Explainable Activity Recognition in Videos

Chiradeep Roy
University of Texas at Dallas
Richardson, TX
Chiradeep.Roy@utdallas.edu

Mahesh Shanbhag
University of Texas at Dallas
Richardson, TX
Mahesh.Shanbhag@utdallas.edu

Tahrima Rahman
University of Texas at Dallas
Richardson, TX
Tahrima.Rahman@utdallas.edu

Vibhav Gogate
University of Texas at Dallas
Richardson, TX
Vibhav.Gogate@utdallas.edu

Nicholas Ruozzi
University of Texas at Dallas
Richardson, TX
Nicholas.Ruozzi@utdallas.edu

Mahsan Nourani
University of Florida
Gainesville, FL
mahsannourani@ufl.edu

Eric D. Ragan
University of Florida
Gainesville, FL
eragan@ufl.edu

Samia Kabir
Texas A&M University
College Station, Texas
samia.kabir@tamu.edu

ABSTRACT

In this paper, we consider the following activity recognition task: given a video, infer the set of activities being performed in the video along with an assignment of activities to each frame in the video. Although this task can be solved accurately using existing deep learning systems, their use is problematic in interactive settings. In particular, deep learning models are black boxes: it is difficult to understand how and why the system assigned a particular activity to a frame. This reduces the users' trust in the system, especially in the case of end-users who need to use the system on a regular basis. We address this problem by feeding the output of deep learning to a tractable interpretable probabilistic graphical model and then performing joint learning over the two. The key benefit of our proposed approach is that deep learning helps achieve high accuracy while the interpretable probabilistic model makes the system explainable. We demonstrate the power of our approach using a visual interface to provide explanations of model outputs for queries about videos.

CCS CONCEPTS

• **Computing methodologies** → **Activity recognition and understanding**; • **Human-centered computing** → *Graphical user interfaces*; User studies.

KEYWORDS

activity recognition; video processing; deep learning; probabilistic graphical models; cutset networks; explanations

ACM Reference Format:
Chiradeep Roy, Mahesh Shanbhag, Tahrima Rahman, Vibhav Gogate, Nicholas Ruozzi, Mahsan Nourani, Eric D. Ragan, and Samia Kabir. 2019. Explainable Activity Recognition in Videos. In *Joint Proceedings of the ACM IUI 2019 Workshops, Los Angeles, USA, March 20, 2019*, 6 pages.

IUI Workshops '19, March 20, 2019, Los Angeles, USA

1 INTRODUCTION

We propose a two-layer architecture that stacks a tractable, interpretable probabilistic graphical model, specifically a cutset network [10], layer on top of a deep learning layer to address the aforementioned drawback. A possible interpretation of this model is that the deep learning layer provides noisy sensory inputs to the cutset network layer which in turn removes the noise and provides explainability. The interaction graph of the cutset network encodes our prior knowledge about the relationship between various (human interpretable) random variables in the network. The rationale is that the prior knowledge will help correct the errors made by the neural network and thus help improve accuracy. The cutset network provides explainability not only because it is interpretable but also because we can perform tractable (linear time in the size of the model) abductive inference to compute explanations for the decisions made by the model. To model temporal aspects in video, we propose a novel tractable dynamic probabilistic modeling framework called dynamic cutset networks and show that they greatly improve the estimation accuracy.

We experimentally demonstrate the efficacy of our proposed approach by building an interactive visual interface and a machine learning system for activity recognition for the Textually Annotated Cooking Scenes (TaCOS) dataset [11]. The purpose of building this system is two-fold. First, we want to show that we can create a working prototype explainable AI system that not only performs accurate activity recognition in videos but can also generate human understandable explanations and answer queries posed by end-users. Second, we aim to use the resulting system as the basis for user studies of how different types of explanations affect user trust and understanding of machine learning models.

2 RELATED WORK

This effort was inspired by the work of Rohrbach et al. [12] on generating a semantic representation from videos at an activity level. Instead of generating sentences in natural language however, we assign a number of pre-defined labels divided into categories. We do this by using deep-architectures with proven results in order to generate high accuracies for predicting the activity labels. Related efforts have considered the task of dense captioning [4], i.e., generating summaries of texts from particular segments. Song et al. [13] attempted to create captioning methods that require minimum supervision on the TaCOS dataset. Duan et al. [1] attempted to combine caption generation and sentence localization to feed off each other to create a weakly supervised training model. These works focus on creating text summaries for video segments, and as is typical of deep learning approaches, they are essentially black boxes. Our approach, on the other hand, aims to create a semantic representation for activities in each frame that can both be used to answer queries easily as well as generate explanations that justify these answers.

There have also been a number of studies on how trust influences interactions between humans and automated systems, e.g., [6], [7], [5] and [2]. These studies examine factors that might affect the trust of the user in the system, such as showing the past performance of the system and making the working of the system more understandable (Lee et al. [5]). Hoffman [3] provides a more detailed taxonomy of such factors and explain how trust is context-specific and dynamic. In other words, trust might vary with respect to specific contexts of automation and must also be maintained over time. Our aim is to be able to control and measure the trust of humans with respect to these systems in order to better understand what kind of explanations influence the trust variable.

3 PROBLEM DESCRIPTION

In this section, we will define the problem in precise terms and also describe the framework we will be using for question-answering and generating explanations.

3.1 Activity Recognition with Explanations

The objective of our proposed system is two-fold: (a) perform accurate activity recognition in videos, and (b) compile knowledge acquired while learning to recognize activities into an explanatory model. The latter can then be used to explain why a particular activity was assigned to a frame by the system.

We define an activity as a (*action*, *object*, *location*) triple. The *action* component forms the core part of the activity. These are usually verbs such as wash, cut, slice, open, etc. The *object* component denotes the entities over which the activity is performed. These are generally nouns such as apples, refrigerator, cutting board, knife, etc. Finally, the *location* component tells us *where* the activity is taking place. These are generally location nouns such as kitchen, bathroom, counter top, sink, etc. but can also overlap with the nouns we use as objects. For example, when we "kick open a door," the activity is "kick" and the object is "door," but the same entity might play a different semantic role in a different activity such as if a baby "draws a picture on the door." Here "draw" is the activity, "picture" is the object, and "door" is the location.

For the purposes of our initial system, we make the following simplifying assumptions.

(1) We train our system on a closed-domain. In this study, we use cooking videos.
(2) We assume that only one major activity is taking place per frame (minor activities are ignored).
(3) The action must always be present, while the object and the location are optional. For reflexive actions, such as "walking," the object is "None."

In future, we plan on making activities more complex (so that we can pose more interesting queries on them). We also plan on defining hierarchies on activities to create 'super' and 'sub' activities. For instance, taking out an egg from a refrigerator might be a sub-activity of cooking the egg, which in turn might be a sub-activity of cooking a full-course meal.

3.2 Formulating Queries

Now that we have precisely defined activities, we can define queries and explanations. A query is similar to an activity in that it is also a triple of the form (*action*, *object*, *location*). Once this triple is formulated, we run a filtering query—which seeks to assign an activity to each frame in the video based on the current and previous frames (but not future frames)—to check how many frames match our query. For instance, if we wanted to ask the system if the person in the video sliced an orange on the cutting board, our query tuple would look something like: (*slice*, *orange*, *cutting-board*). Once we have formulated this tuple, we simply ask the system to search for frames where the probability of this tuple being the actual activity is above a certain threshold.

We envision that our system will be used to answer a wide-range of queries including but not limited to:

(1) **Selection queries.** Did the person slice an orange on the counter?
(2) **Counting.** How many oranges did the person slice on the counter?
(3) **Recipe.** Did the person deviate from a pre-defined recipe?
(4) **Complex**: Combination of all of the above

In this paper, we will focus on selection queries and leave the remaining for future work.

3.3 Generating Explanations

The aim of our system is not only to answer queries but also to explain the predictions to the end-user. As mentioned in the previous section, selection queries involve formulating the query into a (*action*, *object*, *location*) tuple and then filtering on the video to find frames which have a high probability of containing the activity. Note that two of the three parameters can be optional. This means that our queries can be as simple as "Did the person wash something?" (*wash*, *?*, *?*) or "Did anything happen on the kitchen counter?" (*?*, *?*, *counter*), etc.

We seek to build a system that can generate three types of explanations:

(1) **Video Explanations:** When the system answers "yes" we want the system to highlight segments (possibly more than one) of the video where the activity happened. For "no" answers, we want the system to highlight segments where a related activity happened (e.g., carrots were cut in the video but not oranges). If no related activity is found in case of a "no answer," we want the system to output the most likely activity in the video.

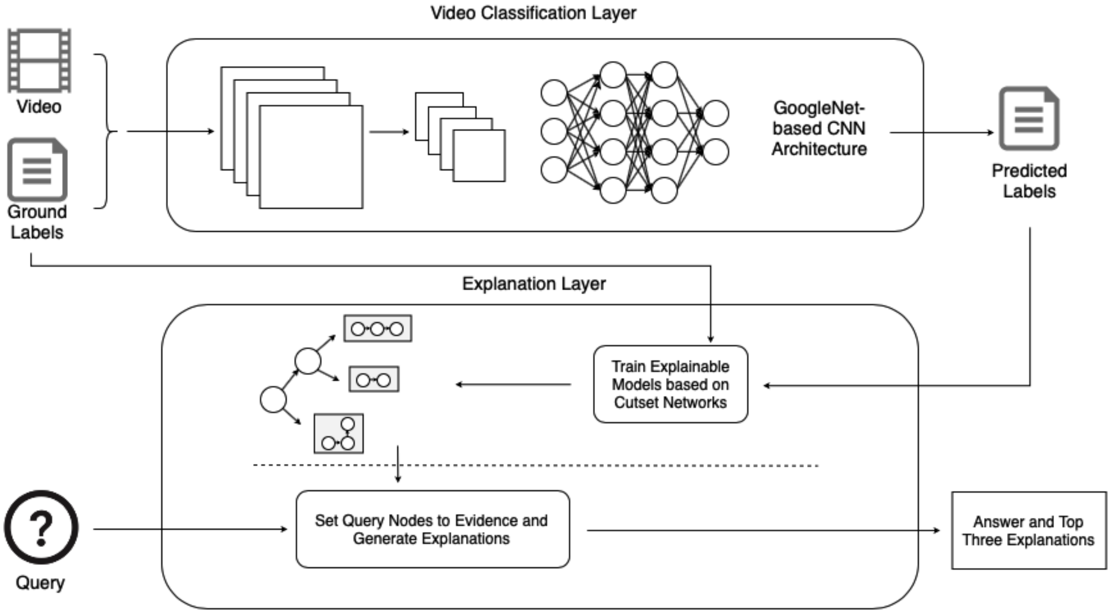

Video Classification Layer
Video
Ground Labels
GoogleNet-based CNN Architecture
Predicted Labels
Explanation Layer
Train Explainable Models based on Cutset Networks
Set Query Nodes to Evidence and Generate Explanations
Query
Answer and Top Three Explanations

Figure 1: **High-level Architecture and Data Processing Pipeline. Our system has two layers: a video classification layer based on deep learning whose output is fed to an explanation layer which is based on cutset networks [10], a tractable interpretable probabilistic model. During the learning phase, the classification layer uses the video and the ground truth (labels) as input and learns a mapping from frames to object, action and location. During the learning phase, the explanation layer uses the labels predicted by the classification layer and ground truth as input and learns a mapping from predicted labels to the ground truth. During the query phase, the system answers questions by performing abductive inference over the cutset network (in the explanation layer).**

(2) **Ranked (action,object,location) Triples:** We want the system to display top-*k* predicted activity triples in the video that are relevant to the query.
(3) **Most Probable Entities:** We want the system to display the most probable actions, objects and locations (along with their likelihood) that are relevant to the query.

4 SYSTEM DESCRIPTION

This section explains the architecture and the functioning of our system in detail. Fig. 1 shows a high-level overview of the components of the system and the processing pipeline. Roughly speaking, the system can be categorized into the following two layers:

(1) **Video Classification Layer**
In this layer, we use a convolutional neural network whose architecture is based on GoogleNet [15]. The network takes as input a number of video frames, a vocabulary file, and a set of annotated ground truths and then uses a version of backpropagation with the Adam algorithm to learn the weights. The output is a set of labels that correspond to each vocabulary word. The accuracy of network for the multilabel classification task is measured using standard information retrieval metrics.
(2) **Explanation Layer**
The ground labels and the predicted labels from the previous layer are fed to this layer and are used to train a Conditional Cutset Network [10]. Once training is done, we are now in a position to pose queries to the system. The system uses the trained model to answer these queries and returns the top *k*-best explanations using sampling-based inference techniques.

Next, we will describe each layer in more detail.

4.1 Video Classification Layer

For this layer, we will be using the BAIR/BLVC GoogleNet Model [14]. This is a pre-trained model that uses the ILSVRC dataset. There are 22 layers in the network. It uses a key component called the Inception Model (for details, please refer [15]) that creatively uses convolutions of size 1x1 to increase the representational power the network without increasing the number of parameters. These layers are stacked one on top of the other. The architecture preserves translational invariance.

The other reason we used GoogleNet is because the computational load does not increase exponentially with the increase in layers. This is because the Inception modules use 1x1 convolutions. The vanishing gradient problem is taken care of by using rectified linear units in the perceptrons. This also avoids the introduction of sparse activations in the hidden layers of the network.

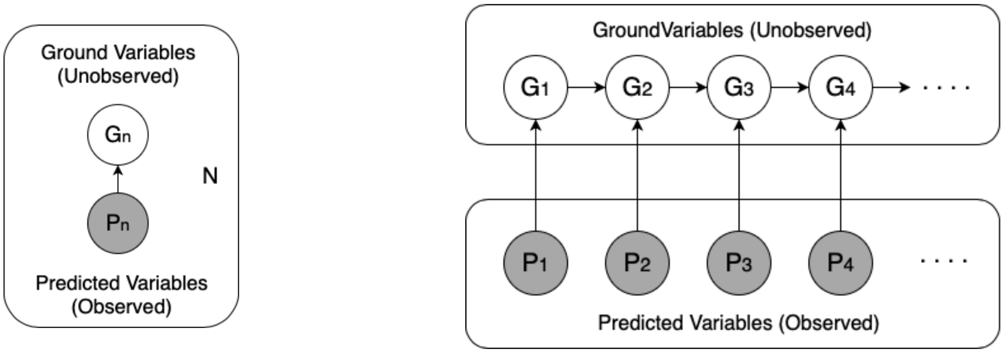

Ground Variables (Unobserved)
Gn
N
Pn
Predicted Variables (Observed)
GroundVariables (Unobserved)
G1
G2
G3
G4
. . . .
P1
P2
P3
P4
. . . .
Predicted Variables (Observed)

Figure 2: (a) Architecture of Sensor Network using Plate Notation (b) Architecture of Dynamic Cutset Network

Finally, we made a slight modification to this architecture by replacing the softmax layer on top with a fully-connected layer with 28 nodes that use the sigmoid cross-entropy loss.

The video is first divided into frames which are then fed to this layer along with the ground labels for each frame. For instance, if we have 10,000 frames in our dataset and the 328th frame has the person taking a carrot out from the fridge, then the 328th row in the ground labels file will have (*take-out*, *carrot*, *fridge*) set to true and all other labels set to false.

The output layer has a node for every label in the vocabulary. For the purposes of this experiment, we are using 28 labels. The neural network takes in the video frame as an input and tries to guess the correct activity labels for each frame. The accuracy of this multi-label classification task is measured using standard metrics such as the Hamming loss and the Jaccard index.

4.2 Explanation Layer

At the end of processing the first layer, we have a list of predicted labels for selected video frames. In the explanation layer, we correct errors in the predicted labels using a probabilistic model at each frame. Also, we model the dynamics as well as persistence (activities don't change rapidly between frames) using a temporal probabilistic model.

The explanation model uses a Conditional Cutset Network (CCN), [10] to correct the errors at each frame. The reason for using this probabilistic model is two-fold. First it is interpretable in that its structure and parameters can be explained to an expert user using concepts from graph theory and probability theory respectively. Second, it is a tractable model. In particular, the model can answer queries in time that scales linearly with its size. At a high level, the CCN treats the output of the neural network as a noisy sensor (see Fig. 2(a)) and computes a conditional joint probability distribution over the true labels given the predicted (noisy) labels.

To model dynamics and persistence, we propose to use dynamic conditional cutset networks (see Fig. 2(b)). To control the number of parameters and learning complexity, we use 1-Markov and stationarity assumptions, which are widely used in temporal models literature [9]. Specifically, we assume that each frame is conditionally independent of all frames before it given the previous frame (1-Markov) and all conditional distributions are identical (stationarity). We model these conditional distributions using conditional cutset networks. Thus, a dynamic conditional cutset network is a two-tuple. At the first frame, we have a conditional cutset network which models the conditional distribution over the labels in the frame given labels predicted by the neural network. At subsequent frames, we have a conditional cutset network which models the conditional distribution over the labels in the frame given labels predicted by the neural network and the true labels in the previous frame.

The three explanation types (video, ranked triples and most probable entities) mentioned in the previous section can be computed from the explanation layer by performing abductive inference (cf. [8]) over the dynamic cutset network. Since inference in cutset networks is linear in the size of the network, once learned from data, our explanation layer yields real-time query answers and explanations.

4.3 Dataset and Data Processing

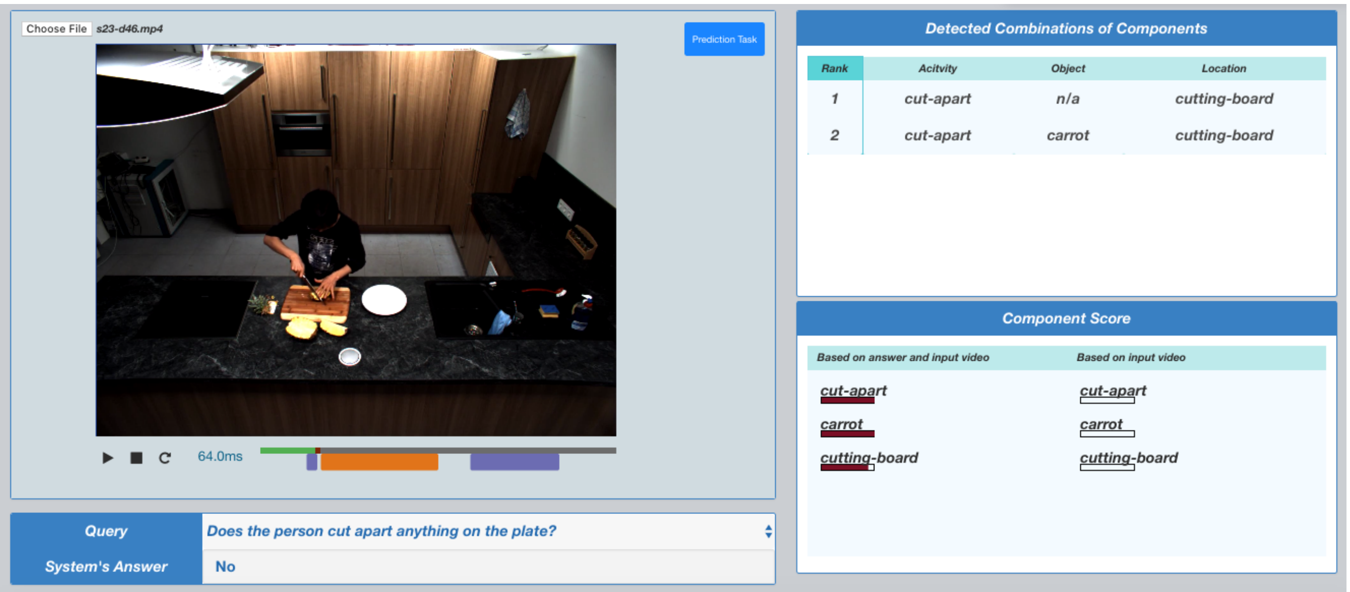
The dataset we are using for this experiment is the TACoS Multi-Level corpus - MPII Cooking 2 dataset by Rohrbach et al. [12]. Each video is annotated as follows. The annotations are *filename* (e.g., s24-d28), *startFrame* (e.g. 781), *endFrame* (e.g., 1098), *descriptionIdx* (e.g., 3), *ignore* (e.g., 0), *sentenceProcessed* (e.g., the person took out the cutting board from the drawer and placed it on the counter), *activity* (e.g., take out), *tool* (e.g., hand), *object* (e.g., cutting board), *source* (e.g., drawer), *target* (e.g., counter).

We generate the ground labels from these annotations. We create a text file where every row corresponds to a frame of some video. In addition to the name of the video and the frame number, we have 28 0/1 values depending on which labels are off and which are on. The order of the labels follows the same order as that of the output nodes in the video classification layer. We extract the *activity*, *object* and *location* by punching together the *source* and *destination* fields. For instance, the location *drawer-counter* indicates that the source of the action is the *drawer* and the final destination is the *counter*. We repeat this process for each video frame in our training set.

After we have a list of ground labels and predicted labels, we train the explanation model and pose queries to it. This part will be discussed in the *Experiments* section.

4.4 User Interface

The prototype uses an interactive visual interface that allows users to load videos, ask queries, and review the model output along with explanations. The goal for the interface design was to limit



Detected Combinations of Components

Rank	Acitvity	Object	Location
1	cut-apart	n/a	cutting-board
2	cut-apart	carrot	cutting-board

Component Score

Based on answer and input video	Based on input video
cut-apart	cut-apart
carrot	carrot
cutting-board	cutting-board

Query	Does the person cut apart anything on the plate?
System's Answer	No

Figure 3: The interactive visual interface allows users to load videos and ask queries. The interface shows the AI's answer along with explanatory elements for the output. The most relevant portions of the video play time are shown by colored bars beneath the video, and the right side shows detected video components and combinations of components relevant to the video and query.

the amount of model information presented to the user in order to avoid overwhelming users with information. For this reason, the system uses simple visual representations in the form of graphical annotations, textual component lists, and simple bar charts. Figure 3 shows the interface.

The interface includes a video player that allows users to watch the selected video to help review and analyze the system's answers to the queries. When a query is submitted, the video player highlights the most relevant segments of the video through visual annotations added under the video play bar (shown as orange and purple bars under the video in Fig. 3). The video player will also automatically jump to the appropriate segment to help users see the video frames most important for determining the output. In addition, the right side of the interface summarizes the detected video components (*activities*, *objects*, and *locations*) for the query as well as detected combinations of components. To help users to quickly judge component scores, graphical bars are shown underneath detected components to visually represent the values of the component scores. Users can select different video segments to view the corresponding component scores and combinations from different portions of the video.

5 EVALUATION

In order to evaluate our system, we designed two experiments using the TACoS video dataset and annotations. We performed a model evaluation to measure how successful the system is at identifying activities and providing explanations, and we performed a preliminary user study to assess system understandability and usability.

5.1 Model (Machine Learning) Evaluation

We selected 60313 frames for training and 9355 frames for testing distributed over 17 videos. For each set, we selected a set of ground labels and used the video classification layer to generate the predicted labels. We performed the following ablation study: (1) Our system in which the explanation layer is removed (GoogleNet); (2) Our system in which the dynamic model is removed but the sensor model is kept at each frame (Sensor Model); and (3) the full system (dynamic CCNs).

Table 1 outlines the accuracy scores for correct activity recognition according to various evaluation metrics. Since predicting each activity correctly is a multilabel classification task, we use K-Group measures to calculate the overall percentage of instances where K labels out of the total number of labels were predicted currently. We use the group heuristics K-1, K-2, and K-3 (since each activity comprises of *action*, *object* and *location*). In addition, we also use standard measures such as the Hamming Loss and the Jaccard Index. We observe that in general (with a few exceptions) dynamic CCNs is more accurate than the sensor model which in turn is more accurate than GoogleNet.

5.2 Human Feedback

We sought user feedback via a preliminary testing with seven participants. All participants were experienced with AI but were unfamiliar with the specifics of the system. Rather than testing the querying functionality and model accuracy, we were interested in general user feedback about perception of the system and explanations. Participants were asked to review a set of four videos with five pre-determined queries per video.

Evaluation Metric	GoogleNet	Sensor	Dynamic CCNs
K-1	0.9335	**0.9677**	0.9649
K-2	0.8557	0.8998	**0.9156**
K-3	0.7918	0.7962	**0.8127**
Jaccard Index	0.8608	0.8559	**0.8628**
Hamming Loss	0.1392	0.1286	**0.1200**

Table 1: Accuracy for Activity Recognition on All Videos. Bold results indicate the best performing model.

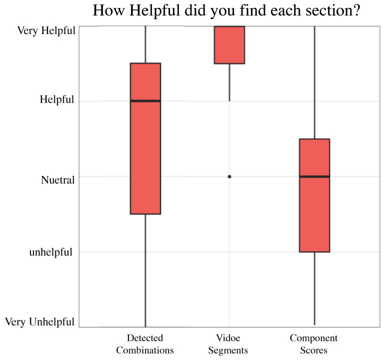


Figure 4: Preliminary results which summarize how participants used each section. We observe that participants found video segments very helpful in completing the task.

After participants finished the study, they were asked about the utility of each section of the interface, in particular, whether the section helped them complete the given task quickly. The results of this questionnaire are shown in Fig. 4. Our testing demonstrates that users were able to effectively and easily use the system to submit queries and review results. The simplicity of the visual design enabled participants to easily see the corresponding relevant segments of the video and to quickly assess the accuracy of the model's output. However, participants had mixed thoughts on component scores (most probable entities) and component combinations (ranked triples).

We plan to further examine differences in interpretation and study the utility of the explanation design through more extensive user testing in the future. In particular, we will run more formal controlled experiments of how different types of explanation and amount of explanatory information affect understanding of the model and perception of its accuracy.

6 CONCLUSION

From our preliminary user studies, a strong positive correlation has been observed between user trust and the goodness of explanations. As a part of our future work, we plan on improving upon our current system in the following manner:

(1) Adding support for more vocabulary in the video classification layer
(2) Adding support for complex models and automatic query conversion from natural language in the explanation layer
(3) Adding support for a larger variety of queries

We expect that adding these features will increase the trust of the users in the system since the range of activities, the precision of the explanations as well as the types of queries will all increase.

7 ACKNOWLEDGMENTS

This work was supported by the DARPA Explainable Artificial Intelligence (XAI) Program under contract number N66001-17-2-4032, and by the National Science Foundation grants IIS-1652835, IIS-1528037, and IIS-1762268.

REFERENCES

[1] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. 2018. Weakly Supervised Dense Event Captioning in Videos. In *Advances in Neural Information Processing Systems*. 3062–3072.

[2] Kevin Anthony Hoff and Masooda Bashir. 2015. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors* 57, 3 (2015), 407–434.

[3] Robert R Hoffman. 2017. 8 A Taxonomy of Emergent Trusting in the Human–Machine Relationship. *Cognitive Systems Engineering: The Future for a Changing World* (2017).

[4] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos.. In *ICCV*. 706–715.

[5] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human factors* 46, 1 (2004), 50–80.

[6] Bonnie M Muir. 1994. Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics* 37, 11 (1994), 1905–1922.

[7] Bonnie M Muir and Neville Moray. 1996. Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation. *Ergonomics* 39, 3 (1996), 429–460.

[8] Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

[9] Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

[10] Tahrima Rahman, Prasanna Kothalkar, and Vibhav Gogate. 2014. Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of Chow-Liu trees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 630–645.

[11] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. 2013. Grounding Action Descriptions in Videos. *Transactions for the Association of the Computational Linguistics (TACL)* 1 (2013), 25–36.

[12] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. 2014. Coherent multi-sentence video description with variable level of detail. In *German conference on pattern recognition*. Springer, 184–195.

[13] Young Chol Song, Iftekhar Naim, Abdullah Al Mamun, Kaustubh Kulkarni, Parag Singla, Jiebo Luo, Daniel Gildea, and Henry A Kautz. 2016. Unsupervised Alignment of Actions in Video with Text Descriptions.. In *IJCAI*. 2025–2031.

[14] Christian Szegedy. 2014. Googlenet pre-trained model. http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.