Improving Interpretability of Deep Neural Networks with Semantic Information *

Yinpeng Dong Hang Su Jun Zhu Bo Zhang
Tsinghua National Lab for Information Science and Technology
State Key Lab of Intelligent Technology and Systems
Center for Bio-Inspired Computing Research
Department of Computer Science and Technology, Tsinghua University
dongyinpeng@gmail.com {suhangss, dcszj, dcszb}@mail.tsinghua.edu.cn

Abstract

Interpretability of deep neural networks (DNNs) is essential since it enables users to understand the overall strengths and weaknesses of the models, conveys an understanding of how the models will behave in the future, and how to diagnose and correct potential problems. However, it is challenging to reason about what a DNN actually does due to its opaque or black-box nature. To address this issue, we propose a novel technique to improve the interpretability of DNNs by leveraging the rich semantic information embedded in human descriptions. By concentrating on the video captioning task, we first extract a set of semantically meaningful topics from the human descriptions that cover a wide range of visual concepts, and integrate them into the model with an interpretive loss. We then propose a prediction difference maximization algorithm to interpret the learned features of each neuron. Experimental results demonstrate its effectiveness in video captioning using the interpretable features, which can also be transferred to video action recognition. By clearly understanding the learned features, users can easily revise false predictions via a human-in-the-loop procedure.

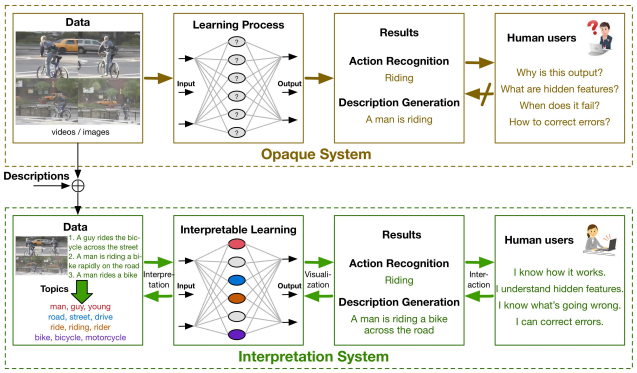


Figure 1. An overview of our interpretation system (bottom) compared with an opaque system (top). An opaque system often learns abstract and incomprehensible features. Human users have to accept the decisions from the system passively, but are unable to understand the rationale of the decisions and interact with it. To address this issue, we incorporate topics embedded in human descriptions as semantic information, to improve interpretability of DNNs during the learning process. The learned features of each neuron can be associated with a topic (*e.g.*, topic "road" with top related words like road, street, and drive can interpret the learned features of the blue neuron). With the aids of these interpretable features, human users can easily visualize and interact with the system, which allows a human-in-the-loop learning procedure.

1. Introduction

Deep Neural Networks (DNNs) have demonstrated state-of-the-art and sometimes human-competitive performance in numerous vision-related tasks [19], including image classification [18, 30], object detection [14, 28] and image/video captioning [33, 34]. With such success, DNNs have been integrated into various intelligent systems as a key component, *e.g.*, autonomous car [16, 5], medical image analysis [15], financial investment [1], *etc*. The high-performance of DNNs highly lies on the fact that they often stack tens of or even hundreds of nonlinear layers, and encode knowledge as numerical weights of various node-to-node connections.

Although DNNs offer tremendous benefits to various applications, they are often treated as "black box" models because of their highly nonlinear functions and unclear working mechanism [3]. Without a clear understanding of what a given neuron in the complex models has learned and how it interacts with others, the development of better models typically relies on trial-and-error [37]. Furthermore, the effectiveness of DNNs is partly limited by its inabilities to ex-

*The work was supported by the National Basic Research Program (973 Program) of China (No. 2013CB329403), National NSF of China Projects (Nos. 61571261, 61620106010, 61621136008), China Postdoctoral Science Foundation (No. 2015M580099), Tiangong Institute for Intelligent Computing, and the Collaborative Projects with Tencent.

plain the reasons behind the decisions or actions to human users. It is far from enough to provide eventual outcomes to the users especially for highly regulated environments, since they may also need to understand the rationale of the decisions. For example, a driver of an autonomous car is eager to recognize why obstacles are reported so that he/she can decide whether to trust it; and radiologists also require a clearly interpretable outcome from the system such that they can integrate the decision with their standard guideline when they make diagnosis. As an extreme case in [24], a DNN can be easily fooled, *i.e.*, it is possible to produce images that DNNs believe to be recognizable objects with nearly certain confidence but are completely unrecognizable to humans. In summary, the counter-intuitive properties and the black-box nature of DNNs make it almost impossible for one to reason about what they do, foreseen what they will do, and fix the errors when potential problems are detected. Therefore, it is imperative to develop systems with good interpretability, which is an essential property for users to clearly understand, appropriately trust, and effectively interact with the systems.

Recently, many research efforts have been devoted to interpreting hidden features of DNNs [12, 25, 38, 37], and have made several steps towards interpretability, *e.g.*, the de-convolutional networks [37] to visualize the layers of convolutional networks, and the activation maximization [12] to associate semantic concepts with neurons of a CNN. A few attempts have also been made to explore the effectiveness of various gates and connections of recurrent neural networks (RNNs) [10, 17]. Interpretability also bring us some benefits like weakly supervised detection [39]. However, these works often focus on analyzing relatively simple architectures such as AlexNet [18] for image classification. There still lack interpretation techniques for more complex architectures that integrates both CNN and RNN, in which the learned features are difficult to interpret and visualize. More importantly, these methods perform interpretation and visualization after the training process. It means that they can only explain a given model, but are unable to learn an interpretable model. Such a decoupling between learning and interpretation makes it extremely hard (if possible at all) to get humans to interact with the models (*e.g.*, correct errors).

In this paper, we address the above limitations by presenting a method that incorporates the interpretability of hidden features as an essential part during the learning process. A key component of our method is to measure the interpretability and properly regularize the learning. Instead of pursuing a generic solution, we concentrate our attention on the video captioning task [32], for which DNNs have proven effective on learning highly predictive features while the interpretability remains an issue as other DNNs do. In this task, we leverage the provided text descriptions, which include rich information, to guide the learning. We first extract a set of semantically meaningful topics from the corpus, which cover a wide range of visual concepts including objects, actions, relationships and even the mood or status of objects, therefore suitable to represent semantic information. Then we parse the descriptions of each video to get a latent topic representation, *i.e.*, a vector in the semantic space. We integrate the topic representation into the training process by introducing an *interpretive loss*, which helps to improve the interpretability of the learned features.

To further interpret the learned features, we present a *prediction difference maximization* algorithm. We also present a human-in-the-loop learning procedure, through which users can easily revise false predictions and the model based on the good interpretation of the learned features. Our results on real-world datasets demonstrate the effectiveness.

2. Methodology

In this section, we present the key components of our interpretation system. We first overview the system on the video captioning task. We then present an attentive encoder-decoder network, which incorporates an interpretive loss to learn interpretable features. Afterwards, we present a prediction difference maximization algorithm to interpret the learned features of each neuron. We will introduce a human-in-the-loop learning procedure by leveraging the interpretability in Section. 4.

2.1. Overview

Our goal is to improve the interpretability of DNNs without losing efficiency. By designing proper learning objective, we expect to learn the hidden features with two properties—discriminability and interpretability. Discriminability defines the ability that the features can distinguish different inputs and predict corresponding outputs. Interpretability measures the extent that human users can understand and manipulate the learned features. These two properties are often contradictory in DNNs. According to the fundamental bias-variance tradeoff, a complex DNN can be highly competitive in prediction performance but its hidden features are often too abstract to be understandable for humans. On the other hand, a simple DNN can lead to more interpretable features, but it may degrade the performance. In order to break this dilemma, we introduce extra semantic information to guide the learning process. In this paper, we will concentrate on the video captioning task [32], although the similar ideas can be generalized to other scenarios.

Specifically, the video captioning task aims to automatically describe video content with a complete and natural sentence. Recent works have demonstrated that video captioning can benefit from the discovery of multiple semantics, including obejcts, actions, relationships and so on. Liu

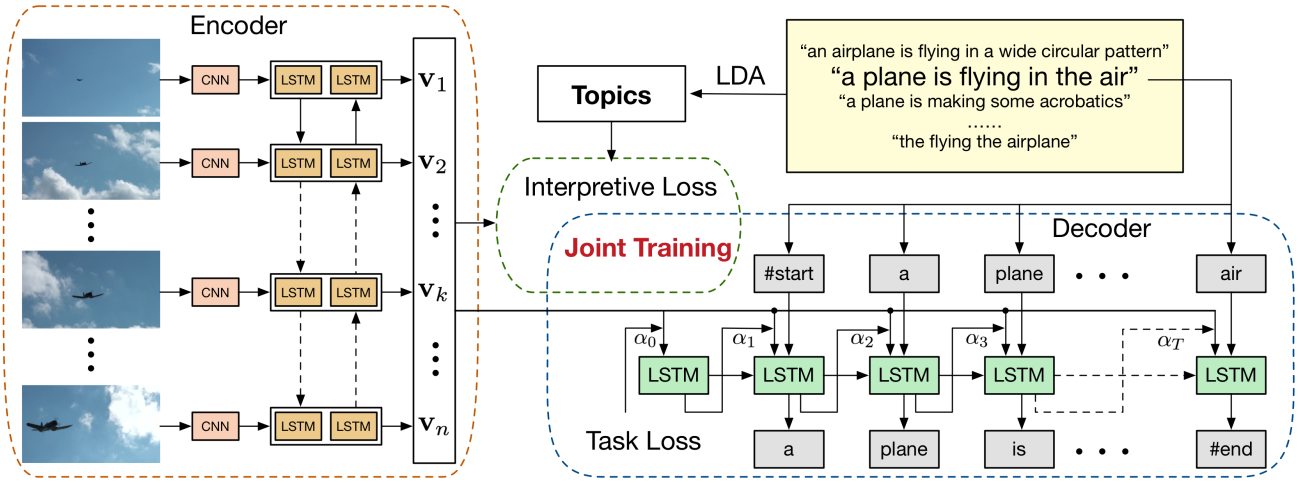


Figure 2. The attentive encoder-decoder framework for the video captioning task, which can automatically learn interpretable features. We stack a CNN model and a bi-directional LSTM model as encoder to extract video features $\{\mathbf{v}_1, ...\mathbf{v}_n\}$, and then feed them to an LSTM decoder to generate descriptions. The attention mechanism is used to let the decoder focus on a weighted sum of temporal features with weight α_t. We extract latent topics from human labeled descriptions as semantic information and introduce an interpretive loss to guide the learning towards interpretable features, which is optimized jointly with the negative log-likelihood of training descriptions.

et al. [20, 21] proposed one original method for joint human action modeling and grouping, which can provide comprehensive information for video caption modeling and explicitly benefit understanding what happens in the given video. As a video is more than a set of static images, in which there are not only the static objects but also the temporal relationships and actions, video analysis often requires more complex network architectures. For example, some works have shown the effectiveness of DNNs on video analysis [2, 35] when stacking a hierarchical RNN on top of some CNN layers. Such a complex network makes it more challenging to learn interpretable hidden features and hinders the interaction between the models and human users. To address this issue, we propose a novel technique to improve the interpretability of the learned features by leveraging the latent topics extracted from video descriptions.

The overall framework is shown in Fig. 2, which consists of an attentive encoder-decoder network for video caption generation and an interpretive loss to guide the learning towards semantically meaningful features.

Formally, in the training set, each video $\mathbf{x}$ has n sample frames along with a set of N_d descriptions $\mathbf{Y} = \{\mathbf{y}^1, \mathbf{y}^2, ..., \mathbf{y}^{N_d}\}$. For each $\mathbf{y} \in \mathbf{Y}$, let $(\mathbf{x}, \mathbf{y})$ denote a training video-description pair, where $\mathbf{y} = \{y_1, y_2, ..., y_{N_s}\}$ is a description with N_s words. We first transform the input $\mathbf{x}$ into a set of D_v-dimensional hidden features $V = \{\mathbf{v}_1, ..., \mathbf{v}_n\}$ by using an encoder network. Then, the hidden features are decoded to generate the description $\mathbf{y}$. We define the task-specific loss as the negative log-likelihood of the correct description

$$L_T(\mathbf{x}, \mathbf{y}) = -\log p(\mathbf{y}|\mathbf{x}). \quad (1)$$

We parse the text descriptions $\mathbf{Y}$ to get a semantically meaningful representation (*i.e*., a topic representation in this paper), which is denoted as $\mathbf{s}$. Then, we introduce an interpretive loss $L_I(V, \mathbf{s})$ to measure the compliance of the learned features with respect to the semantic representation $\mathbf{s}$. Putting together, we define the overall objective function as

$$L(\mathbf{x}, \mathbf{y}, \mathbf{s}) = -\log p(\mathbf{y}|\mathbf{x}) + \lambda L_I(V, \mathbf{s}). \quad (2)$$

The tradeoff between these two contradictory losses is captured by a balancing weight λ. An interpretation system with high-quality can be realized based on an appropriate λ, which can be obtained using the validation set.

After training (See the experimental section for details), we use the prediction difference maximization algorithm to interpret the learned features of each neuron by a topic. Below, we elaborate each part.

2.2. Attentive Encoder-Decoder Framework

We adopt an attentive encoder-decoder framework similar to [34] for video captioning. The attention mechanism is used to let the decoder selectively focus on only a small subset of frames at a time.

A key difference from previous works [32, 34, 26] which use CNN features as video representations is that we stack a bi-directional LSTM model [29] on top of a CNN model to characterize the video temporal variation in both input directions. Such an encoder network makes the vector representation $\mathbf{v}_i$ of the i-th frame capture temporal information, and thus the interpretive loss (defined later in Eq. 8) lets the internal neurons learn to detect latent topics in the video. So the learned features are more likely to be both discriminative and interpretable.

To generate the description sentences, we use an LSTM

model as the decoder. At each time step, the input to the LSTM decoder can be represented by $[\mathbf{y}_{t-1}, \phi_t(V)]$, where $\mathbf{y}_{t-1}$ is the previous word and $\phi_t(V)$ is the dynamic weighted sum of temporal feature vectors

$$\phi_t(V) = \sum_{i=1}^{n} \alpha_i^t \mathbf{v}_i. \quad (3)$$

The attention weight α_i^t reflects the importance of the *i*-th temporal features at time step *t* [34], which is defined as

$$\alpha_i^t = \frac{\exp(\mathbf{w}_a \tanh(\mathbf{U}_a \mathbf{h}_{t-1} + \mathbf{T}_a \mathbf{v}_i + \mathbf{b}_a))}{\sum_{j=1}^{n} \exp(\mathbf{w}_a \tanh(\mathbf{U}_a \mathbf{h}_{t-1} + \mathbf{T}_a \mathbf{v}_j + \mathbf{b}_a))}, \quad (4)$$

where $\mathbf{w}_a$, $\mathbf{U}_a$, $\mathbf{T}_a$ and $\mathbf{b}_a$ are the parameters that are jointly estimated with the other parameters. We adopt the same strategy as [33] to initialize the memory state and hidden state as

$$\begin{bmatrix} \mathbf{c}_0 \\ \mathbf{h}_0 \end{bmatrix} = \begin{bmatrix} f_{init,c} \\ f_{init,h} \end{bmatrix} (\frac{1}{n} \sum_{i=1}^{n} \mathbf{v}_i), \quad (5)$$

where $f_{init,c}$ and $f_{init,h}$ are both multilayer perceptions, which can also be jointly estimated.

At each time step, we use the LSTM hidden state $\mathbf{h}_t$ to predict the following word, and define a probability distribution over the set of possible words by using a softmax layer

$$\mathbf{p}_t = \text{softmax}(\mathbf{W}_p[\mathbf{h}_t, \phi_t(V), \mathbf{y}_{t-1}] + \mathbf{b}_p). \quad (6)$$

Therefore, we can predict the next word based on such probability distribution until the end sign is emitted. The log-likelihood of the sentence is therefore the sum of the log-likelihood over the words

$$\log p(\mathbf{y}|\mathbf{x}) = \sum_{t=1}^{N_s} \log p(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{x}; \theta), \quad (7)$$

where θ are the parameters of the attentive encoder-decoder model.

2.3. Interpretive Loss

The above architecture for video captioning incorporates both CNN and RNN to encode the spatial and temporal information. The complex architecture makes internal neurons learn more abstract features than a single CNN or RNN, and these features are typically hard to interpret by human users. To improve the interpretability, we introduce an interpretive loss, which makes the neurons learn to detect semantic attributes in the text descriptions. For humans, it is natural and easy to understand a concept in text descriptions.

In our method, instead of using the raw description data which can be very sparse and high-dimensional vectors (*e.g.*, in bag-of-words or tf-idf format), we adopt a topic model to learn a semantic representation. As proven in previous work [13, 7], topic models can extract semantically meaningful concepts (or themes) that are useful for visual analysis tasks. Furthermore, compared to the raw text descriptions, the representations by topic models can better capture the global statistics in a corpus as well as synonymy and polysemy [4]. Here, we adopt the most popular topic model, *i.e.*, Latent Dirichlet Allocation (LDA) [4], which has been applied to image/video/text analysis tasks [13, 7, 6, 40]. Specifically, LDA is a hierarchical Bayesian model, in which each document is represented as a finite mixture over topics and each topic is characterized by a distribution over words. In our case, we concatenate all of the single descriptions in $\mathbf{Y}$ together to form a "document". Here, we adopt WarpLDA [9] to efficiently estimate the parameters for an N_t topics LDA model, and set N_t to 100 in experiments. The top words from the learned topics are illustrated in Table. 1. We can see that each topic has a good correspondence to a meaningful semantic attribute.

people	people, group, men, line, crowd
woman	woman, lady, women, female, blond
man	man, guy, unique, kind, bare
dance	dancing, dance, stage, danced, dances
walk	walking, walks, race, turtle, walk
eat	eating, food, eats, eat, ate
play	playing, plays, play, played, instrument
field	grass, field, yard, run, garden
dog	dog, tail, barking, wagging, small
cat	cat, licking, cats, paws, paw

Table 1. Sampled latent topics with their high-probability words. We have named the topics according to these words.

After training, we parse each description document to get the latent topic representation, from which the words are generated. We encode the topic representation for a video as a binary vector $\mathbf{s} = [t_1, t_2, ..., t_{N_t}] \in \{0, 1\}^{N_t}$, whose *i*-th element t_i is set to 1 when *i*-th topic occurs in the descriptions, and 0 otherwise. This vector can be obtained by running the Gibbs sampler in WarpLDA. We use a binary vector here rather than a real-valued vector denoting the average probability of each topic, because it provides an easy way to interpret the learned features of each neuron by a topic, when applying the prediction difference maximization algorithm described in Section. 2.4.

Given the topic representations, we define the interpretive loss as

$$L_I(V, \mathbf{s}) = \|f(\frac{1}{n} \sum_{i=1}^{n} \mathbf{v}_i) - \mathbf{s}\|_2^2, \quad (8)$$

where $f : D_v \rightarrow N_t$ is an arbitrary function mapping video features to topics. This formulation can be individually viewed as a multi-label classification task, where we

predict topics given a set of video features. The choice of the function f is also a tradeoff between interpretability and task performance. A complex function with a large number of parameters will increase the interaction among different neurons, leading to hard-to-interpret features again. On the contrary, a too simple function will limit the discriminability of the learned features. For example, an identity mapping will turn the hidden features into a replica of topics, which may degrade the performance for caption generation. Here, we adopt a two-layer perception as f. To avoid overfitting, we use "mean pooling" features over all frames as input. We use l_2-norm to define interpretive loss because it's simple and effective to build a correspondence between neurons and topics and it performs well in practice. We will see in Section. 3.2 how the interpretive loss help to learn interpretable features.

2.4. Prediction Difference Maximization

To analyze the correspondence between neurons and topics and semantically interpret the learned features, we propose a prediction difference maximization algorithm, which is similar with a concurrent and independent work [41], to represent the learned features of each neuron by a topic. This method is different from the activation maximization methods [12, 25], where they aim to find the input patterns (*e.g.*, image patches) that maximally activate a given neuron. The reason why we do not use activation maximization is that some neurons represent temporal actions (*e.g.*, playing, eating), which cannot be represented by static image patches.

Specifically, in a video with topic representation $\mathbf{s}$, for each topic i that $t_i = 1$ and $i \in [1, ..., N_t]$, we expect to find a neuron j_i^* that

$$j_i^* = \arg\max_j ([f(\mathbf{v})]_i - [f(\mathbf{v}_{\setminus j})]_i), \tag{9}$$

where $\mathbf{v} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i$ are the average video features, $\mathbf{v}_{\setminus j}$ denotes the set of all input features except that the j-th neuron is deactivated (set to zero) and $[f(\cdot)]_i$ is the i-th element of the prediction $f(\cdot)$.

The purpose of Eq. 9 is to find a neuron which contributes most to predicting a topic occurred in the video. We can consider that the identified neuron j_i^* "prefers" topic i, which can then represent the learned features of j_i^*. After we go through all the videos in the training set, we can find one or more neurons associated with each topic. Note that previous work has shown that a neuron may respond to different facets [25], which is also true in our case, that is, a neuron may prefer different topics. Here, we only choose one for simplicity to represent the learned features of the given neuron.

3. Experimental Results

3.1. Experimental Settings

Dataset: We use the YouTubeClips [8] dataset, which is well suited for training and evaluating an automatic video captioning model. The dataset has $1,970$ YouTube clips with around 40 English descriptions per video. The video clips are open-domain, containing a wide range of daily subjects like sports, animals, actions, scenarios, etc. Following [32], we use $1,200$ video clips for training, 100 video clips for validation and 670 video clips for testing.

Training: In the attentive encoder-decoder framework, we select $n = 28$ equally sampled frames in each video and feed each frame into GoogLeNet [30] to extract a 1024 dimensional frame-wise representation from the $pool5/7 \times 7\_s1$ layer. The parameters of GoogLeNet are fixed during training.

The overall objective function for caption generation is

$$L = \frac{1}{N}\sum_{k=1}^{N}\Big(\lambda\|f(\mathbf{v}^k) - \mathbf{s}^k\|_2^2 - \sum_{t=1}^{N_s^k}\log p(\mathbf{y}_t^k|\mathbf{y}_{<t}^k, \mathbf{x}^k)\Big),$$

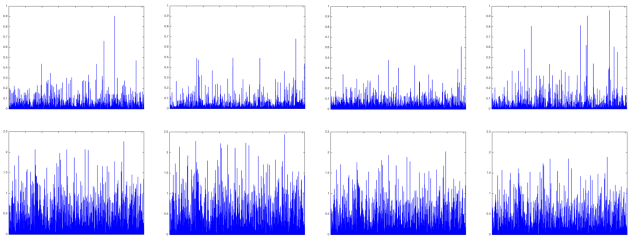
where there are N training video-description pairs $(\mathbf{x}^k, \mathbf{y}^k)$. $\mathbf{v}^k = \frac{1}{n}\sum_{i=1}^{n}\mathbf{v}_i^k$ are the average video features and $\mathbf{s}^k$ is the topic representation for video $\mathbf{x}^k$.

We use Adadelta [36] to jointly estimate the model parameters for bi-directional LSTM of the encoder, attentive LSTM of the decoder and two-layer perception of f. After training, we apply the prediction difference maximization algorithm to interpret the learned features of each neuron.

Baseline: In the experiment, we call our model LSTM-I which jointly models the interpretability of the learned features and video captioning. The hyperparameter λ is set to 0.1 by maximizing the performance on the validation set. To compare the results, we also test a baseline model named LSTM-B without interpretive loss—it is only optimized with respect to the sum of negative log-likelihood over the words. These two models have the same encoder-decoder architecture.

3.2. Feature Visualization

We visualize the learned representations of test videos in Fig. 3. The top and bottom rows show the results of LSTM-I and LSTM-B, respectively. We randomly choose some topics, and for each topic, we find a subset of videos containing this topic and plot the neuron activations by averaging the features from these videos. It can be seen that the entries of LSTM-I representations are very peaky at some specific neurons, indicating a strong correspondence between topics and neurons. Therefore we can rely on the prediction difference maximization algorithm to represent the learned features of neurons by the corresponding semantic topics. The interpretability of the learned features in LSTM-I is



(a) (b) (c) (d)

Figure 3. Examples of learned video representations using LSTM-I model (top) and LSTM-B model (bottom). Each histogram indicates an average of activations of a subset of videos, which have the same topic. (a) representations for topic "dog"; (b) representations for topic "girl"; (c) representations for topic "walk"; (d) representations for topic "dance". The top words for topics are shown in Table. 1.

better than that in LSTM-B because the correspondence between neurons and topics decouples the interaction of different neurons, which makes human users easily understand and manipulate video features (See Section. 4).

To further visualize the variation of the learned features for a video through time, we randomly choose some videos for presentation as shown in Fig. 4. For each video, we select some relevant topics existing in the descriptions and some irrelevant topics which are unimportant or easily confusing[1]. We have named the topics according to their high-probability words (See Table. 1 for the top words with respect to topics). We plot the activations of one neuron associated with each topic. We can see that the neurons associated with the salient topics in videos have high activations through time, which suggests the strong compliance between neuron activations and video contents.

It should be noted that the relevant topics are mapped from average video features in Eq. 8, so the associated neurons may not be activated in every frame. For example, in Fig. 4 (c), the neuron with respect to topic "horse" is not activated in the first frame, but the average video features can be mapped to predict topic "horse" correctly. The fact is also true in Fig. 4 (d), where the neuron with respect to topic "woman" is only activated in the third frame. It proves that the neurons are able to detect corresponding topics when they appear without severe overfitting.

3.3. Performance Comparison

To validate whether the interpretability of the learned features will affect the task performance, we test the quality of the generated sentences measured by BLEU [27] and METEOR [11] scores, which compute the similarity between a hypothesis and a set of references. In the first block of Table. 2, we compare the performance with the baseline model. We can see that LSTM-I significantly outperforms

[1]A video demo is available at http://ml.cs.tsinghua.edu.cn/~yinpeng/papers/demo-cvpr17.mp4

Model	BLEU	METEOR
LSTM-B (GoogLeNet)	0.416	0.295
LSTM-I (GoogLeNet)	**0.446**	**0.297**
LSTM-YT (AlexNet) [32]	0.333	0.291
S2VT (RGB + Optical Flow) [31]	-	0.298
SA (GoogLeNet) [34]	0.403	0.290
LSTM-E (VGG) [26]	0.402	0.295
h-RNN (VGG) [35]	0.443	**0.311**
SA (GoogLeNet + C3D) [34]	0.419	0.296
LSTM-E (VGG + C3D) [26]	0.453	0.310
h-RNN (VGG + C3D) [35]	0.499	0.326

Table 2. BLEU and METEOR scores comparing with the state-of-the-art results of description generation on YouTubeClips dataset.

the baseline LSTM-B in BLEU score and achieves better performance in METEOR score. These results suggest that our model benefits from the proper way of incorporating external semantic information into the training process, which makes the features capture more useful temporal information (*e.g.*, actions, relationships) and thus generates more accurate descriptions. So the proposed LSTM-I with better interpretability also helps to improve the performance for captioning.

To fully evaluate the performance on the video captioning task, we compare our approach with five state-of-the-art methods, namely, LSTM-YT [32], S2VT [31], SA [34], LSTM-E [26], and h-RNN [35]. LSTM-YT translates videos to descriptions with a single network using mean pooling of AlexNet [18] features over frames; S2VT directly maps a sequence of frames to a sequence of words; SA incorporates a soft attention mechanism into the encoder-decoder framework; LSTM-E considers the relationship between the semantics of the entire sentence and video content by embedding visual-semantic; and h-RNN exploits the temporal dependency among sentences in a paragraph by a hierarchical-RNN framework.

We only use 2-D CNN features for simplicity in this work. For fair comparison, we also show the extensive results of SA, LSTM-E and h-RNN which only incorporate 2-D CNN features. By comparing our baseline method LSTM-B to SA, which only uses a single CNN model as encoder and a similar decoder architecture, we can see that our baseline model achieves higher BLEU and METEOR scores, suggesting that the bi-directional LSTM can help us capture temporal variation and lead to better video representations. On the other hand, our LSTM-I achieves state-of-the-art performance and gets higher BLEU score than other methods. These results further demonstrate the effectiveness of our method, and we can conclude that integrating the interpretability of latent features by leveraging semantic information during training is a feasible way to simultaneously improve interpretability and achieve better performance.

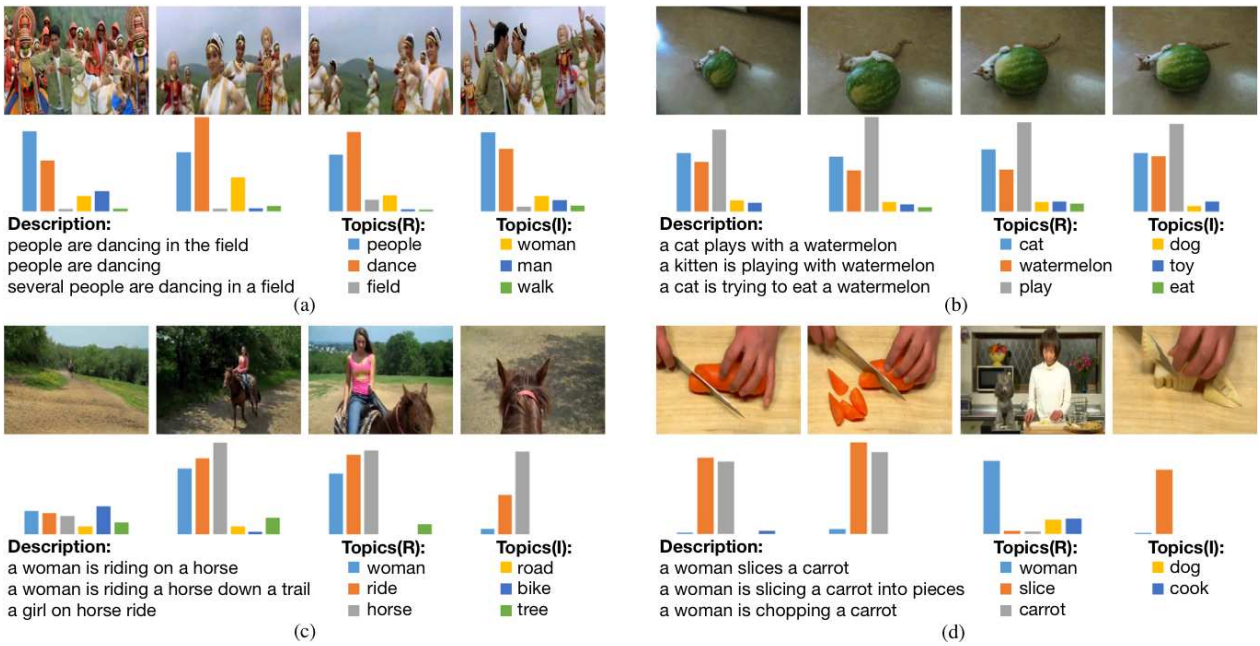


Figure 4. Neuron activations with respect to relevant and irrelevant topics in sampled videos. Topics(R) are relevant topics extracted from video descriptions. Topics(I) are irrelevant topics which are unimportant or easily confusing topics. We plot the activations of one neuron related to each topic through time.

Model	LSTM-B	LSTM-I	LSTM-R
Accuracy(%)	88.50	**91.13**	90.13

Table 3. Video action recognition performance of different models.

3.4. Video Action Recognition

We also demonstrate the generalization ability of the learned interpretable features. Specifically, interpretable features usually contain more general information than the features learned by optimizing a task-specific objective, because task-specific features may overfit the particular dataset, while interpretable features reach a good balance between task-specific fitting and generalization. We test this hypothesis by examining a transfer learning task—we evaluate the performance on video action recognition by transferring the learned features from video captioning.

We use the UCF11 dataset [22], a YouTube action dataset consisting of 1600 videos and 11 actions, including basketball shooting, biking, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking. Each video has only one action associated with it. We randomly choose 800 videos for training and 800 videos for testing. We use the same encoder architecture as in the video captioning task. The decoder is a two-layer perception and we minimize the cross-entropy loss.

Table. 3 presents the results, where we adopt three model variants. We do not compare with other state-of-the-art action recognition models due to the lack of standard train-test splits. In LSTM-B and LSTM-I, the parameters of the encoder are fixed, which are initialized with the trained captioning model. We can consider that the features for action recognition are extracted from the trained captioning encoder. We only optimize the parameters of the two-layer decoder. LSTM-R has the same architecture but all the parameters are initialized randomly and then optimized.

We can see that LSTM-I achieves higher accuracy than LSTM-B, which verifies our hypothesis that the interpretable features contain more general information than the task-specific features and lead to higher performance in other tasks. Another fact is that LSTM-I outperforms LSTM-R, which indicates that the interpretable features learned by captioning task are more effective than the features learned by action recognition. This is because that the interpretive loss makes the neurons learn to detect semantic attributes in LSTM-I model and these features are general and transferable for other tasks.

4. Human-in-the-loop Learning

An important advantage of the interpretable features is that they provide a natural interface to get human users involved in the learning process, and make them understand how the system works, what is going wrong and how to correct errors (if any). Although previous works [37, 23] have provided some applications on human interaction and architecture selection, they still need expert-level users to join the procedure because non-expert users get little insight about potential problems. They also lack a human-in-the-loop learning procedure, which helps models inte-

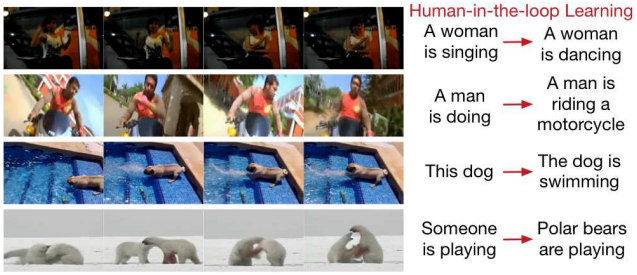


Figure 5. We show the second half (unseen part) of four videos and the predicted captions before and after refining the model. By providing the missing topics ("dance", "motorcycle", "swim" and "polar bear") for the first half of these four videos and refining the model, the predictions for the second half are more accurate.

grate human knowledge into the training process to refine their shortcomings. Here, we present an easy way to allow a human-in-the-loop learning procedure by clearly understanding the learned features without requiring expert-level experience of human users. In our case, when the model outputs an inaccurate description, human users only need to provide the missing topics. The human-in-the-loop learning procedure can diagnose potential problems in the model and refine the architecture, so the similar errors will never occur in future unseen data.

Specifically, the human-in-the-loop learning procedure can be divided into two steps—activation enhancement and correction propagation. First, when it receives a topic t from human users for an inaccurate output, it retrieves a set of neurons associated with t, which are already found by the prediction difference maximization algorithm. For these neurons, it adds the average activations of them in a subset of training videos containing topic t and turns the original features $\mathbf{v}$ to $\mathbf{v}^*$. The purpose of activation enhancement is to let the neurons associated with the missing topic have higher activations, so the decoder propobably maps the new features to more accurate descriptions. Second, to generalize the specific error to future unseen data, we use the correction propagation to fine-tune the parameters of the encoder. We expect to let the encoder learn to generate $\mathbf{v}^*$ instead of $\mathbf{v}$, so we minimize

$$L_{human} = \|\mathbf{v}' - \mathbf{v}^*\|_2^2 + \mu\|\theta' - \theta\|_2^2, \qquad (10)$$

where $\mathbf{v}'$ are the outputs of the refined encoder, θ and θ' are the parameters of the original and the refined encoders, respectively. The first term aims to let features $\mathbf{v}'$ approximate the optimal features $\mathbf{v}^*$, and the second term forces the model to have little variations. The balancing weight μ makes the refined model not overfit to this sample.

In our experiments, it's hard to find a similar error occurred in two videos in the test set due to its small size and rich diversity. So we find 20 videos with inaccurate predictions in the test set and split each of them into two parts. We optimize Eq. 10 using the first half of each video and use the second half as future unseen data to test. We get more accurate captions for 17 videos (second half), which can capture the missing topics. Fig. 5 shows four cases. Taking the polar bears video for example, the model doesn't capture the salient object "polar bear" for every parts of the video. By providing the missing topic and refining the model using the first half, the model can accurately capture "polar bear" and produce more accurate captions for the second half. It proves that the model learns to solve its potential problems with the aid of human users.

We also examine whether this procedure could affect the overall performance. We test the performance of the new model after refining for these 20 videos in turn. We get BLEU score 0.449 and METEOR score 0.298, which are sightly better than those in Table. 2 to prove that refined model does not affect the captioning ability while it makes more accurate predictions for the selected 20 videos.

5. Conclusions

In this work, we propose a novel technique to improve the interpretability of deep neural networks by leveraging human descriptions. We base our technique on the challenging video captioning task. In order to simultaneously improve the interpretability of the learned features and achieve high performance, we extract semantically meaningful topics from the corpus and introduce an interpretive loss during the training process. To interpret the learned features in DNNs, we propose a prediction difference maximization algorithm to represent the learned features of each neuron by a topic. We also demonstrate a human-in-the-loop training procedure which allows humans to revise false predictions and help to refine the network.

Experimental results show that our method achieves better performance than various competitors in video captioning. The learned features in our method are more interpretable than opaque models, which can be transferred to video action recognition. Several examples prove the effectiveness of our human-in-the-loop learning procedure.

References

[1] S. D. Ali. The impact of deep learning on investments: Exploring the implications one at a time. *Predictive Analytics and Futurism*, 13:49–50, 2016. 1

[2] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016. 3

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. 4

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 1

[6] J. L. Boyd-Graber, D. M. Blei, and X. Zhu. A topic model for word sense disambiguation. In *EMNLP-CoNLL*, 2007. 4

[7] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In *ICCV*, 2007. 4

[8] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 5

[9] J. Chen, K. Li, J. Zhu, and W. Chen. Warplda: a simple and efficient o (1) algorithm for latent dirichlet allocation. In *VLDB*, 2016. 4

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Workshop on Deep Learning*, 2014. 2

[11] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *ACL Workshop on Statistical Machine Translation*, 2014. 6

[12] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341, 2009. 2, 5

[13] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 4

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[15] H. Greenspan, B. van Ginneken, and R. M. Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016. 1

[16] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015. 1

[17] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. In *ICLR Workshop*, 2016. 2

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 6

[19] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[20] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):102–114, 2017. 3

[21] A.-A. Liu, N. Xu, W.-Z. Nie, Y.-T. Su, Y. Wong, and M. Kankanhalli. Benchmarking a multimodal and multiview and interactive dataset for human action recognition. *IEEE Transactions on Cybernetics*, 2016. 3

[22] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009. 7

[23] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. In *VAST*, 2016. 7

[24] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015. 2

[25] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. In *ICML Visualization Workshop*, 2016. 2, 5

[26] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 3, 6

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6

[28] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1

[29] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. 3

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1, 5

[31] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *ICCV*, 2015. 6

[32] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT*, 2015. 2, 3, 5, 6

[33] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1, 4

[34] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015. 1, 3, 4, 6

[35] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016. 3, 6

[36] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 5

[37] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 1, 2, 7

[38] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015. 2

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 2

[40] J. Zhu, A. Ahmed, and E. P. Xing. Medlda: maximum margin supervised topic models. *Journal of Machine Learning Research*, 13(Aug):2237–2278, 2012. 4

[41] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017. 5