Unsupervised Learning from Narrated Instruction Videos

Jean-Baptiste Alayrac*† Piotr Bojanowski* Nishant Agrawal *‡ Josef Sivic*
Ivan Laptev* Simon Lacoste-Julien†

Abstract

We address the problem of automatically learning the main steps to complete a certain task, such as changing a car tire, from a set of narrated instruction videos. The contributions of this paper are three-fold. First, we develop a new unsupervised learning approach that takes advantage of the complementary nature of the input video and the associated narration. The method solves two clustering problems, one in text and one in video, applied one after each other and linked by joint constraints to obtain a single coherent sequence of steps in both modalities. Second, we collect and annotate a new challenging dataset of real-world instruction videos from the Internet. The dataset contains about 800,000 frames for five different tasks[1] *that include complex interactions between people and objects, and are captured in a variety of indoor and outdoor settings. Third, we experimentally demonstrate that the proposed method can automatically discover, in an* unsupervised manner*, the main steps to achieve the task and locate the steps in the input videos.*

1. Introduction

Millions of people watch narrated instruction videos[2] to learn new tasks such as assembling IKEA furniture or changing a flat car tire. Many of such tasks have large amounts of videos available on-line. For example, querying for "how to change a tire" results in more than 300,000 hits on YouTube. Most of these videos, however, are made with the intention to teach other people to perform the task and do not provide direct supervisory signal for automatic learning algorithms. Developing unsupervised methods that could learn tasks from myriads of instruction videos on the Internet is therefore a key challenge. Such automatic cognitive ability would enable constructing virtual assistants and smart robots that learn new skills from the Internet to, for example, help people achieve new tasks in unfamiliar situations.

In this work, we consider instruction videos and develop a method that learns a sequence of steps, as well as their textual and visual representations, required to achieve a certain task. For example, given a set of narrated instruction videos demonstrating how to change a car tire, our method automatically discovers consecutive steps for this task such as *loosen the nuts of the wheel*, *jack up the car*, *remove the spare tire* and so on as illustrated in Figure 1. In addition, the method learns the visual and linguistic variability of these steps from natural videos.

Discovering key steps from instruction videos is a highly challenging task. First, linguistic expressions for the same step can have high variability across videos, for example: "...Loosen up the wheel nut just a little before you start jacking the car..." and "...Start to loosen the lug nuts just enough to make them easy to turn by hand...". Second, the visual appearance of each step varies greatly between videos as the people and objects are different, the action is captured from a different viewpoint, and the way people perform actions also vary. Finally, there is also a variability of the overall structure of the sequence of steps achieving the task. For example, some videos may omit some steps or change slightly their order.

To address these challenges, in this paper we develop an unsupervised learning approach that takes advantage of the complementarity of the visual signal in the video and the corresponding natural language narration to resolve their ambiguities. We assume that the same ordered sequence of steps (also called script in the NLP literature [27]) is common to all input videos of the same task, but the actual sequence and the individual steps are unknown and are learnt directly from data. This is in contrast to other existing methods for modeling instruction videos [20] that assume a script (recipe) is known and fixed in advance. We address the problem by first performing temporal clustering of text followed by clustering in video, where the two clustering tasks are linked by joint constraints. The complementary nature of the two clustering problems helps to resolve ambiguities in the two individual modalities. For example, two video segments with very different appearance but depict-

*WILLOW project-team, Département d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France.
†SIERRA project-team, Département d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548, Paris, France.
‡IIIT Hyderabad

[1] How to : change a car tire, perform CardioPulmonary resuscitation (CPR), jump a car, repot a plant and make coffee

[2] Some instruction videos on YouTube have tens of millions of views, e.g. www.youtube.com/watch?v=J4-GRH2nDvw.

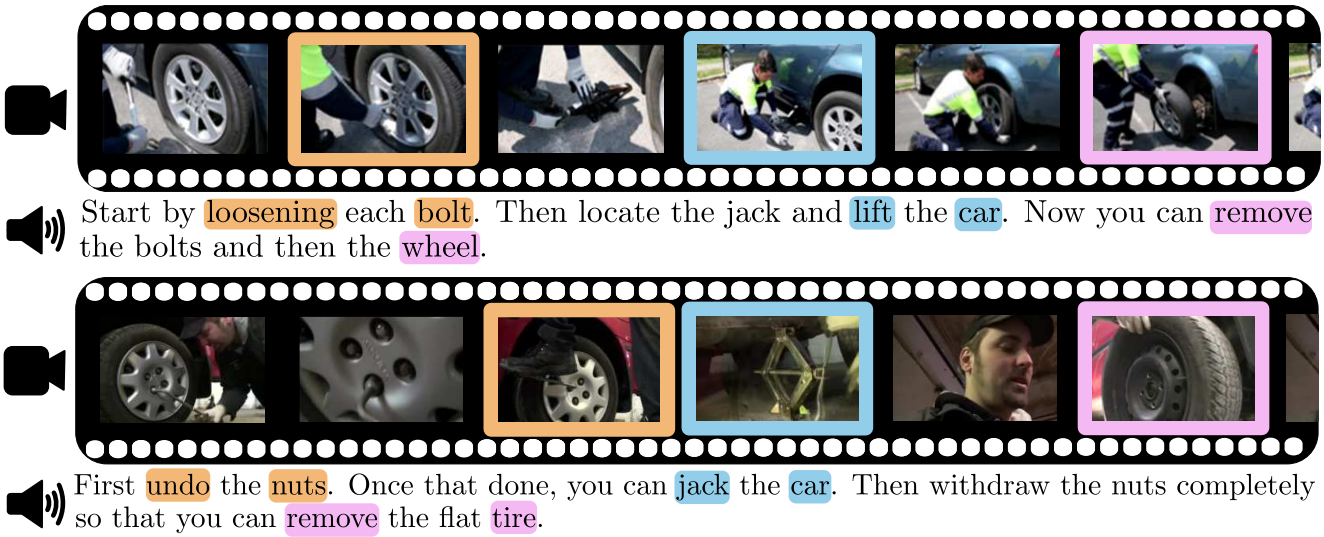


Figure 1: Given a set of narrated instruction videos demonstrating a particular task, we wish to automatically discover the main steps to achieve the task and associate each step with its corresponding narration and appearance in each video. Here frames from two videos demonstrating changing the car tire are shown, together with excerpts of the corresponding narrations. Note the large variations in both the narration and appearance of the different steps highlighted by the same colors in both videos (here only three steps are shown).

ing the same step can be grouped together because they are narrated in a similar language. Conversely, two video segments described with very different expressions, for example, "jack up the car" and "raise the vehicle" can be identified as belonging to the same instruction step because they have similar visual appearance. The output of our method is the script listing the discovered steps of the task as well as the temporal location of each step in the input videos. We validate our method on a new dataset of instruction videos composed of five different tasks with a total of 150 videos and about 800,000 frames.

2. Related work

This work relates to unsupervised and weakly-supervised learning methods in computer vision and natural language processing. Particularly related to ours is the work on learning script-like knowledge from natural language descriptions [7, 12, 27]. These methods aim to discover typical events (steps) and their order for particular scenarios (tasks)[3] such as "cooking scrambled egg", "taking a bus" or "making coffee". While [7] uses large-scale news copora, [27] argues that many events are implicit and are not described in such general-purpose text data. Instead, [12, 27] use event sequence descriptions collected for particular scenarios. Differently to this work, we learn sequences of events from narrated instruction videos on the Internet. Such data contains detailed event descriptions but is not structured and contains more noise compared to the input of [12, 27].

Interpretation of narrated instruction videos has been recently addressed in [20]. While this work analyses cooking videos at a great scale, it relies on readily-available recipes which may not be available for more general scenarios. Differently from [20], we here aim to learn the steps of instruction videos using a discriminative clustering approach. A similar task to ours is addressed in [22] using latent variable structured perceptron algorithm to align nouns in instruction sentences with objects touched by hands in instruction videos. However, similarly to [20], [22] uses laboratory experimental protocols as textual input, whereas here we consider a weaker signal in the form of the real transcribed narration of the video.

In computer vision, unsupervised action recognition has been explored in simple videos [24]. More recently, weakly supervised learning of actions in video using video scripts or event order has been addressed in [4, 5, 6, 10, 17]. Particularly related to ours is the work [5] which explores the known order of events to localize and learn actions in training data. While [5] uses manually annotated sequences of events, we here discover the sequences of main events by clustering transcribed narrations of the videos. Related is also the work of [6] that aligns natural text descriptions to video but in contrast to our approach does not discover automatically the common sequence of main steps. Methods in [23, 26] learn in an unsupervised manner the temporal structure of actions from video but do not discover textual expressions for actions as we do in this work. The recent concurrent work [28] is addressing, independently of our work, a similar problem but with a different approach based on a probabilistic generative model and considering a different set of tasks mainly focussed on cooking activities.

Our work is also related to video summarization and in particular to the recent work on category-specific video

[3] We here assign the same meaning to terms "event" and "step" as well as to terms "script" and "task".

summarization [25, 30]. While summarization is a subjective task, we here aim to extract the key steps required to achieve a concrete task that consistently appear in the same sequence in the input set of videos. In addition, unlike video summarization [25, 30] we jointly exploit visual and linguistic modalities in our approach.

3. New dataset of instruction videos

We have collected a dataset of narrated instruction videos for five tasks: *Making a coffee*, *Changing car tire*, *Performing cardiopulmonary resuscitation (CPR)*, *Jumping a car* and *Repotting a plant*. The videos were obtained by searching YouTube with relevant keywords. The five tasks were chosen so that they have a large number of available videos with English transcripts while trying to cover a wide range of activities that include complex interactions of people with objects and other people. For each task, we took the top 30 videos with English ASR returned by YouTube. We also quickly verified that each video contains a person actually performing the task (as opposed to just talking about it). The result is a total of 150 videos, 30 videos for each task. The average length of our videos is about 4,000 frames (or 2 minutes) and the entire dataset contains about 800,000 frames.

The selected videos have English transcripts obtained from YouTube's automatic speech recognition (ASR) system. To remove the dependence of results on errors of the particular ASR method, we have manually corrected misspellings and punctuations in the output transcriptions. We believe this step will soon become obsolete given rapid improvements of ASR methods. As we do not modify the content of the spoken language in videos, the transcribed verbal instructions still represent an extremely challenging example of natural language with large variability in the used expressions and terminology. Each word of the transcript is associated with a time interval in the video (usually less than 5 seconds) obtained from the closed caption timings.

For the purpose of evaluation, we have manually annotated the temporal location in each video of the main steps necessary to achieve the given task. For all tasks, we have defined the ordered sequence of ground truth steps before running our algorithm. The choice of steps was made by an agreement of 2-3 annotators who have watched the input videos and verified the steps on instruction video websites such as `http://www.howdini.com`. While some steps can be occasionally left out in some videos or the ordering slightly modified, overall we have observed a good consistency in the given sequence of instructions among the input videos. We measured that only 6% of the step annotations did not fit the global order, while a step was missing from the video 28% of the time.[4] We hypothesize that this could be attributed to the fact that all videos are made with the same goal of giving other humans clear, concise and comprehensible verbal and visual instructions on how to achieve the given task. Given the list of steps for each task, we have manually annotated each time interval in each input video to one of the ground truth steps (or no step). The actions of the individual steps are typically separated by hundreds of frames where the narrator transitions between the steps or explains verbally what is going to happen. Furthermore, some steps could be missing in some videos, or could be present but not described in the narration. Finally, the temporal alignment between the narration and the actual actions in video is only coarse as the action is often described before it is performed.

[4] We describe these measurements in more details in the appendix for the paper [2].

4. Modelling narrated instruction videos

We are given a set of N instruction videos all depicting the same task (such as "changing a tire"). The n-th input video is composed of a video stream of T_n segments of frames $(x_t^n)_{t=1}^{T_n}$ and an audio stream containing a detailed verbal description of the depicted task. We suppose that the audio description was transcribed to raw text and then processed to a sequence of S_n text tokens $(d_s^n)_{s=1}^{S_n}$. Given this data, we want to automatically recover the sequence of K main steps that compose the given task and locate each step within each input video and text transcription.

We formulate the problem as two clustering tasks, one in text and one in video, applied one after each other and linked by joint constraints linking the two modalities. This two-stage approach is based on the intuition that the variation in natural language describing each task is easier to capture than the visual variability of the input videos. In the first stage, we cluster the text transcripts into a sequence of K main steps to complete the given task. Empirically, we have found (see results in Sec. 5.1) that it is possible to discover the sequence of the K main steps for each task with high precision. However, the text itself gives only a poor localization of each step in each video. Therefore, in the second stage we accurately localize each step in each video by clustering the input videos using the sequence of K steps extracted from text as constraints on the video clustering. To achieve this, we use two types of constraints between video and text. First, we assume that both the video and the text narration follow the same sequence of steps. This results in a global ordering constraint on the recovered clustering. Second, we assume that people perform the action approximately at the same time that they talk about it. This constraint temporally links the recovered clusters in text and video. The important outcome of the video clustering stage is that the K extracted steps get propagated by visual similarity to videos where the text descriptions are missing or ambiguous.

We first describe the text clustering in Sec. 4.1 and then introduce the video clustering with constraints in Sec. 4.2.

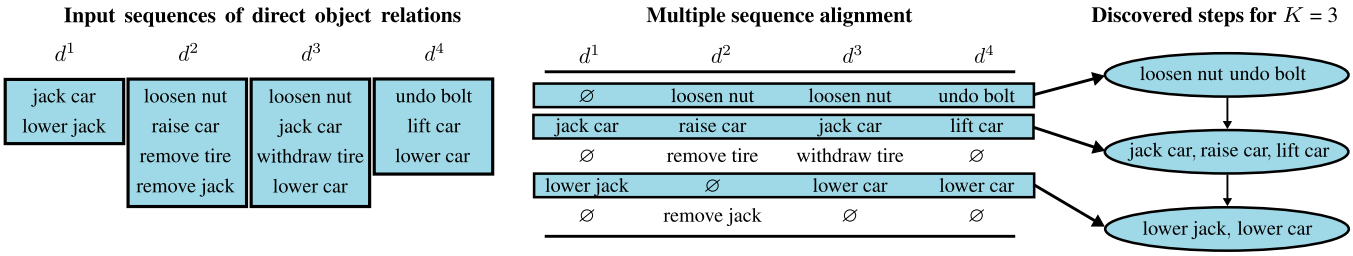


Figure 2: **Clustering transcribed verbal instructions. Left:** The input raw text for each video is converted into a sequence of direct object relations. Here, an illustration of four sequences from four different videos is shown. **Middle:** Multiple sequence alignment is used to align all sequences together. Note that different direct object relations are aligned together as long as they have the same sense, e.g. "loosen nut" and "undo bolt". **Right:** The main instruction steps are extracted as the $K = 3$ most common steps in all the sequences.

4.1. Clustering transcribed verbal instructions

The goal here is to cluster the transcribed verbal descriptions of each video into a sequence of *main steps* necessary to achieve the task. This stage is important as the resulting clusters will be used as constraints for jointly learning and localizing the main steps in video. We assume that the important steps are common to many of the transcripts and that the sequence of steps is (roughly) preserved in all transcripts. Hence, following [27], we formulate the problem of clustering the input transcripts as a multiple sequence alignment problem. However, in contrast to [27] who cluster manually provided descriptions of each step, we wish to cluster transcribed verbal instructions. Hence our main challenge is to deal with the variability in spoken natural language. To overcome this challenge, we take advantage of the fact that completing a certain task usually involves interactions with objects or people and hence we can extract a more structured representation from the input text stream.

More specifically, we represent the textual data as a sequence of *direct object relations*. A direct object relation d is a pair composed of a verb and its direct object complement, such as "remove tire". Such a direct object relation can be extracted from the dependency parser of the input transcribed narration [9]. We denote the set of all different direct object relations extracted from all narrations as $\mathcal{D}$, with cardinality D. For the n-th video, we thus represent the text signal as a sequence of direct object relation tokens: $d^n = (d^n_1, \dots, d^n_{S_n})$, where the length S_n of the sequence varies from one video clip to another. This step is key to the success of our method as it allows us to convert the problem of clustering raw transcribed text into an easier problem of clustering sequences of direct object relations. The goal is now to extract from the narrations the most common sequence of K main steps to achieve the given task. To achieve this, we first find a globally consistent alignment of the direct object relations that compose all text sequences by solving a multiple sequence alignment problem. Second, we pick from this alignment the K most globally consistent clusters across videos.

Multiple sequence alignment model. We formulate the first stage of finding the common alignment between the input sequences of direct object relations as a multiple sequence alignment problem with the *sum-of-pairs score* [32]. In details, a global alignment can be defined by re-mapping each input sequence d^n of tokens to a global common template of L slots, for L large enough. We let $(\phi(d^n))_{1\leq l\leq L}$ represent the (increasing) re-mapping for sequence d^n at the new locations indexed by l: $\phi(d^n)_l$ represents the direct object relation put at location l, with $\phi(d^n)_l = \varnothing$ if a slot is left empty (denoting the insertion of a gap in the original sequence of tokens). See the middle of Figure 2 for an example of re-mapping. The goal is then to find a global alignment that minimizes the following sum-of-pairs cost function:

$$\sum_{(n,m)} \sum_{l=1}^{L} c(\phi(d^n)_l, \phi(d^m)_l), \tag{1}$$

where $c(d_1, d_2)$ denotes the cost of aligning the direct object relations d_1 and d_2 at the same common slot l in the global template. The above cost thus denotes the sum of all pairwise alignments of the individual sequences (the outer sum), where the quality of each alignment is measured by summing the cost c of matches of individual direct object relations mapped into the common template sequence. We use a negative cost when d_1 and d_2 are similar according to the distance in the WordNet tree [11, 21] of their verb and direct object constituents, and positive if they are dissimilar (details are given in Sec. 5). As the verbal narrations can talk about many other things than the main steps of a task, we set $c(d, d') = 0$ if either d or d' is $\varnothing$. An illustration of clustering the transcribed verbal instructions into a sequence of K steps is shown in Figure 2.

Optimization using Frank-Wolfe. Optimizing the cost (1) is NP-hard [32] because of the combinatorial nature of the problem. The standard solution from computational biology is to apply a heuristic algorithm that proceeds by incremental pairwise alignment using dynamic programming [18]. In contrast, we show in the appendix of the paper [2] that the multiple sequence alignment problem given by (1) can be reformulated as an integer quadratic program with combinatorial constraints, for which the Frank-Wolfe optimization algorithm has been used recently with increasing success [5, 14, 15, 16]. Interestingly, we have observed empirically (see the appendix [2]) that the Frank-

Wolfe algorithm was giving better solutions (in terms of objective (1)) than the state-of-the-art heuristic procedures for this task [13, 18]. Our Frank-Wolfe based solvers also offer us greater flexibility in defining the alignment cost and scale better with the length of input sequences and the vocabulary of direct object relations.

Extracting the main steps. After a global alignment is obtained, we sort the global template l by the number of direct object relations aligned to each slot. Given K as input, the top K slots give the main instruction steps for the task, unless there are multiple steps with the same support, which go beyond K. In this case, we pick the next smaller number below K which excludes these ties, allowing the choice of an *adaptive* number of main instruction steps when there is not enough saliency for the last steps. This strategy essentially selects $k \leq K$ salient steps, while refusing to make a choice among steps with equal support that would increase the total number of steps beyond K. As we will see in our results in Sec. 5.1, our algorithm sometimes returns a much smaller number than K for the main instruction steps, giving more robustness to the exact choice of parameter K.

Encoding of the output. We post-process the output of multiple sequence alignment into an assignment matrix $R_n \in \{0,1\}^{S_n \times K}$ for each input video n, where $(R_n)_{sk} = 1$ means that the direct object token d_s^n has been assigned to step k. If a direct object has not been assigned to any step, the corresponding row of the matrix R_n will be zero.

4.2. Discriminative clustering of videos under text constraints

Given the output of the text clustering that identified the important K steps forming a task, we now want to find their temporal location in the video signal. We formalize this problem as looking for an assignment matrix $Z_n \in \{0,1\}^{T_n \times K}$ for each input video n, where $(Z_n)_{tk} = 1$ indicates the visual presence of step k at time interval t in video n, and T_n is the length of video n. Similarly to R_n, we allow the possibility that a whole row of Z_n is zero, indicating that no step is visually present for the corresponding time interval.

We propose to tackle this problem using a discriminative clustering approach with global ordering constraints, as was successfully used in the past for the temporal localization of actions in videos [5], but with additional *weak temporal constraints*. In contrast to [5] where the order of actions was manually given for each video, our multiple sequence alignment approach automatically discovers the main steps. More importantly, we also use the *text caption timing* to provide a fine-grained weak temporal supervision for the visual appearance of steps, which is described next.

Temporal weak supervision from text. From the output of the multiple sequence alignment (encoded in the matrix $R_n \in \{0,1\}^{S_n \times K}$), each direct object token d_s^n has been assigned to one of the possible K steps, or to no step at all. We use the tokens that have been assigned to a step as a constraint on the visual appearance of the same step in the video (using the assumption that people do what they say approximately when they say it). We encode the closed caption timing alignment by a binary matrix $A_n \in \{0,1\}^{S_n \times T_n}$ for each video, where $(A_n)_{st}$ is 1 if the s-th direct object is mentioned in a closed caption that overlaps with the time interval t in video. Note that this alignment is only approximate as people usually do not perform the action exactly at the same time that they talk about it, but instead with a varying delay. Second, the alignment is noisy as people typically perform the action only once, but often talk about it multiple times (e.g. in a summary at the beginning of the video). We address these issues by the following two *weak supervision* constraints. First, we consider a larger set of possible time intervals $[t-\Delta_b, t+\Delta_a]$ in the matrix A rather than the exact time interval t given by the timing of the closed caption. Δ_b and Δ_a are global parameters fixed either qualitatively, or by cross-validation if labeled data is provided. Second, we put as a constraint that the action happens at least once in the set of all possible video time intervals where the action is mentioned in the transcript (rather than every time it is mentioned). These constraints can be encoded as the following linear inequality constraint on Z_n: $A_n Z_n \geq R_n$ (see the appendix [2] for the detailed derivation).

Ordering constraint. In addition, we also enforce that the temporal order of the steps appearing visually is consistent with the discovered script from the text, encoding our assumption that there is a common ordered script for the task across videos. We encode these sequence constraints on Z_n in a similar manner to [6], which was shown to work better than the encoding used in [5]. In particular, we only predict the *most salient* time interval in the video that describes a given step. This means that a particular step is assigned to *exactly one* time interval in each video. We denote by $\mathcal{Z}_n$ this sequence ordering constraint set.

Discriminative clustering. The main motivation behind discriminative clustering is to find a *clustering* of the data that can be easily recovered by a *linear classifier* through the minimization of an appropriate *cost function* over the assignment matrix Z_n. The approach introduced in [3] allows to easily add prior information on the expected clustering. Such priors have been recently introduced in the context of aligning video and text [5, 6] in the form of ordering constraints over the latent label variables. Here we use a similar approach to cluster the N input video streams (x_t) into a sequence of K steps, as follows. We represent each time interval by a d-dimensional feature vector. The feature vectors for the n-th video are stacked in a $T_n \times d$ design matrix denoted by X_n. We denote by X the $T \times d$ matrix obtained by the concatenation of all X_n matrices (and similarly, by Z, R and A the appropriate concatenation of the Z_n, R_n and A_n matrices over n). In order to obtain the temporal localization into K steps, we learn a linear classifier represented by a $d \times K$ matrix denoted by W. This model is shared among all videos.

Changing a tire		Performing CPR		Repot a plant		Make coffee		Jump car	
GT (**11**)	$K \leq 10$	GT (**7**)	$K \leq 10$	GT (**7**)	$K \leq 10$	GT (**10**)	$K \leq 10$	GT (**12**)	$K \leq 10$
get tools out	**get tire**	*open airway*	**open airway**	*take plant*	**remove plant**	*add coffee*	**put coffee**	*connect red A*	**connect cable**
start loose	**loosen nut**	*check pulse*	**put hand**	*put soil*	**use soil**		**fill chamber**		**charge battery**
	put jack		**tilt head**	*loosen roots*	**loosen soil**	*fill water*	**fill water**	*connect red B*	**connect end**
jack car	**jack car**		**lift chin**	*place plant*	**place plant**	*screw filter*	**put filter**	*start car A*	**start car**
unscrew wheel	**remove nut**	*give breath*	**give breath**	*add top*	**add soil**		**see steam**	*remove cable A*	**remove cable**
remove wheel	**take wheel**	*do compressions*	**do compression**	*water plant*	**water plant**	*put stove*	**take minutes**	*remove cable B*	**disconnect cable**
put wheel	**take tire**		**open airway**				**make coffee**		
screw wheel	**put nut**		**start compression**			*see coffee*	**see coffee**		
lower car	**lower jack**		**do compression**			*pour coffee*	**make cup**		
tight wheel	**tighten nut**		**give breath**						
Precision	0.9	Precision	0.4	Precision	1	Precision	0.67	Precision	0.83
Recall	0.9	Recall	0.57	Recall	0.86	Recall	0.6	Recall	0.42

Table 1: Automatically recovered sequences of steps for the five tasks. Each recovered step is represented by one of the aligned direct object relations (shown in bold). Note that most of the recovered steps correspond well to the ground truth steps (shown in italic). The results are shown for the maximum number of discovered steps K set to 10. Note how our method automatically selects less than 10 steps in some cases. These are the automatically chosen $k \leq K$ steps that are the most salient in the aligned narrations as described in Sec. 4.1. For *CPR*, our method recovers fine-grained steps e.g. *tilt head*, *lift chin*, which are not included in the main ground truth steps, but nevertheless could be helpful in some situations, as well as repetitions that were not annotated but were indeed present.

The target assignment $\hat{Z}$ is found by minimizing the clustering cost function h under both the consistent script ordering constraints $\mathcal{Z}$ and our weak supervision constraints:

$$\underset{Z}{\text{minimize}} \quad h(Z) \quad \text{s.t.} \quad \underbrace{Z \in \mathcal{Z}}_{\text{ordered script}}, \quad \underbrace{AZ \geq R}_{\substack{\text{weak textual} \\ \text{constraints}}}. \qquad (2)$$

The clustering cost $h(Z)$ is given as in DIFFRAC [3] as:

$$h(Z) = \min_{W \in \mathbb{R}^{K \times d}} \underbrace{\frac{1}{2T}\|Z - XW\|_F^2}_{\text{Discriminative loss on data}} + \underbrace{\frac{\lambda}{2}\|W\|_F^2}_{\text{Regularizer}}. \qquad (3)$$

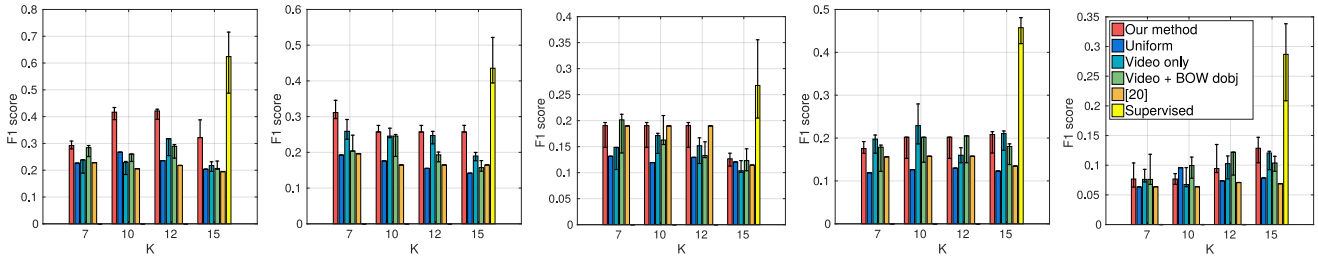
The first term in (3) is the discriminative loss on the data that measures how easy the input data X is separable by the linear classifier W when the target classes are given by the assignments Z. For the squared loss considered in eq. (3), the optimal weights W^* minimizing (3) can be found in closed form, which significantly simplifies the computation. However, to solve (2), we need to optimize over assignment matrices Z that encode sequences of events and incorporate constraints given by clusters obtained from transcribed textual narrations (Sec. 4.1). This is again done by using the Frank-Wolfe algorithm, which allows the use of *efficient dynamic programs* to handle the combinatorial constraints on Z. More details are given in the appendix [2].

5. Experimental evaluation

In this section, we first describe the details of the text and video features. Then we present the results divided into two experiments: (i) in Sec. 5.1, we evaluate the quality of steps extracted from video narrations, and (ii) in Sec. 5.2, we evaluate the temporal localization of the recovered steps in video using constraints derived from text. All the data and code are available at our project webpage [1].

Video and text features. We represent the transcribed narrations as sequences of direct object relations. For this purpose, we run a dependency parser [9] on each transcript. We lemmatize all direct object relations and keep the ones for which the direct object corresponds to nouns. To represent a video, we use motion descriptors in order to capture actions (loosening, jacking-up, giving compressions) and frame appearance descriptors to capture the depicted objects (tire, jack, car). We split each video into 10-frame time intervals and represent each interval by its motion and appearance descriptors aggregated over a longer block of 30 frames. The motion representation is a histogram of local optical flow (HOF) descriptors aggregated into a single bag-of-visual-word vector of 2,000 dimensions [31]. The visual vocabulary is generated by k-means on a separate large set of training descriptors. To capture the depicted objects in the video, we apply the VGG-verydeep-16 CNN [29] over each frame in a sliding window manner over multiple scales. This can be done efficiently in a fully convolutional manner. The resulting 512-dimensional feature maps of conv5 responses are then aggregated into a single bag-of-visual-word vector of 1,000 dimensions, which aims to capture the presence/absence of different objects within each video block. A similar representation (aggregated into compact VLAD descriptor) was shown to work well recently for a variety of recognition tasks [8]. The bag-of-visual-word vectors representing the motion and the appearance are normalized using the Hellinger normalization and then concatenated into a single 3,000 dimensional vector representing each time interval.

WordNet distance. For the multiple sequence alignment presented in Sec. 4.1, we set $c(d_1, d_2) = -1$ if d_1 and d_2 have both their verbs and direct objects that match exactly in the Wordnet tree (distance equal to 0). Otherwise we set $c(d_1, d_2)$ to be 100. This is to ensure a high precision for the resulting alignment.



(a) Change tire (**11**) (b) Perform CPR (**7**) (c) Repot plant (**7**) (d) Make coffee (**10**) (e) Jump car (**12**)

Figure 3: Results for temporally localizing recovered steps in the input videos. We give in **bold** the number of ground truth steps.

5.1. Results of step discovery from text narrations

Results of discovering the main steps for each task from text narrations are presented in Table 1. We report results of the multiple sequence alignment described in Sec. 4.1 when the maximum number of recoverable steps is $K = 10$. Additional results for different choices of K are given in the appendix [2]. With increasing K, we tend to recover more complete sequences at the cost of occasional repetitions, e.g. *position jack* and *jack car* that refer to the same step. To quantify the performance, we measure precision as the proportion of correctly recovered steps appearing in the correct order. We also measure recall as the proportion of the recovered ground truth steps. The values of precision and recall are given at the bottom of Table 1.

5.2. Results of localizing instruction steps in video

In the previous section, we have evaluated the quality of the sequences of steps recovered from the transcribed narrations. In this section, we evaluate how well we localize the individual instruction steps in the video by running our two-stage approach from Sec. 4.

Evaluation metric. To evaluate the temporal localization, we need to have a one-to-one mapping between the discovered steps in the videos and the ground truth steps. Following [19], we look for a one-to-one global matching (shared across all videos of a given task) that maximizes the evaluation score for a given method (using the Hungarian algorithm). Note that this mapping is used only for evaluation, the algorithm does not have access to the ground truth annotations for learning.

The goal is to evaluate whether each ground truth step has been correctly localized in all instruction videos. We thus use the *F1 score* that combines precision and recall into a single score as our evaluation measure. For a given video and a given recovered step, our video clustering method predicts exactly one video time interval t. This detection is considered correct if the time interval falls inside any of the corresponding ground truth intervals, and incorrect otherwise (resulting in a false positive for this video). We compute the recall across all steps and videos, defined as the ratio of the number of correct predictions over the total number of possible ground truth steps across videos. A recall of 1 indicates that every ground truth step has been correctly detected across all videos. The recall decreases towards 0 when we miss some ground truth steps (missed detections). This happens either because this step was not recovered globally, or because it was detected in the video at an incorrect location. This is because the algorithm predicts exactly one occurrence of each step in each video. Similarly, precision measures the proportion of correct predictions among all $N \cdot K_{\text{pred}}$ possible predictions, where N is the number of videos and K_{pred} is the number of main steps used by the method. The F1 score is the harmonic mean of precision and recall, giving a score that ranges between 0 and 1, with the perfect score of 1 when all the steps are predicted at their correct locations in all videos.

Hyperparameters. We set the values of parameters Δ_b and Δ_a to 0 and 10 seconds. The setting is the same for all five tasks. This models the fact that typically each step is first described verbally and then performed on the camera. We set $\lambda = 1/(NK_{\text{pred}})$ for all methods that use (3).

Baselines. We compare results to four baselines. To demonstrate the difficulty of our dataset, we first evaluate a "Uniform" baseline, which simply distributes instructions steps uniformly over the entire instruction video. The second baseline "Video only" [5] does not use the narration and performs only discriminative clustering on visual features with a global order constraint.[5] The third baseline "Video + BOW dobj" basically adds text-based features to the "Video only" baseline (by concatenating the text and video features in the discriminative clustering approach). Here the goal is to evaluate the benefits of our two-stage clustering approach, in contrast to this single-stage clustering baseline. The text features are bag-of-words histograms over a fixed vocabulary of direct object relations.[6] The fourth baseline is our own implementation of the alignment method of [20] (without the supervised vision refinement procedure that requires a set of pre-trained visual classifiers that are not available a-priori in our case). We use [20] to re-align the speech transcripts to the sequence of steps discovered by our method of Sec. 4.1 (as a proxy for the recipe assumed

[5]We use here the improved model from [6] which does not require a "background class" and yields a stronger baseline equivalent to our model (2) *without* the weak textual constraints.

[6]Alternative features of bag-of-words histograms treating separately nouns and verbs also give similar results.

to be known in [20]).[7] To assess the difficulty of the task and dataset, we also compare results with a "Supervised" approach. The classifiers W for the visual steps are trained by running the discriminative clustering of Sec. 4.2 with only ground truth annotations as constraints on the training set. At test time, these classifiers are used to make predictions under the global ordering constraint on unseen videos. We report results using 5-fold cross validation for the supervised approach, with the variation across folds giving the error bars. For the unsupervised discriminative clustering methods, the error bars represent the variation of performance obtained from different rounded solutions collected during the Frank-Wolfe optimization.

Results. Results for localizing the discovered instruction steps are shown in Figure 3. In order to perform a fair comparison to the baseline methods that require a known number of steps K, we report results for a range of K values. Note that in our case the actual number of automatically recovered steps can be (and often is) smaller than K. For *Change tire* and *Perform CPR*, our method consistently outperforms all baselines for all values of K demonstrating the benefits of our approach. For *Repot*, our method is comparable to text-based baselines, underlying the importance of the text signal for this problem. For *Jump car*, our method delivers the best result (for $K = 15$) but struggles for lower values of K, which we found was due to visually similar repeating steps (e.g. start car A and start car B) which are mixed-up for lower values of K. For the *Make coffee* task, the video only baseline is comparable to our method, which by inspecting the output could be attributed to large variability of narrations for this task. Qualitative results of the recovered steps are illustrated in Figure 4.

6. Conclusion and future work

We have described a method to automatically discover the main steps of a task from a set of narrated instruction videos in an unsupervised manner. The proposed approach has been tested on a new annotated dataset of challenging real-world instruction videos containing complex person-object interactions in a variety of indoor and outdoor scenes. Our work opens up the possibility for large scale learning from instruction videos on the Internet. Our model currently assumes the existence of a common script with a fixed ordering of the main steps. While this assumption is often true, e.g. one cannot remove the wheel before jacking up the car, or make coffee before filling the water, some tasks can be performed while swapping (or even leaving out) some of the steps. Recovering more complex temporal structures is an interesting direction for future work.

Acknowledgments This research was supported in part by a Google Research Award, and the ERC grants VideoWorld (no. 267907), Activia (no. 307574) and LEAP (no. 336845).

[7] Note that our method finds at the same time the sequence of steps (a recipe in [20]) and the alignment of the transcripts.

Figure 4: **Examples of three recovered instruction steps for each of the five tasks in our dataset.** For each step, we first show clustered direct object relations, followed by representative example frames localizing the step in the videos. Correct localizations are shown in green. Some steps are incorrectly localized in some videos (red), but often look visually very similar. **See the appendix [2] for additional results.**

References

[1] Project webpage (code/dataset). http://www.di.ens.fr/willow/research/instructionvideos/. 6

[2] Supplementary material (appendix) for the paper. http://arxiv.org/abs/1506.09215. 3, 4, 5, 6, 7, 8

[3] F. Bach and Z. Harchaoui. DIFFRAC: A discriminative and flexible framework for clustering. In *NIPS*, 2007. 5, 6

[4] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *ICCV*, 2013. 2

[5] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 2, 4, 5, 7

[6] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015. 2, 5, 7

[7] N. Chambers and D. Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, 2008. 2

[8] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *CVPR*, 2015. 6

[9] M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC*, 2006. 4, 6

[10] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009. 2

[11] C. Fellbaum. Wordnet: An electronic lexical database. *Cambridge, MA: MIT Press.*, 1998. 4

[12] L. Frermann, I. Titov, and M. Pinkal. A hierarchical Bayesian model for unsupervised induction of script knowledge. In *EACL*, 2014. 2

[13] D. G. Higgins and P. M. Sharp. Clustal: A package for performing multiple sequence alignment on a microcomputer. *Gene*, 1988. 5

[14] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013. 4

[15] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *ECCV*, 2014. 4

[16] S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *NIPS*, 2015. 4

[17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[18] C. Lee, C. Grasso, and M. Sharlow. Multiple sequence alignment using partial order graphs. *Bioinformatics*, 2002. 4, 5

[19] T. Liao. Clustering of time series data, a survey. *Pattern recognition*, 2014. 7

[20] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy. What's cookin'? Interpreting cooking videos using text, speech and vision. In *NAACL*, 2015. 1, 2, 7, 8

[21] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 1995. 4

[22] I. Naim, Y. Chol Song, Q. Liu, L. Huang, H. Kautz, J. Luo, and D. Gildea. Discriminative unsupervised alignment of natural language instructions with corresponding video segments. In *NAACL*, 2015. 2

[23] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 2

[24] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 2008. 2

[25] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014. 3

[26] M. Raptis and L. Sigal. Poselet key-framing: A model for human activity recognition. In *CVPR*, 2013. 2

[27] M. Regneri, A. Koller, and M. Pinkal. Learning script knowledge with Web experiments. In *ACL*, 2010. 1, 2, 4

[28] O. Sener, A. Zamir, S. Savarese, and A. Saxena. Unsupervised semantic parsing of video collections. In *ICCV*, 2015. 2

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

[30] M. Sun, A. Farhadi, and S. Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014. 3

[31] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 6

[32] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of computational biology*, 1(4):337–348, 1994. 4