Identifying Visible Actions in Lifestyle Vlogs

Abstract

We consider the task of identifying human actions visible in online videos. We focus on the widely spread genre of lifestyle vlogs, which consist of videos of people performing actions while verbally describing them. Our goal is to identify if actions mentioned in the speech description of a video are visually present. We construct a dataset with crowdsourced manual annotations of visible actions, and introduce a multimodal algorithm that leverages information derived from visual and linguistic clues to automatically infer which actions are visible in a video. We demonstrate that our multimodal algorithm outperforms algorithms based only on one modality at a time.

Introduction

There has been a surge of recent interest in detecting human actions in videos. Work in this space has mainly focused on learning actions from articulated human pose BIBREF7 , BIBREF8 , BIBREF9 or mining spatial and temporal information from videos BIBREF10 , BIBREF11 . A number of resources have been produced, including Action Bank BIBREF12 , NTU RGB+D BIBREF13 , SBU Kinect Interaction BIBREF14 , and PKU-MMD BIBREF15 .

Most research on video action detection has gathered video information for a set of pre-defined actions BIBREF2 , BIBREF16 , BIBREF1 , an approach known as explicit data gathering BIBREF0 . For instance, given an action such as “open door,” a system would identify videos that include a visual depiction of this action. While this approach is able to detect a specific set of actions, whose choice may be guided by downstream applications, it achieves high precision at the cost of low recall. In many cases, the set of predefined actions is small (e.g., 203 activity classes in BIBREF2), and for some actions, the number of

visual depictions is very small.

An alternative approach is to start with a set of videos, and identify all the actions present in these videos BIBREF17 , BIBREF18 . This approach has been referred to as implicit data gathering, and it typically leads to the identification of a larger number of actions, possibly with a small number of examples per action.

In this paper, we use an implicit data gathering approach to label human activities in videos. To the best of our knowledge, we are the first to explore video action recognition using both transcribed audio and video information. We focus on the popular genre of lifestyle vlogs, which consist of videos of people demonstrating routine actions while verbally describing them. We use these videos to develop methods to identify if actions are visually present.

The paper makes three main contributions. First, we introduce a novel dataset consisting of 1,268 short video clips paired with sets of actions mentioned in the video transcripts, as well as manual annotations of whether the actions are visible or not. The dataset includes a total of 14,769 actions, 4,340 of which are visible. Second, we propose a set of strong baselines to determine whether an action is visible or not. Third, we introduce a multimodal neural architecture that combines information drawn from visual and linguistic clues, and show that it improves over models that rely on one modality at a time.

By making progress towards automatic action recognition, in addition to contributing to video understanding, this work has a number of important and exciting applications, including sports analytics BIBREF19 , human-computer interaction BIBREF20 , and automatic analysis of surveillance video footage BIBREF21 .

The paper is organized as follows. We begin by discussing related work, then describe our data collection

and annotation process. We next overview our experimental set-up and introduce a multimodal method for identifying visible actions in videos. Finally, we discuss our results and conclude with general directions for future work.

Related Work

There has been substantial work on action recognition in the computer vision community, focusing on creating datasets BIBREF22 , BIBREF23 , BIBREF5 , BIBREF2 or introducing new methods BIBREF24 , BIBREF25 , BIBREF26 , BIBREF27 . Table TABREF1 compares our dataset with previous action recognition datasets.

The largest datasets that have been compiled to date are based on YouTube videos BIBREF2 , BIBREF16 , BIBREF1 . These actions cover a broad range of classes including human-object interactions such as cooking BIBREF28 , BIBREF29 , BIBREF6 and playing tennis BIBREF23 , as well as human-human interactions such as shaking hands and hugging BIBREF4 .

Similar to our work, some of these previous datasets have considered everyday routine actions BIBREF2 , BIBREF16 , BIBREF1 . However, because these datasets rely on videos uploaded on YouTube, it has been observed they can be potentially biased towards unusual situations BIBREF1 . For example, searching for videos with the query “drinking tea" results mainly in unusual videos such as dogs or birds drinking tea. This bias can be addressed by paying people to act out everyday scenarios BIBREF5 , but this can end up being very expensive. In our work, we address this bias by changing the approach used to search for videos. Instead of searching for actions in an explicit way, using queries such as “opening a fridge” or “making the bed,” we search for more general videos using queries such as “my morning routine.” This approach has been referred to as implicit (as opposed to explicit) data gathering, and was shown to result in a greater number of videos with more realistic action depictions BIBREF0 .

Although we use implicit data gathering as proposed in the past, unlike BIBREF0 and other human action recognition datasets, we search for routine videos that contain rich audio descriptions of the actions being performed, and we use this transcribed audio to extract actions. In these lifestyle vlogs, a vlogger typically performs an action while also describing it in detail. To the best of our knowledge, we are the first to build a video action recognition dataset using both transcribed audio and video information.

Another important difference between our methodology and previously proposed methods is that we extract action labels from the transcripts. By gathering data before annotating the actions, our action labels are post-defined (as in BIBREF0). This is unlike the majority of the existing human action datasets that use pre-defined labels BIBREF5 , BIBREF2 , BIBREF16 , BIBREF1 , BIBREF4 , BIBREF29 , BIBREF6 , BIBREF3 . Post-defined labels allow us to use a larger set of labels, expanding on the simplified label set used in earlier datasets. These action labels are more inline with everyday scenarios, where people often use different names for the same action. For example, when interacting with a robot, a user could refer to an action in a variety of ways; our dataset includes the actions “stick it into the freezer,” “freeze it,” “pop into the freezer,” and “put into the freezer,” variations, which would not be included in current human action recognition datasets.

In addition to human action recognition, our work relates to other multimodal tasks such as visual question answering BIBREF30 , BIBREF31 , video summarization BIBREF32 , BIBREF33 , and mapping text descriptions to video content BIBREF34 , BIBREF35 . Specifically, we use an architecture similar to BIBREF30 , where an LSTM BIBREF36 is used together with frame-level visual features such as Inception BIBREF37 , and sequence-level features such as C3D BIBREF27 . However, unlike BIBREF30 who encode the textual information (question-answers pairs) using an LSTM, we chose instead to encode our textual information (action descriptions and their contexts) using a large-scale language model ELMo BIBREF38 .

Similar to previous research on multimodal methods BIBREF39 , BIBREF40 , BIBREF41 , BIBREF30 , we also perform feature ablation to determine the role played by each modality in solving the task. Consistent with earlier work, we observe that the textual modality leads to the highest performance across individual modalities, and that the multimodal model combining textual and visual clues has the best overall performance.

Data Collection and Annotation

We collect a dataset of routine and do-it-yourself (DIY) videos from YouTube, consisting of people performing daily activities, such as making breakfast or cleaning the house. These videos also typically include a detailed verbal description of the actions being depicted. We choose to focus on these lifestyle vlogs because they are very popular, with tens of millions having been uploaded on YouTube; tab:nbresultssearchqueries shows the approximate number of videos available for several routine queries. Vlogs also capture a wide range of everyday activities; on average, we find thirty different visible human actions in five minutes of video.

By collecting routine videos, instead of searching explicitly for actions, we do implicit data gathering, a form of data collection introduced by BIBREF0 . Because everyday actions are common and not unusual, searching for them directly does not return many results. In contrast, by collecting routine videos, we find many everyday activities present in these videos.

Data Gathering

We build a data gathering pipeline (see Figure FIGREF5) to automatically extract and filter videos and their transcripts from YouTube. The input to the pipeline is manually selected YouTube channels. Ten channels are chosen for their rich routine videos, where the actor(s) describe their actions in great detail.

From each channel, we manually select two different playlists, and from each playlist, we randomly download ten videos.

The following data processing steps are applied:

Transcript Filtering. Transcripts are automatically generated by YouTube. We filter out videos that do not contain any transcripts or that contain transcripts with an average (over the entire video) of less than 0.5 words per second. These videos do not contain detailed action descriptions so we cannot effectively leverage textual information.

Extract Candidate Actions from Transcript. Starting with the transcript, we generate a noisy list of potential actions. This is done using the Stanford parser BIBREF42 to split the transcript into sentences and identify verb phrases, augmented by a set of hand-crafted rules to eliminate some parsing errors. The resulting actions are noisy, containing phrases such as “found it helpful if you” and “created before up the top you.”

Segment Videos into Miniclips. The length of our collected videos varies from two minutes to twenty minutes. To ease the annotation process, we split each video into miniclips (short video sequences of maximum one minute). Miniclips are split to minimize the chance that the same action is shown across multiple miniclips. This is done automatically, based on the transcript timestamp of each action. Because YouTube transcripts have timing information, we are able to line up each action with its corresponding frames in the video. We sometimes notice a gap of several seconds between the time an action occurs in the transcript and the time it is shown in the video. To address this misalignment, we first map the actions to the miniclips using the time information from the transcript. We then expand the miniclip by 15 seconds before the first action and 15 seconds after the last action. This increases the chance that all actions will be captured in the miniclip.

Motion Filtering. We remove miniclips that do not contain much movement. We sample one out of every one hundred frames of the miniclip, and compute the 2D correlation coefficient between these sampled frames. If the median of the obtained values is greater than a certain threshold (we choose 0.8), we filter out the miniclip. Videos with low movement tend to show people sitting in front of the camera, describing their routine, but not acting out what they are saying. There can be many actions in the transcript, but if they are not depicted in the video, we cannot leverage the video information.

Visual Action Annotation

Our goal is to identify which of the actions extracted from the transcripts are visually depicted in the videos. We create an annotation task on Amazon Mechanical Turk (AMT) to identify actions that are visible.

We give each AMT turker a HIT consisting of five miniclips with up to seven actions generated from each miniclip. The turker is asked to assign a label (visible in the video; not visible in the video; not an action) to each action. Because it is difficult to reliably separate not visible and not an action, we group these labels together.

Each miniclip is annotated by three different turkers. For the final annotation, we use the label assigned by the majority of turkers, i.e., visible or not visible / not an action.

To help detect spam, we identify and reject the turkers that assign the same label for every action in all five miniclips that they annotate. Additionally, each HIT contains a ground truth miniclip that has been pre-labeled by two reliable annotators. Each ground truth miniclip has more than four actions with labels that were agreed upon by both reliable annotators. We compute accuracy between a turker's answers and the ground truth annotations; if this accuracy is less than 20%, we reject the HIT as spam.

After spam removal, we compute the agreement score between the turkers using Fleiss kappa BIBREF43 . Over the entire data set, the Fleiss agreement score is 0.35, indicating fair agreement. On the ground truth data, the Fleiss kappa score is 0.46, indicating moderate agreement. This fair to moderate agreement indicates that the task is difficult, and there are cases where the visibility of the actions is hard to label. To illustrate, Figure FIGREF9 shows examples where the annotators had low agreement.

Table TABREF8 shows statistics for our final dataset of videos labeled with actions, and Figure 2 shows a sample video and transcript, with annotations.

For our experiments, we use the first eight YouTube channels from our dataset as train data, the ninth channel as validation data and the last channel as test data. Statistics for this split are shown in Table TABREF10 .

Discussion

The goal of our dataset is to capture naturally-occurring, routine actions. Because the same action can be identified in different ways (e.g., “pop into the freezer”, “stick into the freezer"), our dataset has a complex and diverse set of action labels. These labels demonstrate the language used by humans in everyday scenarios; because of that, we choose not to group our labels into a pre-defined set of actions. Table TABREF1 shows the number of unique verbs, which can be considered a lower bound for the number of unique actions in our dataset. On average, a single verb is used in seven action labels, demonstrating the richness of our dataset.

The action labels extracted from the transcript are highly dependent on the performance of the constituency parser. This can introduce noise or ill-defined action labels. Some acions contain extra words (e.g., “brush my teeth of course”), or lack words (e.g., “let me just”). Some of this noise is handled

during the annotation process; for example, most actions that lack words are labeled as “not visible” or “not an action” because they are hard to interpret.

Identifying Visible Actions in Videos

Our goal is to determine if actions mentioned in the transcript of a video are visually represented in the video. We develop a multimodal model that leverages both visual and textual information, and we compare its performance with several single-modality baselines.

Data Processing and Representations

Starting with our annotated dataset, which includes miniclips paired with transcripts and candidate actions drawn from the transcript, we extract several layers of information, which we then use to develop our multimodal model, as well as several baselines.

Action Embeddings. To encode each action, we use both GloVe BIBREF44 and ELMo BIBREF38 embeddings. When using GloVe embeddings, we represent the action as the average of all its individual word embeddings. We use embeddings with dimension 50. When using ELMo, we represent the action as a list of words which we feed into the default ELMo embedding layer. This performs a fixed mean pooling of all the contextualized word representations in each action.

Part-of-speech (POS). We use POS information for each action. Similar to word embeddings BIBREF44 , we train POS embeddings. We run the Stanford POS Tagger BIBREF45 on the transcripts and assign a POS to each word in an action. To obtain the POS embeddings, we train GloVe on the Google N-gram corpus using POS information from the five-grams. Finally, for each action, we average together the POS embeddings for all the words in the action to form a POS embedding vector.

Context Embeddings. Context can be helpful to determine if an action is visible or not. We use two types of context information, action-level and sentence-level. Action-level context takes into account the previous action and the next action; we denote it as Context INLINEFORM0 . These are each calculated by taking the average of the action's GloVe embeddings. Sentence-level context considers up to five words directly before the action and up to five words after the action (we do not consider words that are not in the same sentence as the action); we denote it as Context INLINEFORM1 . Again, we average the GLoVe embeddings of the preceding and following words to get two context vectors.

Concreteness. Our hypothesis is that the concreteness of the words in an action is related to its visibility in a video. We use a dataset of words with associated concreteness scores from BIBREF46 . Each word is labeled by a human annotator with a value between 1 (very abstract) and 5 (very concrete). The percentage of actions from our dataset that have at least one word in the concreteness dataset is 99.8%. For each action, we use the concreteness scores of the verbs and nouns in the action. We consider the concreteness score of an action to be the highest concreteness score of its corresponding verbs and nouns. tab:concr1 shows several sample actions along with their concreteness scores and their visiblity.

Video Representations. We use Yolo9000 BIBREF47 to identify objects present in each miniclip. We choose YOLO9000 for its high and diverse number of labels (9,000 unique labels). We sample the miniclips at a rate of 1 frame-per-second, and we use the Yolo9000 model pre-trained on COCO BIBREF48 and ImageNet BIBREF49 .

We represent a video both at the frame level and the sequence level. For frame-level video features, we use the Inception V3 model BIBREF37 pre-trained on ImageNet. We extract the output of the very last layer before the Flatten operation (the “bottleneck layer"); we choose this layer because the following fully connected layers are too specialized for the original task they were trained for. We extract Inception V3 features from miniclips sampled at 1 frame-per-second.

For sequence-level video features, we use the C3D model BIBREF27 pre-trained on the Sports-1M dataset BIBREF23 . Similarly, we take the feature map of the sixth fully connected layer. Because C3D captures motion information, it is important that it is applied on consecutive frames. We take each frame used to extract the Inception features and extract C3D features from the 16 consecutive frames around it.

We use this approach because combining Inception V3 and C3D features has been shown to work well in other video-based models BIBREF30 , BIBREF25 , BIBREF1 .

Baselines

Using the different data representations described in Section SECREF12 , we implement several baselines.

Concreteness. We label as visible all the actions that have a concreteness score above a certain threshold, and label as non-visible the remaining ones. We fine tune the threshold on our validation set; for fine tuning, we consider threshold values between 3 and 5. Table TABREF20 shows the results obtained for this baseline.

Feature-based Classifier. For our second set of baselines, we run a classifier on subsets of all of our features. We use an SVM BIBREF50 , and perform five-fold cross-validation across the train and validation sets, fine tuning the hyper-parameters (kernel type, C, gamma) using a grid search. We run experiments with various combinations of features: action GloVe embeddings; POS embeddings; embeddings of sentence-level context (Context INLINEFORM0) and action-level context (Context INLINEFORM1); concreteness score. The combinations that perform best during cross-validation on the combined train and validation sets are shown in Table TABREF20 .

LSTM and ELMo. We also consider an LSTM model BIBREF36 that takes as input the tokenized action sequences padded to the length of the longest action. These are passed through a trainable embedding layer, initialized with GloVe embeddings, before the LSTM. The LSTM output is then passed through a feed forward network of fully connected layers, each followed by a dropout layer BIBREF51 at a rate of 50%. We use a sigmoid activation function after the last hidden layer to get an output probability distribution. We fine tune the model on the validation set for the number of training epochs, batch size, size of LSTM, and number of fully-connected layers.

We build a similar model that embeds actions using ELMo (composed of 2 bi-LSTMs). We pass these embeddings through the same feed forward network and sigmoid activation function. The results for both the LSTM and ELMo models are shown in Table TABREF20 .

Yolo Object Detection. Our final baseline leverages video information from the YOLO9000 object detector. This baseline builds on the intuition that many visible actions involve visible objects. We thus label an action as visible if it contains at least one noun similar to objects detected in its corresponding miniclip. To measure similarity, we compute both the Wu-Palmer (WUP) path-length-based semantic similarity BIBREF52 and the cosine similarity on the GloVe word embeddings. For every action in a miniclip, each noun is compared to all detected objects and assigned a similarity score. As in our concreteness baseline, the action is assigned the highest score of its corresponding nouns. We use the validation data to fine tune the similarity threshold that decides if an action is visible or not. The results are reported in Table TABREF20 . Examples of actions that contain one or more words similar to detected objects by Yolo can be seen in Figure FIGREF18 .

Multimodal Model

Each of our baselines considers only a single modality, either text or video. While each of these

modalities contributes important information, neither of them provides a full picture. The visual modality is inherently necessary, because it shows the visibility of an action. For example, the same spoken action can be labeled as either visible or non-visible, depending on its visual context; we find 162 unique actions that are labeled as both visible and not visible, depending on the miniclip. This ambiguity has to be captured using video information. However, the textual modality provides important clues that are often missing in the video. The words of the person talking fill in details that many times cannot be inferred from the video. For our full model, we combine both textual and visual information to leverage both modalities.

We propose a multimodal neural architecture that combines encoders for the video and text modalities, as well as additional information (e.g., concreteness). Figure FIGREF19 shows our model architecture. The model takes as input a (miniclip INLINEFORM0 , action INLINEFORM1) pair and outputs the probability that action INLINEFORM2 is visible in miniclip INLINEFORM3 . We use C3D and Inception V3 video features extracted for each frame, as described in Section SECREF12 . These features are concatenated and run through an LSTM.

To represent the actions, we use ELMo embeddings (see Section SECREF12). These features are concatenated with the output from the video encoding LSTM, and run through a three-layer feed forward network with dropout. Finally, the result of the last layer is passed through a sigmoid function, which produces a probability distribution indicating whether the action is visible in the miniclip. We use an RMSprop optimizer BIBREF53 and fine tune the number of epochs, batch size and size of the LSTM and fully-connected layers.

Evaluation and Results

Table TABREF20 shows the results obtained using the multimodal model for different sets of input features. The model that uses all the input features available leads to the best results, improving

significantly over the text-only and video-only methods.

We find that using only Yolo to find visible objects does not provide sufficient information to solve this task. This is due to both the low number of objects that Yolo is able to detect, and the fact that not all actions involve objects. For example, visible actions from our datasets such as “get up", “cut them in half", “getting ready", and “chopped up" cannot be correctly labeled using only object detection. Consequently, we need to use additional video information such as Inception and C3D information.

In general, we find that the text information plays an important role. ELMo embeddings lead to better results than LSTM embeddings, with a relative error rate reduction of 6.8%. This is not surprising given that ELMo uses two bidirectional LSTMs and has improved the state-of-the-art in many NLP tasks BIBREF38 . Consequently, we use ELMo in our multimodal model.

Moreover, the addition of extra information improves the results for both modalities. Specifically, the addition of context is found to bring improvements. The use of POS is also found to be generally helpful.

Conclusion

In this paper, we address the task of identifying human actions visible in online videos. We focus on the genre of lifestyle vlogs, and construct a new dataset consisting of 1,268 miniclips and 14,769 actions out of which 4,340 have been labeled as visible. We describe and evaluate several text-based and video-based baselines, and introduce a multimodal neural model that leverages visual and linguistic information as well as additional information available in the input data. We show that the multimodal model outperforms the use of one modality at a time.

A distinctive aspect of this work is that we label actions in videos based on the language that

accompanies the video. This has the potential to create a large repository of visual depictions of actions, with minimal human intervention, covering a wide spectrum of actions that typically occur in everyday life.

In future work, we plan to explore additional representations and architectures to improve the accuracy of our model, and to identify finer-grained alignments between visual actions and their verbal descriptions. The dataset and the code introduced in this paper are publicly available at http://lit.eecs.umich.edu/downloads.html.

Acknowledgments

This material is based in part upon work supported by the Michigan Institute for Data Science, by the National Science Foundation (grant #1815291), by the John Templeton Foundation (grant #61156), and by DARPA (grant #HR001117S0026-AIDA-FP-045). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Michigan Institute for Data Science, the National Science Foundation, the John Templeton Foundation, or DARPA.