A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering

Tegan Maharaj[1] Nicolas Ballas[2] Anna Rohrbach[3] Aaron Courville[2]
Christopher Pal[1] [1]Polytechnique Montréal [2]Université de Montréal
[3]Max-Planck-Institut für Informatik, Saarland Informatics Campus
[1]tegan.maharaj,christopher.pal@polymtl.ca [2]nicolas.ballas,aaron.courville@umontreal.ca
[3]arohrbach@mpi-inf.mpg.de

Abstract

While deep convolutional neural networks frequently approach or exceed human-level performance in benchmark tasks involving static images, extending this success to moving images is not straightforward. Video understanding is of interest for many applications, including content recommendation, prediction, summarization, event/object detection, and understanding human visual perception. However, many domains lack sufficient data to explore and perfect video models. In order to address the need for a simple, quantitative benchmark for developing and understanding video, we present MovieFIB, a fill-in-the-blank question-answering dataset with over 300,000 examples, based on descriptive video annotations for the visually impaired. In addition to presenting statistics and a description of the dataset, we perform a detailed analysis of 5 different models' predictions, and compare these with human performance. We investigate the relative importance of language, static (2D) visual features, and moving (3D) visual features; the effects of increasing dataset size, the number of frames sampled; and of vocabulary size. We illustrate that: this task is not solvable by a language model alone; our model combining 2D and 3D visual information indeed provides the best result; all models perform significantly worse than human-level. We provide human evaluation for responses given by different models and find that accuracy on the MovieFIB evaluation corresponds well with human judgment. We suggest avenues for improving video models, and hope that the MovieFIB challenge can be useful for measuring and encouraging progress in this very interesting field.

1. Introduction

Most current work investigating multimodal question answering (QA) focuses either on the natural language aspects of the problem (e.g. [41, 36]), or QA in static images (e.g. [3, 21, 43]). Our goal is to use QA to eliminate ambiguity in natural language evaluation, in order to target the benchmarking and development of video models. More specifically, we are interested in models of moving visual information which will be leveraged for a task in another modality - in this case, text-based QA. Our proposed dataset, **MovieFIB**, is used in the fill-in-the-blank track of LSMDC (Large Scale Movie Description and Understanding Challenge) [29].

Figure 1. Two examples from the training set of our fill-in-the-blank dataset.

1.1. Video understanding

A long-standing goal in computer vision research is complete understanding of visual scenes: recognizing entities, describing their attributes and relationships. In video data, this difficult task is complicated by the need to understand and remember temporal dynamics. The task of automatically translating videos containing rich and open-domain activities into natural language (or some other modality) requires tackling the above-mentioned challenges, which stand as open problems for computer vision.

A key ingredient sparking the impressive recent progress in object category recognition [17] has been the development of large scale image recognition datasets [8]. Accordingly, several large video datasets have been pro-

posed [28, 37] to address the problem of translating video to natural language problem. These datasets rely on transcriptions of Descriptive Video Services (DVS), also known as Audio Description (AD), included in movies as an aide for the visually impaired, to obtain text-based descriptions of movie scenes. DVS provides an audio narration of the most important aspects of the visual information relevant to a movie which typically consists of descriptions of human actions, gestures, scenes, and character appearance [28].

While the extraction of scene descriptions from DVS has proven to be a reliable way to automatically associate video with text based descriptions, DVS provides only one textual description per segment of video despite the fact that multiple descriptions for a given scene are often equally applicable and relevant. This is problematic from an evaluation perspective. Standard evaluation metrics used for the video to natural language translation task, such as BLEU [24], ROUGE [18], METEOR [9] or CIDEr [39], have been shown to not correlate well with human assessments when few target descriptions are available [39]. Therefore, it is questionable to rely on such automated metrics to evaluate and compare different approaches on those datasets.

1.2. Our contributions

To address the issues with evaluating video models, we propose recasting the video description problem as a more straightforward classification task by reformulating description as a fill-in-the-blank question-answering (QA) problem. Specifically, given a video and its description with one word blanked-out, our goal is to predict the missing word as illustrated in Figure 1.

Our approach to creating fill-in-the-blank questions allows them to be easily generated automatically from a collection of video descriptions; it does not require extra manual work and can therefore be scaled to a large number of queries. Through this approach we have created over 300,000 fill-in-the-blank question-answer and video pairs. The questions concern entities, actions and attributes. Answering such questions therefore implies that a model must obtain some level of understanding of the visual content of a scene, in order to be able to detect objects and people, aspects of their appearance, activities and interactions, as well as features of the general scene context of a video.

We compare performance of 7 models on MovieFIB; 5 run by us, 2 by independent works using our dataset, and additionally compare with an estimate of human performance. We have humans compare all models' responses. We discuss results, empirically demonstrate that classification accuracy on MovieFIB correlates well with human judgment, and suggest avenues for future work.

Dataset & Challenge: `sites.google.com/site/describingmovies/lsmdc-2016/download`

Code: `github.com/teganmaharaj/movieFIB`

2. Related work

2.1. Video Captioning

The problem of bridging the gap between video and natural language has attracted significant recent attention. Early models tackling video captioning such as [16, 30], focused on constrained domains with limited appearance of activities and objects in videos and depended heavily on hand-crafted video features, followed by a template-based or shallow statistical machine translation. However, recent models such as [4, 10, 40, 42] have shifted towards a more general encoder-decoder neural approach to tackle the captioning problem for open-domain videos. In such architectures, videos are usually encoded into a vector representation using a convolutional neural network, and then fed to a caption decoder usually implemented with a recurrent neural network.

The development of these encoder-decoder models has been made possible by the release of large scale datasets such as [37, 28]. In particular, [37, 28] exploit Descriptive Video Service (DVS) data to construct captioning datasets that have a large number of video clips. DVS is a type of narration designed for the visually impaired; it supplements the original dialogue and audio tracks of the movie by describing the visual content of a scene in detail, and is produced for many movies and TV shows. This type of description is very appealing for machine learning methods, because the things described tend to be those which are relevant to the plot, but they also stand alone as 'local' descriptions of events and objects with associated visual content. In [29], the authors create a dataset of 200 HD Hollywood movies split into 128,085 short (4-5 second) clips, aligned to transcribed DVS track and movie scripts. This dataset was used as the basis of the Large Scale Movie Description Challenge (LSMDC) presented in 2015 and 2016[1].

While the development of these datasets has lead to new models which can produce impressive descriptions in terms of their syntactic and semantic quality, the evaluation of such techniques is challenging [29]. Many different descriptions may be valid for a given image and as we have motivated above, commonly used metrics such as BLEU, METEOR, ROUGE-L and CIDEr have been found to correlate poorly with human judgments of description quality and utility [29].

2.2. Image and Video QA

One of the first large scale visual question-answering datasets is the visual question answering (VQA) challenge introduced in [3]. It consists of 254,721 images from the MSCOCO [19] dataset, plus imagery of cartoon-like drawings from an abstract scene dataset [46]. There are 3 questions per image for a total of 764,163 questions with 10

[1] https://sites.google.com/site/describingmovies/

ground truth answers per question. The dataset includes questions with possible responses of yes, no, or maybe as well as open-ended and free-form questions and answers provided by humans. Other work has looked at algorithmically transforming MSCOCO descriptions into question format creating the COCO-QA dataset [27]. The DAtaset for QUestion Answering on Real-world images (DAQUAR) was introduced in [21]. It was built on top of the NYU-Depth V2 dataset which consists of 1,449 RGBD images [31]. They collected 12,468 human question-answer pairs focusing on questions involving identifying 894 categories of objects, colors of objects and the number of objects in a scene. In [43], the authors take a similar approach to ours by transforming the description task into fill-in-the-blank questions about images.

Following this effort, [45] compile various video description datasets including TACoS [26], MPII-MD [28] and the TRECVID MEDTest 14 [1]. As in our work, they reformulate the descriptions into QA tasks, and use an encoder-decoder RNN architecture for examining the performance of different approaches to solve this problem. Their work differs in approach from ours; they evaluate on questions describing the past, present, and future around a clip. It also differs in that they use a multiple choice format, and thus the selection of possible answers has an important impact on model performance. To avoid these issues here we work with an open vocabulary fill-in-the-blank format for our video QA formulation.

Other recent work develops MovieQA, a dataset and evaluation based on a QA formulation for story comprehension, using both video and text resources associated with movies [36]. MovieQA is composed of 408 subtitled movies with summaries of the movie from Wikipedia, scripts obtained from the Internet Movie Script Database (IMSDb), which are available for almost half of the movies, and descriptive video service (DVS) annotations, available for 60 movies using the MPII-MD [28] annotations. The composition of MovieQA orients it heavily towards story understanding; there are 14,944 questions but only 6,462 are paired with video clips (see Table 1).

3. MovieFIB: a fill-in-the-blank question-answering dataset

In the following we describe the dataset creation process and provide some statistics and analysis.

3.1. Creating the dataset

The LSMDC 2016 description dataset [29] forms the basis of our proposed fill-in-the-blank dataset (MovieFIB) and evaluation. Our procedure to generate a fill-in-the-blank question from an annotation is simple. For each annotation, we use a pretrained maximum-entropy parser [25, 20] from the Natural Language Toolkit (NLTK) [2] to tag all words in the annotation with their part-of-speech (POS). We keep nouns, verbs, adjectives, and adverbs as candidate blanks, and filter candidates through a manually curated stop-list (see supplementary materials). Finally, we keep only words which occur ≥ 50 times in the training set.

Table 1. Comparison of statistics of the proposed MovieFIB dataset with the MovieQA[36] dataset. Number of words includes the blank for MovieFIB.

MovieQA dataset	Train	Val	Test	Total
#Movies	93	21	26	140
#Clips	4,385	1,098	1,288	6,771
Mean clip dur. (s)	201.0	198.5	211.4	202.7±216.2
#QA	4,318	886	1,258	6,462
Mean #words in Q	9.3	9.3	9.5	9.3±3.5
MovieFIB dataset	Train	Val	Test	Total
#Movies	153	12	17	180
#Clips	101,046	7,408	10,053	118,507
Mean clip dur. (s)	4.1	4.1	4.2	4.1
#QA	296,960	21,689	30,349	348,998
Mean #words in Q	9.94	9.75	8.67	9.72

3.2. Dataset statistics and analysis

The procedure described in Section 3.1 gives us 348,998 examples: an average of 3 per original LSMDC annotation. We refer to the annotation with a blank (e.g. 'She _____ her head') as the **question** sentence, and the word which fills in the blank as the **answer**. We follow the training-validation-test split of the LSMDC2016 dataset and create 296,960 training, 21,689 validation, and 30,349 test QA pairs. Validation and test sets come from movies which are disjoint from the training set. We use only the public test set, so as not to provide ground truth for the blind test set used in the captioning challenge. Some examples from the training set are shown in Figure 1, and Table 1 compares statistics of our dataset with the MovieQA dataset. For a more thorough comparison of video-text datasets, see [29]

Figure 2 is the histogram of answer counts for the training set, showing that most words occur 100-200 times, with a heavy tail of more frequent words going up to 12,541 for the most frequent word (her). For ease of viewing, we have binned the 20 most frequently-occurring words together in the last red bin. Figure 3 shows a word-cloud of the top 100 most frequently occurring words, with a list of the most frequent 20 words with their counts. In Figure 4 we examine the distribution by part-of-speech (POS) tag, showing the most frequent words for each of the categories.

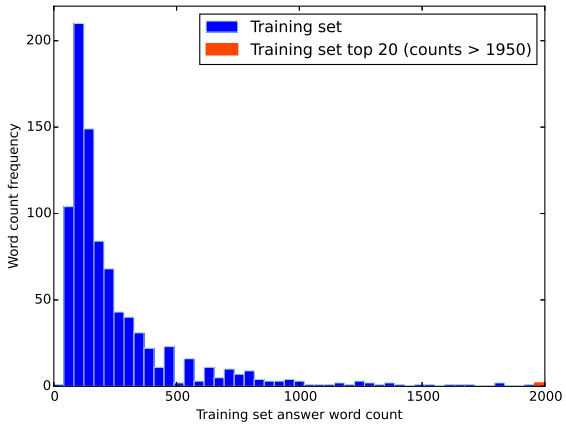

Figure 2. Histogram showing frequencies of counts for answers (blanks) in the training set. Note that the last red bin of the histogram covers the interval [1,950 : 12,541], containing the 20 most frequent words which are listed in Figure 3.

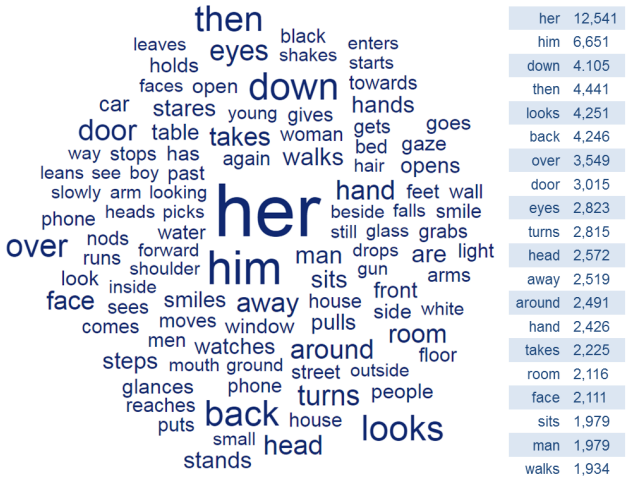

Figure 3. Word cloud showing the top 100 most frequently-occurring words in the training set answers (font size scaled by frequency) and list with counts of the 20 most frequent answers.

4. Neural framework for video fill-in-the-blank question-answering

In this section, we describe a general neural network-based approach to address the fill-in-the-blank video question-answering problem. This neural network provides a basis for all of our baseline models.

We consider a training set $(\mathbf{v}^i, \mathbf{q}^i, y^i)_{i\in(1..N)}$ with videos $\mathbf{v}^i$, questions $\mathbf{q}^i$ and their associated answers y^i. Our goal is to learn a model that predicts y^i given $\mathbf{v}^i$ and $\mathbf{q}^i$.

We first use encoder networks Φ_v and Φ_q to extract fixed-length representations from a video and a question respectively, as illustrated in Figure 5. The fixed length representations are then fed to a classifier network f that outputs a probability distribution over the different answers, $p(y \mid \mathbf{v}^i, \mathbf{q}^i)) = f(\Phi_v(\mathbf{v}^i), \Phi_q(\mathbf{q}^i))_y$. f is a single-layer MLP with a softmax. We estimate the model parameters $\boldsymbol{\theta}$ composed of the encoder and classifier networks parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_v, \boldsymbol{\theta}_q, \boldsymbol{\theta}_f\}$ by maximizing the model log-likelihood on the training set:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \log p(y^i \mid \mathbf{v}^i, \mathbf{q}^i), \boldsymbol{\theta}). \quad (1)$$

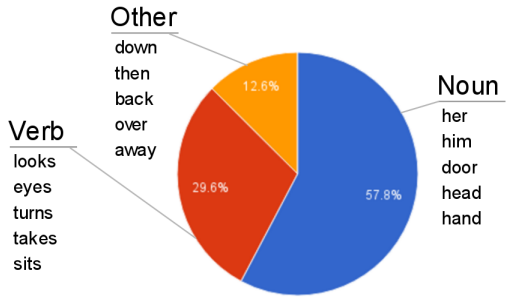

Figure 4. Pie chart showing the answer words of the training set by POS-tag supercategory (noun, verb, or other), with the five most frequent words per category.

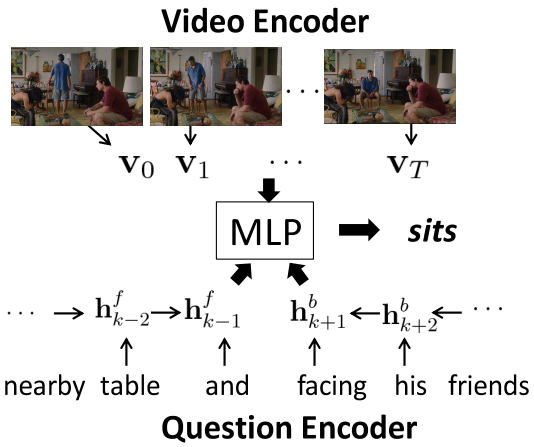

Figure 5. Fill-in-the-blank model architecture, showing the video encoder Φ_v, question encoder Φ_q, and MLP classifier network f.

4.1. Question Encoder

Recurrent neural networks have become the standard neural approach to encode text, as text data is composed of a variable-length sequence of symbols [6, 34]. Given a sequence of words $\mathbf{w}_t$ composing a question $\mathbf{q}$, we define our encoder function as $\mathbf{h}_t = \Phi_q(\mathbf{h}_{t-1}, \mathbf{w}_t)$ with $\mathbf{h}_0$ being a learned parameter. For the fill-in-the-blank task, a question $\mathbf{q}$ composed by l words can be written as $\mathbf{q} = \{\mathbf{w}_0, \ldots \mathbf{w}_{k-1}, \mathbf{b}, \mathbf{w}_{k+1}, \mathbf{w}_l\}$, where $\mathbf{b}$ is the symbol representing the blanked word. To exploit this structure, we decompose our encoder Φ_q in two recurrent networks, one forward RNN Φ_q^f applied on the sequence $\{\mathbf{w}_0, \ldots, \mathbf{w}_{k-1}\}$, and one backward RNN applied on the reverse sequence

$\{\mathbf{w}_l, \ldots, \mathbf{w}_{k+1}\}$. The forward hidden state $\mathbf{h}^f_{k-1}$ and backward hidden state $\mathbf{h}^b_{k+1}$ are concatenated and provided as input to the classifier network f. A similar network structure for fill-in-the-blank QA is explored in [22].

Forward and backward functions Φ^f_q and Φ^b_q could be implemented using vanilla RNNs, however training such models using stochastic gradient descent is notoriously difficult due to the exploding/vanishing gradients problems [5, 12]. Although solving gradient stability is fundamentally difficult [5], effects can be mitigated through architectural variations such as LSTM [13], GRU [6]. In this work, we rely on the Batch-Normalized variant of LSTM [7]:

$$\begin{pmatrix} \tilde{\mathbf{i}}_t \\ \tilde{\mathbf{f}}_t \\ \tilde{\mathbf{o}}_t \\ \tilde{\mathbf{g}}_t \end{pmatrix} = \mathrm{BN}(\mathbf{W}_w \mathbf{w}_t, \gamma_w) + \mathrm{BN}(\mathbf{W}_h \mathbf{h}_{t-1}, \gamma_h) + \mathbf{b} \quad (2)$$

where

$$\mathbf{c}_t = \sigma(\tilde{\mathbf{i}}_t) \odot \tanh(\tilde{\mathbf{g}}_t) + \sigma(\tilde{\mathbf{f}}_t) \odot \mathbf{c}_{t-1} \quad (3)$$

$$\mathbf{h}_t = \sigma(\tilde{\mathbf{o}}_t) \odot \tanh(\mathrm{BN}(\mathbf{c}_t; \gamma_c) + \mathbf{b}_c) \quad (4)$$

and where

$$\mathrm{BN}(\mathbf{x}; \gamma) = \gamma \odot \frac{\mathbf{x} - \widehat{\mathbb{E}}[\mathbf{x}]}{\sqrt{\widehat{\mathrm{Var}}[\mathbf{x}] + \epsilon}} \quad (5)$$

is the batch-normalizing transform with $\widehat{\mathbb{E}}[\mathbf{x}], \widehat{\mathrm{Var}}[\mathbf{x}]$ being the activation mean and variance estimated from the mini-batch samples. $\mathbf{W}_h \in \mathbb{R}^{d_h \times 4d_h}, \mathbf{W}_w \in \mathbb{R}^{d_w \times 4d_h}, \mathbf{b} \in \mathbb{R}^{4d_h}$ and the initial states $\mathbf{h}_0 \in \mathbb{R}^{d_h}, \mathbf{c}_0 \in \mathbb{R}^{d_h}$ are model parameters. σ is the logistic sigmoid function, $\tilde{\mathbf{i}}_t, \tilde{\mathbf{f}}_t, \tilde{\mathbf{o}}_t$, and $\tilde{\mathbf{g}}_t$ are the LSTM gates, and the $\odot$ operator denotes the Hadamard product.

4.2. Video Encoder

Following recent work on video modeling [10, 33], we use 2D (or 3D[2], as indicated) convolutional neural networks which map each frame (or sequence of frames) into a sequence vector. The video encoder Φ_v then extracts a fixed-length representation from the sequence of 2D frames composing a video. As described for the question encoder, we rely on the Batch-Normalized LSTM [7] to model the sequence of vectors.

5. Experiments and Discussion

First in Section 5.1 we describe 5 baseline models which investigate the relative importance of 2D vs. 3D features, as well as early vs. late fusion of text information (by initializing the video encoder with the question encoding and then finetuning). Using these models, we investigate the importance of various aspects of text and video preprocessing and the effects of dataset size in Section 5.2. We then describe the setup and results of getting an estimate of human performance on MovieFIB in Section 5.3. Next, Section 5.4 describes the models and results of two independent works [44, 22] which use our dataset. All of these results are summarized in Table 2. Finally, in Section 5.5 we perform a human evaluation of all of these different models' responses, and show that using the standard metric of accuracy for comparing models yields results which correlate well with human assessment.

[2]In this work, 2D = (height,width) and 3D = (height,width,time)

Table 2. Accuracy results for single models, and estimated human performance (both human experiments are conducted with a subset of 569 examples from the test set). Finetuned indicates the question encoder was initialized with the parameters of the Text-only model. Vocabulary* indicates the output softmax was reduced to only consider words with frequency ≥ 50 in the training set.

Model	Validation	Test
Text-only	33.8	34.4
GoogleNet-2D	34.1	34.9
C3D	34.0	34.5
GoogleNet-2D Finetuned	34.7	35.3
GoogleNet-2D + C3D Finetuned	35.0	35.7
Vocabulary* Text-only	34.3	35.0
Vocabulary* 2D + C3D Finetuned	35.4	36.3
Human text-only	-	30.2
Human text+video	-	68.7
VGG-2D-MergingLSTMs [22]	-	34.2
ResNet-2D-biLSTM-attn [44]	-	38.0

5.1. Comparison of baseline models

Text Preprocessing. We preprocess the questions and answers with wordpunct tokenizer from the NLTK toolbox [2]. We then lowercase all the word tokens, and end up with a vocabulary of 26,818 unique words. In Section 5.2 we analyze the impact of changing the vocabulary size.

Video Preprocessing. To leverage the video input, we investigate both 2D (static) and 3D (moving) visual features. We rely on a GoogLeNet convolutional neural network [35] that has been pretrained on ImageNet [8] to extract static features. Features are extracted from the pool5/7x7 layer. 3D moving features are extracted using the C3D model [38], pretrained on Sports-1M [14]. We apply the C3D on chunks of 16 consecutive frames in a video and retrieve the "fc7"-layer activations. We do not finetune the 2D and 3D CNN parameters during training.

To reduce memory and computational requirements, we only consider a fixed number of frames/temporal segments from the different videos. Unless otherwise specified, we consider 25 frames/temporal segments per video. Those frames/temporal segments are sampled randomly during

Figure 6. Qualitative examples for the Text-only, 2D (GoogleNet-2D), and 3D (Googlenet-2D+C3D) showing the importance of visual information; in particular the importance of 3D features in recognizing actions.

training while being equally spaced during inference on the validation and test sets. We investigate the effects of sampling different numbers of frames in Section 5.2.

Language, static visual (2D), and moving visual (3D) information. We test model variations for video fill-in-the-blank task based on the framework described in section 4. Specifically, we investigate the performance a baseline model using only the question encoder (i.e. a language model), which we call **Text-only** and the impact of 2D and 3D features individually as well as their combination. We train our baseline models using stochastic gradient descent with Adam update rules [15]. Model hyperparameters can be found in the supplementary materials. Results are reported in Table 2.

While the Text-only baseline obtains reasonable results by itself, adding a visual input in any form (2D, 3D, or combination) improves accuracy. We observe that the contributions of the different visual features seems complimentary, as they can be combined to further improve performance. To illustrate this qualitatively, in Figure 6 we show two examples which the Text-only model gets wrong, but which GoogleNet-2D+C3D model gets right. Whereas MovieQA authors find that adding video information actually hurts performance [36], our experiments demonstrate the utility of our dataset for targeting video understanding, compared to MovieQA which targets story understanding.

We also compare models with parameters initialized randomly versus model having the question encoder parameters initialized directly from the Text-only baseline, which we refer to as **Finetuned** in Table 2). Finetuned initialization leads to better results; we empirically observe that it tends to reduce the model overfitting.

5.2. Effects of amount and preprocessing of data

Vocabulary size We first look at the impact of the input vocabulary size. In addition to the text preprocessing described in Section 5.1, we eliminate rare tokens from the vocabulary applied on the input question, to only consider words that occur more than 3 times in the training set. Rare words are replaced with an "unknown" token. This leads to vocabulary of size $18,663$. We also reduce the vocabulary size of the output softmax for answer words, considering only words present more than 50 times in the training set, resulting in a vocabulary of size $3,994$. We denote this variant as "Vocabulary*" in Table 2 and observe that reducing the vocabulary sizes results in improved performance, highlighting the importance of the text preprocessing.

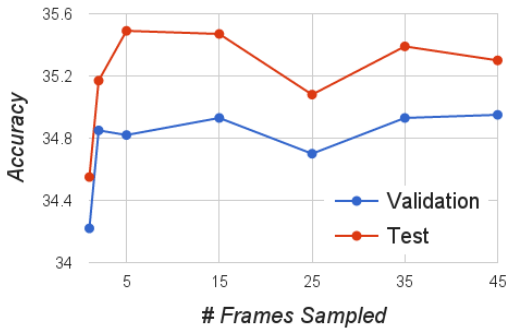

Figure 7. Performance on the test set for GoogleNet-2D (fine-tuned) showing that comparable performance is achieved with only two sampled frames.

Number of input frames We also investigate the importance of the number of input frames for the GoogleNet-2D baseline model. Results are reported in Figure 7. We observe that the validation performance saturates quickly, as we almost reach the best performance with only 2 sampled frames from the videos on the validation set.

Effects of increasing dataset size As evidenced by performance on large datasets like ImageNet, the amount of training data available can be a huge factor in the success of deep learning models. We are interested to know if the dataset size is an important factor in the performance of video models, and specifically, if we should expect to see an increase in the performance of existing models simply by increasing the amount of training data available.

Figure 8 reports the validation and test accuracies of Text-only and GoogLeNet-2D+C3D baselines as we increase the number of training videos. It shows that at 10% of training data (9,511 videos), Text-only and video models perform very similarly (20.7% accuracy for Text-only versus 21.0% for GoogleNet-2D+C3D on the valid set). It suggests that at 10% of training data, there are not enough video examples for the model to leverage useful information that generalizes to unseen examples from the visual input.

However, we observe that increasing the amount of training data benefit more to the video-based model relatively to the Text-only model. As data increases the performance of the video model increases more rapidly than the Text-only model. This suggests that existing video models are in fact able to gain some generalization from the visual input given

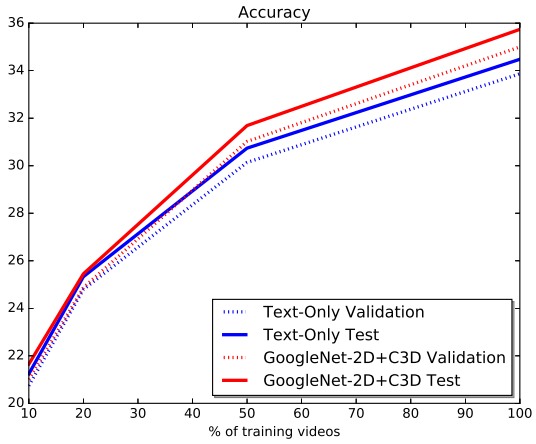

Figure 8. Fill-in-the-blank accuracy results for the Text-only and GoogleNet-2D+C3D-finetuned models on validation and test sets, trained on varying percentages (10,20,50, and 100%) of the training data, showing a larger gain in test performance relative to validation for the video model (Note that results for models trained with 100% of training data are the same as reported in Table 2).

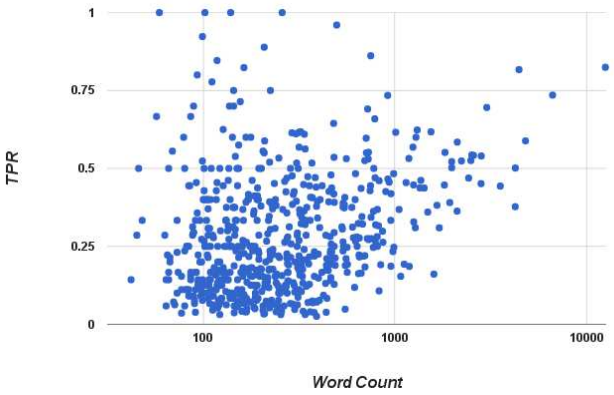

Figure 9. The true positive rate (TPR) per answer word for the GoogleNet-2D+C3D-5frame model, plotted by answer word frequency in the training set (note log scale), showing that the TRP (aka recall, sensitivity) is correlated with answer word frequency.

enough training examples. Hence, Figure 8 highlights that further increasing the dataset size should be more beneficial for the video-based models.

Figure 9 shows that per-word true positive rate (TPR) is highly correlated with answer prevalence in the training set, indicating that increasing the number of examples for each target would likely also increase performance. We plot here the results only for GoogleNet-2D + C3D for brevity, but similar correlations are seen for all models, which can be viewed in the supplementary material.

5.3. Human performance on the test set

Table 2 also reports human performance on a subset of the test set. In order to obtain an estimate for human performance on the test set, we use Amazon Mechanical Turk to employ humans to answer a sample of 569 test questions, which is representative of the test set at a confidence of $95\%/^{+}_{-}4$. To mimic the information given to a neural network model, we require humans to fill in the blank using words from a predefined vocabulary in a searchable dropdown menu. In order to ensure quality of responses, we follow [8] in having 3 humans answer each question. If two or more humans answer the same for a given question, we take that as the answer; if all disagree, we randomly choose one response as the answer out of the 3 candidates.

We perform two experiments with this setup: **Human text-only** and **Human text+video**. In the text-only experiment, workers are only shown the question, and not the video clip, while in the text+video setting workers are given both the video clip and the question. As in the automated models, we observe that adding video input drastically improves human performance. This confirms that visual information is of prime importance for solving this task.

We also observe in Table 2 that there is a significant gap between our best automated model and the best human performance (on text+video), leaving room for improvement of future video models. Interestingly, we notice that our text-only model outperforms the human text-only accuracy. Descriptive Video Service (DVS) annotations are written by movie industry professionals, and have a certain linguistic style which appears to induce some statistical regularities in the text data. Our text-only baseline, directly trained on DVS data, is able to exploit these statistical regularities, while a Mechanical Turk worker who is not familiar with the DVS style of writing may miss them.

5.4. Related works using MovieFIB

We have made the MovieFIB dataset publicly available, and two recent works have made use of it. We include the results of these models in our comparisons, and report the best single-model performance of these model in 2.

In [44], the authors use an LSTM on pretrained ImageNet features from layer conv5b of a ResNet [11] to encode the video, with temporal attention on frames, and a bidirectional LSTM with semantic attention for encoding the question. We refer to this model as **ResNet-2D-biLSTM-attn**, and it achieves the highest reported accuracy on our dataset so far - 38.0% accuracy for a single model, and 40.7 for an ensemble.

In [22], the authors use a model similar to our baseline, encoding video with an LSTM on pretrained VGG [32] features, combined with the output of two LSTMs running in opposite directions on the question by an MLP. We refer to this model as **VGG-2D-MergingLSTMs**. Their method differs from ours in that they first train a Word2Vec [23] embedding space for the questions. Like us, they find that using a pretrained question encoding improves performance.

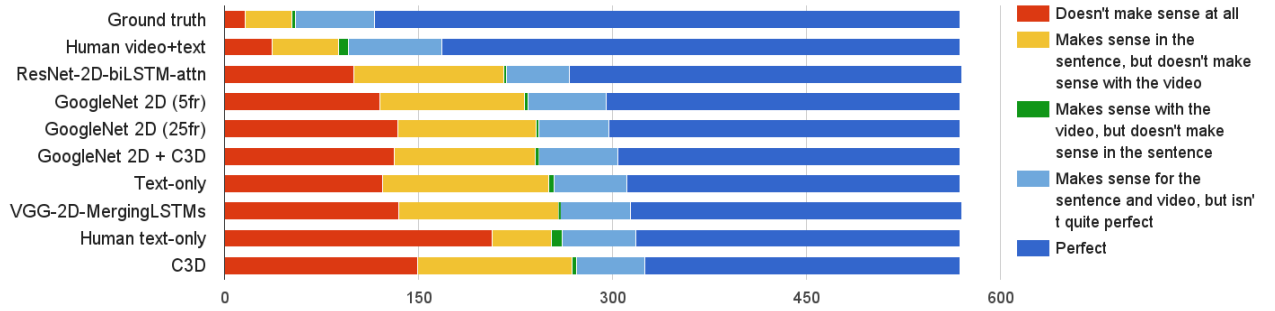

Figure 10. Human evaluation of different models' answers.

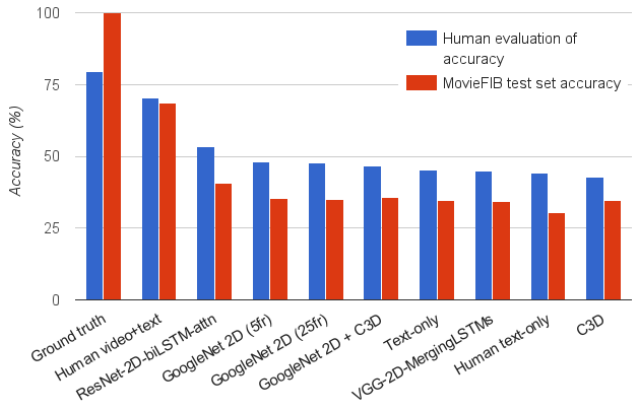

Figure 11. Performance on the test set and performance according to human evaluation, demonstrating that these metrics correspond well.

5.5. Human evaluation of results

We employ Mechanical Turk workers to rank the responses from the models described in Table 2. Workers are given the clip and question, and a list of the different models' responses (including ground truth). Figure 10 shows how humans evaluated different models' responses. Interestingly, humans evaluate that the ground truth is "Perfect" in about 80% of examples, an additional 11% "Make sense for the sentence and video, but isn't quite perfect", and for 3% of ground truth answers (16 examples) workers say the ground truth "Doesn't make sense at all". We observe that for most of these examples, the issue appears to be language style; for example "He _____ her" where the ground truth is "eyes". This may be an unfamiliar use of language for some workers, which is supported by the Human text-only results (see Section 5.3). Figure 11 shows that accuracy tracks the human evaluation well on the test set, in other words, that accuracy on MovieFIB is a representative metric.

6. Conclusion

We have presented MovieFIB, a fill-in-the-blank question-answering dataset, based on descriptive video annotations for the visually impaired, with over 300,000 question-answer and video pairs.

To explore our dataset, and to better understand the capabilities of video models in general, we have evaluated five different models and compared them with human performance, as well as with two independent works using our dataset [22, 44]. We observe that using both visual and temporal information is of prime importance to model performance on this task. However, all models still perform significantly worse than human-level in their use of video.

We have studied the importance of quantity of training data, showing that models leveraging visual input benefit more than text-only models from an increase of the training samples. This suggests that performance could be further improved just by increasing the amount of training data.

Finally, we have performed a human evaluation of all models' responses on the dataset so far. These results show that accuracy on MovieFIB is a robust metric, corresponding well with human assessment.

We hope that the MovieFIB dataset will be useful to develop and evaluate models which better understand the moving-visual content of videos, and that it will encourage further research and progress in this field.

For future work, we suggest: (1) transforming a difficult-to-evaluate task (e.g. 'translation' between modalities, generation, etc.) into a classification task is a broadly applicable idea, useful for benchmarking models; (2) exploring spatio-temporal attention; (3) determining which factors contribute most to improvement of video model performance - increasing data, refinement of existing architectures, development of novel spatio-temporal architectures, etc; (4) further investigation of multimodal fusion in video (e.g. better combining text and visual, leveraging audio).

References

[1] Trecvid med 14. `http://nist.gov/itl/iad/mig/med14.cfm`. Accessed: 2016-11-13.

[2] S. B. aand Edward Loper and E. Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

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

[5] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 1994.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv:1406.1078*, 2014.

[7] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville. Recurrent batch normalization. *arXiv:1603.09025*, 2016.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

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

[10] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv:1411.4389*, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[12] S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Master's thesis*, 1991.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[14] 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.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[16] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 2002.

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

[18] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*. Barcelona, Spain, 2004.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[20] G. Malecha and I. Smith. Maximum entropy part-of-speech tagging in nltk, 2010.

[21] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014.

[22] A. Mazaheri, D. Zhang, and M. Shah. Video fill in the blank with merging lstms. *arXiv:1610.04062*, 2016.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

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

[25] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *EMNLP*. 1996.

[26] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *ACL*, 2013.

[27] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015.

[28] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *CVPR*, 2015.

[29] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *International Journal of Computer Vision*, 2017.

[30] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[31] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[33] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

[34] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[35] 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.

[36] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016.

[37] A. Torabi, C. Pal, H. Larochelle, and A. Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv:1503.01070*, 2015.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[39] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[40] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *NAACL*, 2015.

[41] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, 2015.

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

[43] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank image generation and question answering. *CoRR*, abs/1506.00278, 2015.

[44] Y. Yu, H. Ko, J. Choi, and G. Kim. Video captioning and retrieval models with semantic attention. *arXiv:1610.02947*, 2016.

[45] L. Zhu, Z. Xu, Y. Yang, and A. G. Hauptmann. Uncovering temporal context for video question and answering. *arXiv:1511.04670*, 2015.

[46] C. L. Zitnick, R. Vedantam, and D. Parikh. Adopting abstract images for semantic scene understanding. *PAMI*, 2016.