i given all the sets of steps for the same task. These sets of steps generated by the second LLM pass are the general set of steps that we will be treating as ground-truth step labels.

3.2. General Step Localization and Annotation.

After generating a set of generalized steps G_i for each task category, we localize them back into the video, both to see if they are valid and as a preprocessing method for other downstream tasks. To do this, we take the ASR



Figure 2: **Generalization and Temporal Localization:** The top row displays Turkish coffee making video frames and the bottom row displays pour-over butter coffee frames. LUSE leverages 2-LLM passes to *automatically* extract and generalize steps to any coffee recipe, while still being *object specific* and *action specific*.

(Automatic Speech Recognition) transcript of each video ASR_i and match them to the generated steps, after embedding them using a pre-trained sentence embedding module. To align our generated steps with the transcript, we use Drop-Dynamic Time Warping (Drop-DTW) [3], which is an algorithm for sequence-to-sequence alignment of variable length sequences. Drop-DTW provides unique advantages over normal Dynamic Time Warping because of its ability to drop outliers, which is especially useful in our context. This is because instructional videos often contain segments that do not directly correspond with a step. For example, many videos have a lengthy introduction where the author talks about the inspiration for their recipe, or segments in the middle where the author goes on a tangent and tells a personal story. Drop-DTW allows us to identify and drop these outliers, which allows us to achieve cleaner and more precise step localization.

3.3. Video and Image Finetuning

After we generate step labels and localize them, we treat them as new ground-truth annotations. We then use these new annotations to fine-tune the video captioning model TimeSFormer [8], which is pre-trained on HowTo100M [9]. This model takes a video with temporal segments as input and outputs classified labels to the nearest LUSE-COIN (generated) label. We then test our fine-tuned TimeSFormer model by segmenting our input video into 10-second clips and inferring the label for each clip.

To eliminate the need for pre-segmenting our videos, we also test LUSE annotations using a single-image captioning model so that we can operate on a frame-level. We chose to use the BLIP-2 image captioning model [6] that was pre-trained on COCO [7] for captioning, and fine-tuned it on our LUSE-generated annotations for the COIN dataset. We sample captions at 1 FPS and randomly pick a caption in every 10 second interval to represent the final step output.

4. Experiments

We evaluate LUSE-generated steps against existing COIN annotations. We do this by fine-tuning the video-language models BLIP-2 [6] and distant-supervision TimeSFormer [8] using our generated step annotations.

Because our LUSE labels are different from the COIN labels, we use MPNet [13] to convert our generated LUSE labels to their closest embedded matching COIN labels. To do this, we first embed both the generated labels and the COIN labels using MPNet. We then calculate cosine similarities to select the COIN label with the highest similarity score.

4.1. Qualitative Comparison

Generalization and Temporal Alignment. We show two versions of a coffee making task in Fig. 2 to demonstrate the generalized labels LUSE outputs and their corresponding automatic temporal alignments. We can see that even though the two methods of preparing coffee (Turkish Coffee and Butter Pour-Over Coffee) are very different, our generated steps are generalizable across different variations of the same tasks. Moreover, the use of LLMs in our step generation often allows for labels that are less rigid and more descriptive than the COIN human annotations. Finally, for more specialized tasks, LLMs may possess more domain knowledge than individual human annotators, allowing for more precise and relevant generated steps.

Dense Coverage and Descriptivity. We show in Fig. 4 a comparison of steps for the same *making burgers* video between LUSE and COIN. We can see that our generated steps provide denser annotation coverage, with 7 key steps being captured in comparison to the 3 labeled by the COIN ground-truth annotations. Moreover, our generated steps are more precise than the ground truth annotations. For example, in the context of making burgers, the generated step *form the mixture into patties* is more descriptive than the



COIN GT	BLIP2 + GPT	Sentence-transformer (MPNet)
1. "pour the orange juice into the cup" 2. "cut oranges" 3. "juice the oranges"	1. 'place a tray of orange juice on top of a wooden table' 2. 'cut an orange with a knife' 3. 'hold an orange in front of a cutting board' 4. 'fill a food processor with oranges' 5. 'view the word "cookist" in a black and white photo' 6. 'arrange a bunch of pictures of people on a wall'	1. pour the orange juice into the cup' 2. 'cut oranges' 3. 'cut oranges' 4. 'cut oranges' 5. 'paint on the paper', 6. 'paste and level the wallpaper'
Example Frame 	Example Outlier Frame 	

Figure 3: **BLIP2 FT.** Outlier frames like outros or ending frames (blue) get captioned as well. We are working on modules to automatically remove these.

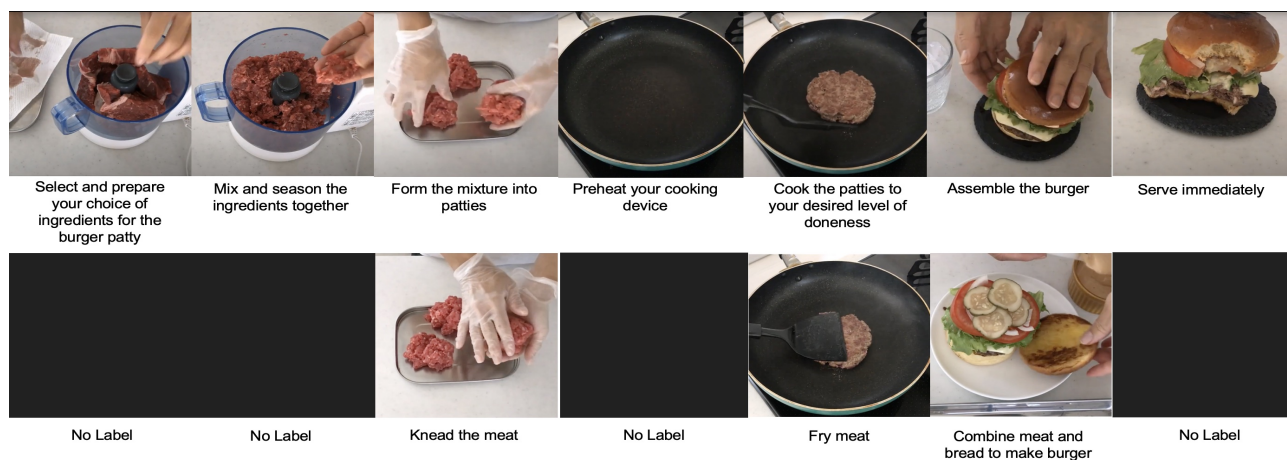


Figure 4: **Dense Coverage and Descriptivity** The top row displays LUSE-generated labels, while the bottom row displays COIN ground truth [15] labels provided by 3 human annotators (per video). Labels are matched to corresponding action frame. We show that (1) LUSE steps are denser in coverage pairing to video frames and (2) are more comprehensive, generalized, and nuanced than that of COIN. For coverage, LUSE outputs 7 out of 7 key steps, while only 3 out of 7 are manually annotated (and checked by 3 people) in COIN [15].

COIN ground-truth label *knead the meat*. This descriptiveness may be a result of our denser annotations, which allows longer steps to be broken down into smaller, more descriptive steps. This increased descriptiveness can also be seen in Fig. 3, which compares our fine-tuned BLIP-2 output to COIN ground-truth annotations. Specifically, we capture more visual information, such as information about the orange juice being in a tray and the table being wooden (step 1). We also receive more instruction details, such as *fill a food processor with oranges* (step 4) and *hold an orange in front of the cutting board* (step 3), which are important instructions that are not included in the ground-truth labels.

5. Conclusion

LUSE is a promising method of unsupervised step extraction from instructional videos. Currently, we have

achieved good qualitative results from labels generated in this manner. In the future, we will obtain quantitative results and evaluate performance on downstream tasks. Some more immediate improvements can come in the form of removing outlier video segments when captioning new videos. For example, in 3, we can see that the outro fade-out frame of the video gets sampled and captioned. As a result, BLIP-2 labels the steps *"view the word 'cookist' in a black and white photo"* and *"arrange a bunch of pictures with people on the wall"*, which brings down accuracy considerably. In the future, we can consider various modules to remove outliers such as Drop-DTW [3] or another GPT pass to remove irrelevant frames or captions.

References

- [1] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos, 2016. 2
- [2] Yizhak Ben-Shabat, Xin Yu, Fatemehsadat Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. 2020. 2
- [3] Nikita Dvornik, Isma Hadji, Konstantinos G. Derpanis, Animesh Garg, and Allan D. Jepson. Drop-dtw: Aligning common signal between sequences while dropping outliers, 2021. 1, 2, 3, 4
- [4] Daniel Fried, Jean-Baptiste Alayrac, Phil Blunsom, Chris Dyer, Stephen Clark, and Aida Nematzadeh. Learning to segment actions from observation and narration, 2020. 2
- [5] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Juergen Gall. Unsupervised learning of action classes with continuous temporal embedding, 2019. 2
- [6] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 1, 3
- [7] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 3
- [8] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. Distant supervision. 3
- [9] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips, 2019. 2, 3
- [10] F. Sener, D. Chatterjee, D. Shelepov, K. He, D. Singhanian, R. Wang, and A. Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. *CVPR 2022*. 2
- [11] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video, 2018. 2
- [12] Ozan Sener, Amir Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections, 2016. 2
- [13] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding, 2020. 3
- [14] Sanjay Subramanian, Medhini Narasimhan, Kushal Khangaonkar, Kevin Yang, Arsha Nagrani, Cordelia Schmid, Andy Zeng, Trevor Darrell, and Dan Klein. Modular visual question answering via code generation, 2023. 2
- [15] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis, 2019. 1, 2, 4
- [16] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *arXiv preprint arXiv:2212.04501*, 2022. 1
- [17] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos, 2019. 2