ViCom: Benchmark and Methods for Video Comprehension

Du Tran\textsuperscript{1,2}, Manohar Paluri\textsuperscript{2}, Lorenzo Torresani\textsuperscript{1}
\textsuperscript{1}Dartmouth College, \textsuperscript{2}Facebook AI Research
{trandu,mano}@fb.com, lt@dartmouth.edu

Abstract

There is a widespread agreement that future technology for organizing, browsing and searching videos hinges on the development of methods for high-level semantic understanding of video. But, so far the community has not reached to a consensus on the best way to train and assess models for this task. Casting video understanding as a form of action or event categorization is unsatisfying as it is not clear what the semantic classes or abstractions of this domain should be. Language has been exploited to sidestep the problem of defining abstract video categories by formulating video understanding as the task of captioning or description. However, language is highly complex, redundant and sometimes ambiguous. Many different captions may express the same semantic concept. To account for this ambiguity, quantitative evaluation of video description requires sophisticated metrics, whose performance scores are typically hard to interpret by humans.

This paper provides four contributions on this problem. First, we formulate Video Comprehension as a new well-defined task with an easy-to-interpret performance measure. Second, we describe a general semi-automatic procedure to create benchmarks for this task. Third, we publicly release a large-scale video benchmark created with an implementation of this procedure and we include a human study that assesses human performance on our dataset. Finally, we propose and test a varied collection of approaches on this benchmark for the purpose of gaining a better understanding of the new challenges posed by video comprehension.

1 Introduction

Over the last few years deep learning has revolutionized the field of still-image analysis by producing breakthrough results in several domains including object categorization \cite{15, 20}, detection \cite{9}, scene classification \cite{31}, and semantic segmentation \cite{17}. While there has been widespread expectation that these performance improvements will naturally extend to the video domain, the results so far have been lagging compared to the image setting.

We argue that progress in this field has been held back primarily by the small-scale of existing labeled video datasets and by the low-quality of the annotations available to train machine learning models. Deep architectures have extensive learning capacity but require large-scale training sets (containing millions of examples) in order to learn effectively. Conversely, they are extremely prone to overfitting and poor performance when trained on small datasets. Unfortunately, today even the largest video analysis benchmarks, such as UCF101 \cite{24} and Sports-1M \cite{11}, are too small in size to enable effective learning of deep models. Furthermore, the labels manually collected on these datasets merely specify the class of the action in each video (e.g., \textit{walking} or \textit{sitting}) but do not indicate where the action is performed. Thus, the learning algorithm is left with the burden of discovering on its own the portion of the video that is truly representative of the action. Moreover, these datasets are quite limited in semantic scope, as they include only a small number of action categories (e.g., 101 classes for UCF101 and just sport activities for the case of Sports-1M). Therefore, features learned from such datasets are unlikely to perform well on videos containing more general, everyday actions. Finally, one can argue that understanding a video is much more than recognizing the actions contained therein: it also requires understanding the scene context (office vs home), the objects (book vs laptop), the interactions between the subjects (discussing vs arguing) and much more.
For these reasons, several authors have proposed to abandon the classic view of video understanding as a form of action categorization and they suggested to reformulate the task as a description or captioning problem, where the objective is to generate a sentence summarizing the input video. On one hand, this makes the output directly readable by human subjects, which is desirable for application scenarios such as video browsing or searching. The downside however is that the outputs produced by such methods are hard to evaluate quantitatively. This happens because for each input there is not a single correct output, as many different sentences can reasonably describe a given video. To address this ambiguity, one can resort to comparative evaluation by human judges \[6\]. But this would require a huge crowdsourcing effort for every new algorithm to assess. Another approach is to design sophisticated metrics (e.g., METEOR or BLUE) that can capture the similarities of captions expressing the same semantic concept. However, it is hard for humans to interpret the meaning of these scores, e.g., is a METEOR score of 28\% representing an acceptable captioning performance, or how big a difference in these scores would lead to a noticeable difference in the predictions?

In this paper we propose to cast video understanding in the form of multiple choice tests that assess the ability of the algorithm to comprehend the semantics of the video. Figure[1] illustrates an example of video comprehension. The algorithm is presented with an input video and \(k\) possible descriptions. Only one of them represents the correct caption for the video. The task is well-posed as a traditional classification problem, with performance numbers easy to interpret (e.g., random chance produces an accuracy of \(1/k\). Yet, our classification task does not entail the definition of arbitrary video classes or action categories. Furthermore, we describe a procedure to construct multiple-choice tests for video comprehension with very little human intervention. This makes it possible to generate large-scale benchmarks for training and testing deep models on this task. Using this procedure we built a dataset that we will release to the research community. Although this is only the first version of our benchmark, it has already size comparable with the largest existing datasets for video analysis. Furthermore, in the supplementary material we discuss our plan to scale up considerably the size of this benchmark within the next year since it involves very little human intervention. Finally, in this paper we also present preliminary results achieved with three distinct approaches to video comprehension, respectively based on the methodologies of regression, metric learning and video captioning. Perhaps surprisingly, video captioning yields the worst results, which suggests that models trained to generate description are not effective for video comprehension. Conversely, we demonstrate that metric learning methods produce by far the best accuracy on this task. In summary, the contributions of our work are four-fold:

- We propose a new high-level video understanding task which is well-posed and easy to evaluate.
- We describe a general semi-automatic procedure to construct benchmarks for video comprehension (section 3.2). The procedure requires a limited set of manual annotations, independent of the size of the dataset. This renders our approach applicable to build benchmarks of unprecedented scale.
- We present an implementation of this procedure, which we used to create a video comprehension benchmark of size comparable to the biggest existing datasets (section 3.3).
- We introduce and assess a varied set of baselines and methods to tackle the problem of video comprehension on our benchmark (section 4).

2 Related Work

Video understanding has been studied for many years. Early approaches focused on action recognition \[2,7,24\], event detection \[12\], irregularity detection \[4\], action similarity labeling \[14\]. Most of these methods rely on hand-crafted features \[16,30\] and train machine learning models on top of these representations. Recent advances in deep learning have opened up the possibility of learning models from raw videos. Simonyan and Zisserman introduced a two-stream network that achieved strong results on action recognition \[22\]. Tran et. al. proposed to use 3D ConvNets to learn spatiotemporal...
features from a large-scale dataset [27]. Despite their good performance on action categorization, these approaches are by design limited to predict a single label per video and thus are not really addressing a semantic understanding of the video.

Inspired by recent promising results on image captioning [6], different approaches have been proposed for video description [26, 29]. These methods are based on recurrent neural networks (e.g., LSTM) and are trained to predict a single sentence to describe the input video. This area shows good promise for developing algorithms that can understand and describe videos in a human-readable language. However, captioning is very hard to assess. This limitation makes it hard to compare competing algorithms, even when resorting to human judges. Visual question and answer (QA) was also recently introduced for both images [11] and videos [25]. Compared to captioning, Visual QA is better-posed, as the problem is conditioned on the question being asked. However, in the free-form QA setting, there are still multiple correct answers that can be correctly applied to a single question. Moreover, collecting ground truth annotations for QA is very expensive. This makes it hard to build large-scale datasets on this task. Our video comprehension problem shares similarities with video captioning and QA, as our task also assesses algorithms on their ability to understand the semantics of the video. However, our task is different in its formulation: it entails selecting one of the \( k \) possible sentences from a multiple choice test, rather than asking to describe or to answer a particular question. This renders the quantitative evaluation easy to carry out and makes performance scores very intuitive. Our work relates also to [8, 11] in terms of the aim at building large-scale video datasets. However, our dataset is purposely constructed for video comprehension while the aforementioned ones are for classification and detection.

3 Video Comprehension

3.1 Problem statement

Given an input video clip \( V \) and a set of \( k \) English sentences \( s_1, s_2, \ldots, s_k \), the problem of video comprehension is to predict which of these \( k \) choices best describes the visual content of the input clip. Note that readable text in the frames is automatically blurred and audio is removed. A concrete example of video comprehension is provided in Figure 1. We argue that for a system to do well on this task it must be able to infer the true semantics of the video, including context, the nature of the interactions among the subjects, and the objects appearing in the scene. Thus, we believe this to be a more fitting assessment of video comprehension by machines than the tasks of action recognition or video captioning, explored in prior benchmarks.

3.2 Procedure for Constructing a Video Comprehension Dataset

In order to assess and compare methods on the task of video comprehension, a dataset must be constructed to enable the training and testing of models on this problem. Here we review the desiderata that inspired the construction design of our benchmark. Ideally, the dataset must be:

1. **Large-scale.** In the still-image domain we have witnessed a dramatic revolution in methodology and breakthrough results with the introduction of a large-scale dataset. In fact, recent research has shown that the problem of overfitting and difficult optimization with deep models are vastly reduced when leveraging large datasets for training. Thus, our desired benchmark for video comprehension must be large enough to enable the training of these powerful models.

2. **Semi-automatic.** The process of dataset construction must be semi-automatic and must require little human intervention. This is a fundamental requirement in order to be build a massive collection of examples. We note that the limited scale of prior datasets in the video domain is a direct consequence of the high human cost and time consumption needed to label video clips.

3. **Semantically diverse.** As our objective is to train universal models that can comprehend video of arbitrary nature, the training and testing sets must contain a wide representation of subjects, including politics, sports, science, technology, arts, and travel.

To meet these criteria, we design a procedure that generates semi-automatically video comprehension tests (as shown in Figure 1) by leveraging an existing gigantic repository of TV news programs – the TV News Archive¹. We note that access to the Archive’s Collections is granted at no cost for scholarship and research purposes. Thus it represents a fitting platform for the construction of video benchmarks. Furthermore, as TV news cover all social, cultural, and natural aspects of modern life, the collection is inherently semantically diverse. Finally, the videos have accurate associated time-synchronized English captions providing a well-aligned textual transcription of the audio (the TV

¹https://archive.org/details/tv
the false positive rate of relevant clip detector evaluated on the validation set. At Figure 2: Relevant clip detection. The ROC curve of the false positive rate of 0.1, the true positive rate is 0.83.

We use a subset of 20 news videos (randomly selected from our original 4,990 TV videos) as a training set exclusively for the development of our relevant/irrelevant clip detector. Note that we remove these 20 videos from the collection used for dataset construction. We manually labeled all clips segmented from this set as either irrelevant (e.g. studio, advertisement, weather-forecast clips) or relevant (out-of-studio footage, such as dynamic scenes where human subjects or the camera are moving). The detector is trained on the visual component of each clip (thus, without considering textual descriptions). The details of our detector are discussed in section 3.3.

In this section we discuss in detail the construction of the dataset. Each video downloaded from the Archive is a complete TV news show from a particular channel recorded and broadcasted on a specific day (e.g. ABC News Good Morning America on August 27, 2011 from 8am to 9am) with lengths varying from 30 minutes to 2 hours. Our procedure then performs a sequence of steps aimed at generating a set of video comprehension tests from each program. The steps include clip segmentation, clip elimination, and multiple-choice test generation.

Clip segmentation. Each TV news video is segmented into short clips, corresponding to individual sentences (terminated by a period) of the closed captions. For each such sentence, using the time stamps of its beginning and ending, we segment the corresponding clip from the video. This process yields a massive number of clips from each video. In order to build a dataset of clips having fairly homogeneous length and to have enough temporal context in each clip, we eliminate clips that are shorter than 2 seconds or longer than 5 seconds. Similarly, we discard clips corresponding to sentences that are either too short (5 words or fewer) or too long (more than 60 words).

Clip elimination. This step is carried out to remove clips whose visual content is not informative. Examples include advertisement, static scenes, segments showing anchors speaking, sections inside the news studio, such as the weather forecast portion of the news program. Such clips are not useful for training general computer vision models. In order to make our dataset construction scalable, we develop a detector to automatically discard irrelevant clips. The detector is trained on a small collection of clips manually labeled as either irrelevant (e.g. studio, advertisement, weather-forecast clips) or relevant (out-of-studio footage, such as dynamic scenes where human subjects or the camera are moving). The detector is trained on the visual component of each clip (thus, without considering its CC). The details of our detector are discussed in section 3.3.

Multiple-choice test generation. Given a video clip, we form a multiple-choice test of \(k\) potential textual descriptions by including \(k - 1\) distractors and the true associated CC sentence. The distractors can be selected in different ways. The simplest solution is to randomly sample the \(k - 1\) distractors from the entire set of CC sentences. In order to make the test more challenging, one may want to select distractors that are not too distant from the correct response, according to a semantic metric over text descriptions, such as the Euclidean distance of \textsc{word2vec} vectors representing sentences.

3.3 The ViCom Dataset

In this section we discuss a specific implementation of the general procedure outlined above. This implementation was used to construct a dataset of 310,216 multiple-choice video comprehension tests, which we will make publicly available to the research community. The benchmark is split into 218,331 training examples and 91,885 test examples. We name our dataset ViCom.

The dataset is constructed from 4,990 news videos from the TV News Archive. These videos were obtained by considering 77 distinct TV news shows (BBC World News, MSNBC News Live, PBS News Hour, etc.). In order to yield a dataset with heterogenous news content, we sampled (roughly uniformly) each of this daily news shows in the period from January 1, 2009 to December 31, 2014.

We use a subset of 20 news videos (randomly selected from our original 4,990 TV videos) as a training set exclusively for the development of our relevant/irrelevant clip detector. Note that we remove these 20 videos from the collection used for dataset construction. We manually labeled all clips segmented from this set as either irrelevant (e.g. studio, advertisement, weather-forecast clips) or relevant (out-of-studio footage, such as dynamic scenes where human subjects or the camera are moving).
relevant. We represent each clip using the C3D spatiotemporal features [27], which are activations of a convolutional neural network (ConvNet) optimized for action classification. We use the activations from layer fc6. We opted for this descriptor as it has been shown by the authors to yield good performance on a variety of tasks involving semantic analysis of video. We train a simple a linear SVM on this representation to classify whether a clip is relevant or not. We evaluate this detector on our training set of 20 videos using 20-fold cross validation. The resulting ROC curve is shown in Figure[2]. The detector achieves an area under the curve (AUC) of 0.94. We use this ROC curve to choose the cutoff threshold to reject irrelevant clips. We chose the threshold corresponding to a false positive rate of 0.1, which yields a true positive rate of 0.83. This represents a good trade-off in terms of recall vs error (in other words, to retrieve 83% of the relevant clips we must cope with only 10% of irrelevant clips). Further filtering of these clips could be performed via crowdsourcing at a fairly limited financial cost. However, our experiments suggest that a 10% of irrelevant clips (i.e., clips whose visual content is not strongly correlated their associated CC sentence) in our dataset does not prevent the training of effective models for video comprehension but it offers the big benefit of a semi-automatic solution.

Applying this detector to the remaining 4970 videos yields a total of 310K clips deemed relevant for the purpose of video comprehension. We partition this dataset into training and testing splits (using a ratio of 7:3), with the additional constraint that all clips from a video are inserted in the same split (either training or testing). This is done as clips from the same video are often strongly correlated and this would bias the statistical assessment.

Table 1 compares ViCom to existing video datasets in terms of size, task, and video type. ViCom is the second biggest dataset in terms of both number of total hours and number of clips. However, while in Sports-1M and ActivityNet (the largest datasets in this comparison) each clip is labeled with an action class, each clip of ViCom is labeled with a textual description that typically provides a semantically richer annotation than an action tag. Some examples of CC sentences associated to video clips are shown in the supplementary material. Furthermore, we point out that this only represents the first version of our benchmark. Our plan is to scale the dataset to much larger sizes in the near future. The little reliance on human intervention renders this plan easy to implement.

In order to understand the distribution of subject matters represented in ViCom clips, we trained an LDA [3] model on the CC sentences of our entire training set using 10 topics. We visually inspected the most frequent words of each topic in order to manually assign a subject tag to each topic (“politics”, “economics”, “technology”, etc). The most frequent words are listed in a table included in the supplementary material. Figure[3] shows the subject distribution computed on the 310K clips of ViCom. It can be seen that the breadth of topics covered in ViCom distinguishes our dataset from prior video collections, which are much more focused in content (e.g., videos depicting only sports or movies). This makes ViCom particularly fitting for the training of models for general and comprehensive video understanding.

For each clip (in both the training and the testing split) we formed a multiple-choice test by randomly selecting 4 distractor sentences from our entire pool of CC sentences, in addition to the correct answer (the true CC). Thus, each test includes 5 sentences from which the correct one must be chosen.

4 Approaches to Video Comprehension

In this section, we consider different approaches to tackle the task of video comprehension on our ViCom dataset. For clarity, we first introduce our notation. Let us denote the training set with \(\{x^i, y^i_1, y^i_2, \ldots, y^i_k, t^i\}_{i=1..m}\), where \(x^i\) is the \(i\)-th video clip, \(y^i_1, y^i_2, \ldots, y^i_k\) are the \(k\) sentences defining the multiple choice test, and \(t^i \in [1..k]\) is the answer key, i.e., the index to the correct answer. Let \(\phi_v(x)\) be a visual embedding (i.e., a feature representation computed from pixel values) of video clip \(x\) and \(\phi_l(y)\) the language embedding of the text sentence \(y\). Examples of possible choices
for the visual embedding include aggregations of deep image features computed from individual frames of the clip (e.g., average pooling of VGG activations [23]) or deep video clip descriptors (e.g., C3D 1×6 activations [27]). The language embedding can be produced by averaging the word2vec representation [19] of all words in the sentence.

4.1 Regression
A simple strategy to video comprehension is to train a regression model \( R(x; W) \) parameterized by weights \( W \) to map from the visual embedding to the language embedding, i.e., such that \( R(x^t; W) \approx \phi_l(y^t) \). A simple instantiation of this method consists in learning a linear transformation of \( \phi_v(x) \), i.e., \( R(x^t; W) = W \phi_v(x) \), where the parameter matrix \( W \) can be estimated via least-square regression. Predictions can then be made by choosing the sentence whose language descriptor is closest to the transformed visual descriptor, i.e., \( t^* = \arg\min_{t \in \{1, \ldots, k\}} \| W \phi_v(x) - \phi_l(y^t) \|_2 \).

This proposed strategy can be made more powerful by replacing the linear regression model with a deep convolutional network \( R(x; W) \) (here \( W \) denotes weights) that is trained directly on the raw video input \( x^t \) to regress the associated CC language embedding vector \( \phi_l(y^t) \). After training, we make predictions by choosing \( t^* = \arg\min_{t \in \{1, \ldots, k\}} \| R(x; W) - \phi_l(y^t) \|_2^2 \). We explore both of these proposed models in our experiments.

4.2 Metric learning
It can be argued that the regression strategy outlined above is overly aggressive as it forces the visual vectors to be mapped into their language counterparts. This objective is difficult to realize. We can relax this desideratum by stating that the visual vector mapped to the language embedding should be closer to the correct answer than to any of the distractors. This can be achieved by learning a mapping \( M \) that projects a raw video \( x \) to the language embedding space \( \phi_l(y) \) by minimizing the triplet metric learning loss used in [21], i.e.,:

\[
W^* = \arg\min_W \sum_{i=1}^n \sum_{t \neq t^i} \left[ \| M(x^t; W) - \phi_l(y^t) \|_2^2 - \| M(x^{t^i}; W) - \phi_l(y^{t^i}) \|_2^2 + \alpha \right]_+. \tag{1}
\]

\( M(x; W) \) is a mapping with parameters \( W \). \( M(x; W) \) can be a deep ConvNet trained on raw input video \( x \). In a simpler case, it simply takes a predefined \( \phi_v(x) \) as input and learns a simple linear projection. \([\cdot]_+ \) is the hinge function. Finally, \( \alpha \) is a hyper-parameter to control the margin between the distance to the true sentence \( y^t \), and the distances to the wrong sentences \( y^{t^i} \) (with \( t^i \neq t^i \)). After training, this approach also makes predictions by searching the nearest neighbor in the language embedding space \( t^* = \arg\min_{t \in \{1, \ldots, k\}} \| M(x; W^*) - \phi_l(y^t) \|_2^2 \).

4.3 Video captioning
An alternative approach consists in generating a textual description \( y = C(x) \) by running an existing video captioning model \( C(x) \) (e.g. S2VT [29]) on the input video clip. Effectively this strategy maps the input clip \( x \) into an English sentence. The resulting textual description is then embedded into the language space to identify the closest answer: \( t^* = \arg\min_{t \in \{1, \ldots, k\}} \| \phi_l(C(x)) - \phi_l(y^t) \|_2^2 \).

5 Experiments

5.1 Experimental setup
Language embedding: In all of our experiments, we use word2vec [18] as the language embedding \( \phi_l(y) \). word2vec is a shallow neural network trained on a large-corpus to reconstruct the linguistic context of the words. It is used as a word embedding which maps words that share similar contexts into vectors that are close (in distance). We use the word2vec model provided by [18] which is pre-trained on the Google News dataset. This gives a 300-dimensional vector representation for each word. To represent \( \phi_l(y) \), we extract word2vec vectors for all words in \( y \), average these vectors, then L2-normalize the averaged vector to build a language representation for the sentence \( y \).
Visual embedding: We use different visual representations for $\phi_v(x)$ in different experiments. These representations are computed from different ConvNet architectures pre-trained on different datasets. We input our frames (or the entire clip in the case of C3D) into these pre-trained ConvNets to extract activations of a particular layer and use them as representations. We specify a visual representation by a pair of an architecture name and a layer name. We use the AlexNet [15] implemented in [10], the VGG architecture [23], and C3D [27]. The pre-trained models are provided by the authors of [10, 23, 27]. For simplicity, from now on we denote these representations as AlexNet, VGG, and C3D, respectively. It is worth noting that most example clips in ViCom have varying length (a few dozens to few hundreds frames), while AlexNet and VGG operates on frames, and C3D uses short clips of 16 frames. We average the frame (or 16-frame clip) features, and then L2-normalize the averaged vector to make visual representations for the long clips.

Regression models: We experimented with linear regression applied to C3D-fc6 and AlexNet-fc6. We call these two approaches LR-C3D and LR-Alex, respectively. We also use a 3D ConvNet as a deep regressor. We choose to use an architecture similar to that of C3D. The network layers are identical to those of C3D up to fc6. We then add a fully-connected layer with linear activation and 300 output units (effectively a linear projection). We optimize this model with a regression loss. Because this architecture is similar to C3D, we have the option either to train from scratch or to initialize the bottom layers from C3D. We name these two methods DR-C3D-0 and DR-C3D-FT.

Captioning models: We use the S2VT pre-trained model provided by [29] which is trained on MSVD [5]. S2VT is a 2-hidden-layer LSTM which takes a sequence of VGG-fc7 frame features as input and predicts a sentence. We name this approach S2VT-MS. We note that the language used in TV news is very different from the sentences in MSVD. In order to allow the captioning method to adapt to the news language, we also trained S2VT on ViCom. We name this method S2VT-Vi.

Metric learning models: We experiment with two different sets of architectures: shallow and deep networks. In the shallow network setting, we assume that we have a reasonable good visual representation, and we just learn a single fully-connected layer without nonlinear unit. We optimize this model by the triplet loss (as described in section 4.2). We test the shallow metric learning with three different representations: Alex-fc6, VGG-fc6, and C3D-fc6. We name these approaches SML-Alex, SML-VGG, and SML-C3D, respectively. We also test this shallow net applied to a combined representation of Alex-fc6, VGG-fc6, and C3D-fc6 (a simple concatenation). We name this approach SML-Com. For the deep network setting, we use again an architecture similar to C3D. We use all layers identical to those of C3D up to fc6. We then add a linear fully-connected layer with 300 output units. We can either train this network from scratch or finetune it from C3D. We name these approaches DML-C3D-0 and DML-C3D-FT, respectively.

Training settings: Both shallow and deep networks are trained using SGD with a momentum of 0.9. For shallow networks, we use a mini-batch size of 128. The initial learning rate is 0.01 and it is reduced by a factor of 0.1 every 10K iterations. Training is stopped at 60K iterations. For deep networks, we use a mini-batch size of 30. The initial learning rates are $3 \times 10^{-4}$ and $3 \times 10^{-5}$ for DML-C3D-0 and DML-C3D-FT, respectively. They are reduced by 0.1 for every 100K iterations and the training is stopped at 600K iterations. Since training deep networks is time-consuming, we choose $\alpha$ by cross validation on shallow networks. Our experiments show that using $\alpha = 0.1$ gives the best results among the tested margins of 0.01, 0.1, and 1 for all visual features features. Thus, we use $\alpha = 0.1$ in all deep metric learning networks.

5.2 Experimental results

Video comprehension: Table 2 presents the accuracy of different approaches on ViCom. Among the regression methods, LR-C3D performs the best, but is just 5.3% better than random chance. Deep regression approaches perform poorly because training regression in high-dimensional space is difficult. S2VT-MS gives a very low accuracy which is only slightly above random chance. We observe that this method generates very simple sentences because it was trained on simple description sentences of MSVD. S2VT-Vi, trained on ViCom, generates sentences more similar to those in the news reports. Figure 4 shows some generated captions of these two methods compared to the ground truth. However, the performance of S2VT-Vi is still low which indicates that video comprehension is a different task compared to video description. Shallow metric learning methods perform reasonably well on various visual representations, and SML-C3D gives the best performance (53.5%) among the shallow networks with a single representation. Combining the three visual representations boosts the accuracy to 54.4%. The deep network DML-C3D-FT gives the highest accuracy (55.5%).
Table 2: Video comprehension accuracy on ViCom. Accuracy of different approaches on ViCom compared with random chance and human performance. Methods based on deep metric learning give the best performance (the best gives 55.5%). Random chance is at 20%, while humans achieve an accuracy of 78.8%.

Human study: To better understand the challenges posed by the task, we also conducted a human study. For this study, a subset of 200 clips were randomly drawn from the test split. Each clip and its associated multiple-choice test were shown (without audio) to 5 human annotators. We asked them to select the sentence best describing the video clip out of the 5 choices. Because TV news often contain tickers with informative text, we trained a HOG-based SVM text detector to detect and blur the text tickers. We performed two human study experiments: the first one is applied on the original 200 clips while the second one is applied on the same 200 clips but with tickers detected and blurred. We denote these two experiments as Human and Human-0-ticker, respectively. We note that the 5 annotators involved in the first experiment are completely different from the 5 annotators in the second one. The first experiment answers how well humans can comprehend videos provided all information (e.g. text and visual) while the second one tells us the human-level performance on video comprehension given only visual inputs. Results from these experiments show that humans achieve an accuracy of 78.8% ± 1.35 when provided the original clips and 66.5% ± 5.6 when provided only visual inputs. This indicates that video comprehension is a hard task and that our best approach, DML-C3D-FT, is still 23.3% below human-level video comprehension.

We present in Table 3 comprehension accuracy by topic for different approaches and humans. Humans outperform machines across all topics except for “Technology” where DML-C3D-FT does better than humans. DML-C3D-FT performs reasonably well across all topics while LR-C3D and S2VT-Vi perform worse than random chance on 2 and 7 topics, respectively, out of a total of 10 topics.

6 Conclusions

In this paper we introduced a new high-level video understanding task, we presented a general procedure to construct semi-automatically benchmarks for this task, we created a dataset (ViCom) that we plan to release to the community and we evaluated a series of approaches on it. ViCom fulfills the following desirable properties: 1) it defines a well-posed task with a good quantitative evaluation metric; 2) it assesses the ability to semantically comprehend video; 3) it is large-scale, thus enabling effective training of deep models. We hope that this new task and our benchmark will become important stepping stones to fundamentally transform video analysis into higher-level video understanding. We have seen this happening in the image domain where a new large-scale benchmark [20] married with a powerful machine learning model [15] gave rise to a new generation

Table 3: Video comprehension accuracy details. Performance by topic of different methods compared with human performance. Humans achieve the highest accuracy across all topics except for ‘Technology’ where DML-C3D-FT outperforms humans. LR-C3D performs below random chance on 2 out of 10 topics, and S2VT-Vi has 7 out of 10 topic are below random chance (in underlined text).
of computer vision algorithms. We also expect that our benchmark will spur active research at the intersection between video understanding and natural language processing.

Most of our future work will be devoted to scaling up the size of ViCom. As our dataset construction is semi-automatic we believe that it will be possible to scale up ViCom quickly to a much larger benchmark with little human, computational and financial cost. We expect to increase the dataset by an order of magnitude within the next year. In order to stimulate steady progress in this area, we plan to organize a series of grand challenges built around our benchmark. We will release the ViCom dataset, all implementations and models upon publication of this article.

**Acknowledgment:** We would like to thank Trevor von Stein and Roger Macdonald from Internet Archive for providing raw videos and closed captions, Subhashini Venugopalan for providing S2VT code, pre-trained models, and helpful discussions about the S2VT parameter choices, and Facebook AI Research for their support.

### Appendices

**A ViCom development plan**

We plan to scale the ViCom dataset with one version every year. Version 1 (ViCom 2016) is a starting point with 300+ thousand clips and corresponding multiple choice options. At the time of release we will scale ViCom 2016 to 1M clips. Our focus will then be to organize a challenge and provide evaluation scripts for the community to evaluate their submissions. Depending on the adoption and excitement of the community we will host an annual challenge and/or workshop dedicated to Video Comprehension. Since, the process of scaling ViCom is less arduous compared to other video tasks (semi-automatic with little human intervention), we plan to keep scaling ViCom in the future versions.

**B ViCom topics and examples**

In the main paper we presented an experiment where we use LDA [3] to model topics of ViCom sentences using 10 topics. The top words for each topic are presented in Table 4.

| Topic       | Top words                                                      |
|-------------|----------------------------------------------------------------|
| Politics    | government, problem, country, american, war, military, protest, attack, unit, security, question, force, leader, nation, call |
| Climate     | area, city, storm, fire, hour, water, north, across, air, mile, center, snow, force, south, weather, power, rain, thousand, hit, through |
| Election    | house, republican, big, obama, romney, white, senate, democrat, vote, last, campaign, election, party, game, mitt, governor, win, night, race, poll |
| Time        | next, tonight, story, world, hour, weekend, around, few, numberth, week, ahead, show, begin, america, chuck, start, stay, york, daily |
| Technology  | san, kill, old, man, francisco, west, future, police, cover, bloomberg, shot, business, technology, pier, welcome, men, hospital |
| Legal       | case, court, call, charge, investigate, police, law, response, release, decision, office, official, former, action, death, against, depart, defense |
| Economics   | dollar, million, care, job, health, cut, tax, plan, than, paid, money, program, billion, government, american, company, announcement, raise, develop |
| Crime       | close, car, police, street, off, school, fire, video, inside, scene, show, build, last, crash, worker, open, wall, park, home, office |
| Emotion     | thank, much, little, let, join, learn, way, washington, stephanie, love, read, early |
| Miscellaneous | thing, them, lot, put, because, really, work, got, happen, did, keep, these, very, something, way, try, need, well, any |

Table 4: **ViCom topics.** Topic modeling of ViCom sentences using LDA [3] with 10 topics. The topic names are picked by authors based on the sharing semantic among of the top words.

Figure 5 presents some examples from ViCom dataset with their corresponding ground truth sentences.
Bee sting therapy is gaining popularity in parts of the world.

Pope Francis has held private talks at the Vatican with Russian President Vladimir Putin.

600 flights canceled, as much as three feet of snow with five-foot drifts.

And it kept spinning into the distance, completely out of control.

He spent two days drifting before Japanese troops were able to rescue him yesterday.

---

Figure 5: Examples from ViCom. Some example clips from the ViCom dataset with their true closed caption sentences. It can be noted that they cover a wide range of subjects ranging from environmental events, to science, politics, and accidents.

C t-SNE embedding of ViCom using different features

We compare the semantic embedding learned by DML-C3D-FT with that of C3D with respect to ViCom topics. We randomly select 10,000 clips from our ViCom test split. For visualization purpose, we only pick 5 topics. We use t-SNE [28] to project the two comparing embedding: C3D (fc6) and DML-C3D-FT (fc6) into 2-dimensional spaces. Figure 6 visualizes these results embedding. In the figure, each dot is a clip projected in 2-dimensional space and colored according to its topic label. We quantitatively verify that DML-C3D-FT (fc6) are more semantically clustered according ViCom topic labels. This indicates that DML-C3D-FT learns a meaningful metric and also explains why it performs much better than LR-C3D. For a complete view, we also present t-SNE embedding of word2vec with respect to ViCom topics.

Figure 6: ViCom topics on different embedding. Semantic embedding of word2vec, C3D (fc6), and DML-C3D-FT(fc6) on a random subset of 10,000 ViCom test clips. Each dot is a clip representation (language or visual) vector projected into 2 dimension and is labeled (colored) by the topic.

D Long sentence modeling

We study if modeling sentences in a holistic manner improves video comprehension performance. For this purpose, we use skip-thought vectors [13] as an alternative to our language embedding.
skip-thought can be considered as a whole sentence-to-vector embedding which maps a sentence to a vector of 4,800 dimensions. We use the shallow metric learning networks (SML-Alex, SML-VGG, SML-C3D, and SML-Com as described previously) with the only difference in which we replace word2vec by skip-thought. Table 5 shows the performance of our shallow metric learning networks on ViCom using skip-thought vectors compared with their corresponding networks using word2vec. This replacement consistently degrades the accuracy about 2-6% across different feature representations.

| φ_l option | SML-Alex | SML-VGG | SML-C3D | SML-Com |
|------------|----------|----------|----------|----------|
| word2vec   | 39.8     | 52.7     | 53.5     | 54.5     |
| skip-thought | 37.3     | 47.2     | 47.6     | 48.4     |

Table 5: Experiments with Skip-thought Vectors. Replacing word2vec by skip-thought vectors for the language representation consistently degrades the performance across all different visual embedding on shallow metric learning networks.

E  S2VT predictions

Additional sample predictions from S2VT-MS and S2VT-Vi are shown in Figure 7.

| S2VT-MS: A man is hitting a soccer. | S2VT-Vi: The president is expected to be the most famous state of the president and the president of the United States has a new plan to stop the violence. |
| S2VT-Vi: The man who was the first victim in the case was the only one in the world and he was the first person to be alive and he was the first person to be alive and he was in the hospital and he was on the scene of the car. | Ground truth: I heard they said we're not going to feel it until we're back home. |
| S2VT-MS: The men are dancing in the other. | S2VT-Vi: The New York City Police Department is investigating a state of emergency in the country's capital of a New York City Council who will be held in the New Year with a new report on a new report on the President's health care law. |
| S2VT-Vi: True had head of Edward Snowden 18 months ago they know who he is now his reservations of mass surveillance has given his government headaches. | Ground truth: I think he is going to shoot for that memory tomorrow. |

Figure 7: S2VT predictions. More prediction samples from S2VT. The sentences generated by S2VT-MS and S2VT-Vi, and the CC ground truth. S2VT-MS predicts simple sentences. S2VT-Vi is well adapted to news language and predicts longer sentences.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In ICCV, 2015.
[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In ICCV, pages 1395–1402, 2005.
[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993–1022, 2003.
[4] O. Boiman and M. Irani. Detecting irregularities in images and in video. IVC, 2007.
[5] D. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In ACL, pages 190–200, 2011.
[6] X. Chen, T.-Y. L. Hao Fang, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325, 2015.
[7] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In Proc. International Conference on Computer Vision, pages 726–733, 2003.
[8] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In CVPR, pages 961–970, 2015.
[9] R. Girshick. Fast r-cnn. In International Conference on Computer Vision (ICCV), 2015.
[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.
[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In CVPR, 2014.
[12] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In Proc. International Conference on Computer Vision, 2005.
[13] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In NIPS, pages 3294–3302, 2015.
[14] O. Kliper-Gross, T. Hassner, and L. Wolf. The action similarity labeling challenge. TPAMI, 2012.
[15] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.
[16] I. Laptev and T. Lindeberg. Space-time interest points. In ICCV, 2003.
[17] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In NIPS, pages 3111–3119, 2013.
[19] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In CVPR, pages 3202–3212, 2015.
[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2012.
[21] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In CVPR, pages 815–823, 2015.
[22] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In NIPS, 2014.
[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
[24] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In CRCV-TR-12-01, 2012.
[25] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In CVPR, 2016.
[26] A. Torabi, C. J. Pal, H. Larochelle, and A. C. Courville. Using descriptive video services to create a large data source for video annotation research. arXiv:1503.01070v1, 2015.
[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015.
[28] L. van der Maaten and G. Hinton. Visualizing data using t-sne. JMLR, 9(2579-2605):85, 2008.
[29] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence – video to text. In ICCV, 2015.
[30] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. IJCV, 103(1):60–79, 2013.
[31] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In NIPS, 2014.