MovieQA: Understanding Stories in Movies through Question-Answering

Citation: Tapaswi, Makarand, et al. "MovieQA: Understanding Stories in Movies through Question-Answering." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27-30 June, 2016, Las Vegas, Nevada, IEEE, 2016, pp. 4631–40.

As Published: http://dx.doi.org/10.1109/CVPR.2016.501

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: http://hdl.handle.net/1721.1/113894

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

MovieQA: Understanding Stories in Movies through Question-Answering

Makarand Tapaswi[1], Yukun Zhu[3], Rainer Stiefelhagen[1]
Antonio Torralba[2], Raquel Urtasun[3], Sanja Fidler[3]
[1]Karlsruhe Institute of Technology, [2]Massachusetts Institute of Technology, [3]University of Toronto
{tapaswi,rainer.stiefelhagen}@kit.edu, torralba@csail.mit.edu, {yukun,urtasun,fidler}@cs.toronto.edu
http://movieqa.cs.toronto.edu

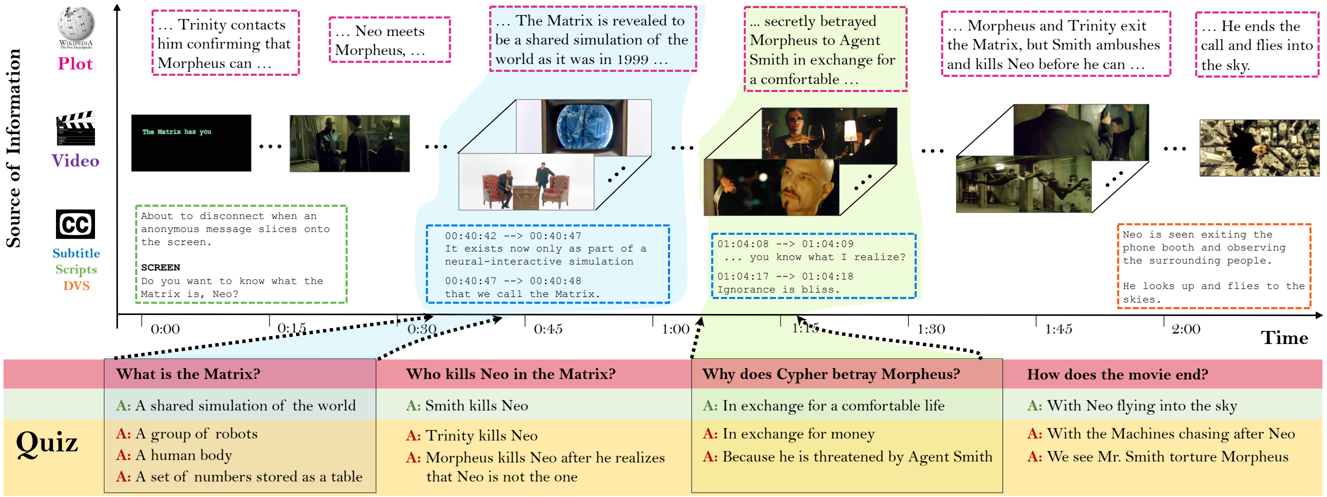
Source of Information
Plot
Video
Subtitle
Scripts
DVS
… Trinity contacts him confirming that Morpheus can …
… Neo meets Morpheus, …
… The Matrix is revealed to be a shared simulation of the world as it was in 1999 …
… secretly betrayed Morpheus to Agent Smith in exchange for a comfortable …
… Morpheus and Trinity exit the Matrix, but Smith ambushes and kills Neo before he can …
… He ends the call and flies into the sky.
The Matrix has you
About to disconnect when an anonymous message slices onto the screen.
SCREEN
Do you want to know what the Matrix is, Neo?
00:40:42 --> 00:40:47
It exists now only as part of a neural-interactive simulation
00:40:47 --> 00:40:48
that we call the Matrix.
01:04:08 --> 01:04:09
... you know what I realize?
01:04:17 --> 01:04:18
Ignorance is bliss.
Neo is seen exiting the phone booth and observing the surrounding people.
He looks up and flies to the skies.
0:00
0:15
0:30
0:45
1:00
1:15
1:30
1:45
2:00
Time
Quiz
What is the Matrix?
A: A shared simulation of the world
A: A group of robots
A: A human body
A: A set of numbers stored as a table
Who kills Neo in the Matrix?
A: Smith kills Neo
A: Trinity kills Neo
A: Morpheus kills Neo after he realizes that Neo is not the one
Why does Cypher betray Morpheus?
A: In exchange for a comfortable life
A: In exchange for money
A: Because he is threatened by Agent Smith
How does the movie end?
A: With Neo flying into the sky
A: With the Machines chasing after Neo
A: We see Mr. Smith torture Morpheus

Figure 1: Our MovieQA dataset contains 14,944 questions about 408 movies. It contains multiple sources of information: plots, subtitles, video clips, scripts, and DVS transcriptions. In this figure we show example QAs from *The Matrix* and localize them in the timeline.

Abstract

We introduce the MovieQA dataset which aims to evaluate automatic story comprehension from both video and text. The dataset consists of 14,944 questions about 408 movies with high semantic diversity. The questions range from simpler "Who" did "What" to "Whom", to "Why" and "How" certain events occurred. Each question comes with a set of five possible answers; a correct one and four deceiving answers provided by human annotators. Our dataset is unique in that it contains multiple sources of information – video clips, plots, subtitles, scripts, and DVS [32]. We analyze our data through various statistics and methods. We further extend existing QA techniques to show that question-answering with such open-ended semantics is hard. We make this data set public along with an evaluation benchmark to encourage inspiring work in this challenging domain.

1. Introduction

Fast progress in Deep Learning as well as a large amount of available labeled data has significantly pushed forward the performance in many visual tasks such as image tagging, object detection and segmentation, action recognition, and image/video captioning. We are steps closer to applications such as assistive solutions for the visually impaired, or cognitive robotics, which require a holistic understanding of the visual world by reasoning about all these tasks in a common framework. However, a truly intelligent machine would ideally also infer high-level semantics underlying human actions such as motivation, intent and emotion, in order to react and, possibly, communicate appropriately. These topics have only begun to be explored in the literature [27, 49].

A great way of showing one's understanding about the scene is to be able to answer any question about it [23]. This idea gave rise to several question-answering datasets which provide a set of questions for each image along with multi-choice answers. These datasets are either based on RGB-D images [23] or a large collection of static photos such as Microsoft COCO [1, 47]. The types of questions typically asked are "What" is there and "Where" is it, what attributes an object has, what is its relation to other objects in the scene, and "How many" objects of certain type are present. While these questions verify the holistic nature of

Q: How does E.T. show his happiness that he is finally returning home?
A: His heart lights up

Q: Why do Joy and Jack get married that first night they meet in Las Vegas?
A: They are both vulnerable and totally drunk

Q: Why does Forrest undertake a three-year marathon?
A: Because he is upset that Jenny left him

Q: How does Patrick start winning Kat over?
A: By getting personal information about her likes and dislikes

Figure 2: Examples from the MovieQA dataset. For illustration we show a single frame, however, all these questions/answers are time-stamped to a much longer clip in the movie. Notice that while some questions can be answered using vision or dialogs alone, most require both. Vision can be used to locate the scene set by the question, and semantics extracted from dialogs can be used to answer it.

our vision algorithms, there is an inherent limitation in what can be asked about a static image. High-level semantics about actions and their intent is mostly lost and can typically only be inferred from temporal, possibly life-long visual observations.

Movies provide us with snapshots from people's lives that link into stories, allowing an experienced human viewer to get a high-level understanding of the characters, their actions, and the motivations behind them. Our goal is to create a question-answering dataset to evaluate machine comprehension of both, complex videos such as movies and their accompanying text. We believe that this data will help push automatic semantic understanding to the next level, required to truly understand stories of such complexity.

This paper introduces MovieQA, a large-scale question-answering dataset about movies. Our dataset consists of 14,944 multiple-choice questions with five deceiving options, of which only one is correct, sourced from 408 movies with high semantic diversity. For 140 of these movies (6,462 QAs), we have timestamp annotations indicating the location of the question and answer in the video. The questions range from simpler "Who" did "What" to "Whom" that can be solved by vision alone, to "Why" and "How" something happened, that can only be solved by exploiting both the visual information and dialogs (see Fig. 2 for a few example "Why" and "How" questions). Our dataset is unique in that it contains multiple sources of information: video clips, subtitles, scripts, plots, and DVS [32] as illustrated in Fig. 1. We analyze the data through various statistics and intelligent baselines that mimic how different "students" would approach the quiz. We further extend existing QA techniques to work with our data and show that question-answering with such open-ended semantics is hard. We have created an online benchmark with a leaderboard (http://movieqa.cs.toronto.edu/leaderboard), encouraging inspiring work in this challenging domain.

2. Related work

Integration of language and vision is a natural step towards improved understanding and is receiving increasing attention from the research community. This is in large part due to efforts in large-scale data collection such as Microsoft's COCO [22], Flickr30K [46] and Abstract Scenes [50] providing tens to hundreds of thousand images with natural language captions. Having access to such data enabled the community to shift from hand-crafted language templates typically used for image description [19] or retrieval-based approaches [11, 26, 45] to deep neural models [6, 13, 15, 42] that achieve impressive captioning results. Another way of conveying semantic understanding of both vision and text is by retrieving semantically meaningful images given a natural language query [13]. An interesting direction, particularly for the goals of our paper, is also the task of learning common sense knowledge from captioned images [40]. This has so far been demonstrated only on synthetic clip-art scenes which enable perfect visual parsing.

Video understanding via language. In the video domain, there are fewer works on integrating vision and language, likely due to less available labeled data. In [10, 41], the authors caption video clips using LSTMs, [33] formulates description as a machine translation model, while older work uses templates [3, 8, 18]. In [21], the authors retrieve relevant video clips for natural language queries, while [29] exploits captioned clips to learn action and role models. For TV series in particular, the majority of work aims at recognizing and tracking characters in the videos [2, 4, 28, 35]. In [7, 34], the authors aligned videos with movie scripts in order to improve scene prediction. [39] aligns movies with their plot synopses with the aim to allow semantic browsing of large video content via textual queries. Just recently, [38, 49] aligned movies to books with the aim to ground temporal visual data with verbose and detailed descriptions available in books.

Question-answering. QA is a popular task in NLP with significant advances made recently with neural models such as memory networks [36], deep LSTMs [12], and structured prediction [43]. In computer vision, [23] proposed a Bayesian approach on top of a logic-based QA system [20], while [24, 30] encoded both an image and the question using an LSTM and decoded an answer. We are not aware of QA methods addressing the temporal domain.

	TRAIN	VAL	TEST	TOTAL
Movies with Plots and Subtitles				
#Movies	269	56	83	408
#QA	9848	1958	3138	14944
Q #words	9.3	9.3	9.5	9.3 ± 3.5
CA. #words	5.7	5.4	5.4	5.6 ± 4.1
WA. #words	5.2	5.0	5.1	5.1 ± 3.9
Movies with Video Clips				
#Movies	93	21	26	140
#QA	4318	886	1258	6462
#Video clips	4385	1098	1288	6771
Mean clip dur. (s)	201.0	198.5	211.4	202.7 ± 216.2
Mean QA #shots	45.6	49.0	46.6	46.3 ± 57.1

Table 1: MovieQA dataset stats. Our dataset supports two modes of answering: text and video. We present the split into train, val, and test splits for the number of movies and questions. We also present mean counts with standard deviations in the total column.

QA Datasets. Most available datasets focus on image [17, 22, 46, 50] or video description [5, 32, 9]. Particularly relevant to our work is the MovieDescription dataset [32] which transcribed text from the Described Video Service (DVS), a narration service for the visually impaired, for a collection of over 100 movies. For QA, [23] provides questions and answers (mainly lists of objects, colors, *etc*.) for the NYUv2 RGB-D dataset, while [1, 47] do so for MS-COCO with a dataset of a million QAs. While these datasets are unique in testing the vision algorithms in performing various tasks such as recognition, attribute induction and counting, they are inherently limited to static images. In our work, we collect a large QA dataset sourced from over 400 movies with challenging questions that require semantic reasoning over a long temporal domain.

Our dataset is also related to purely text QA datasets such as MCTest [31] which contains 660 short stories with 4 multi-choice QAs each, and [12] which converted 300K news summaries into Cloze-style questions. We go beyond these datasets by having significantly longer text, as well as multiple sources of available information (plots, subtitles, scripts and DVS). This makes our data one of a kind.

3. MovieQA dataset

The goal of our paper is to create a challenging benchmark that evaluates semantic understanding over long temporal data. We collect a dataset with very diverse sources of information that can be exploited in this challenging domain. Our data consists of quizzes about movies that the automatic systems will have to answer. For each movie, a quiz comprises of a set of questions, each with 5 multiple-choice answers, only one of which is correct. The system has access to various sources of textual and visual information, which we describe in detail below.

We collected 408 subtitled movies, and obtained their extended summaries in the form of plot synopses from *Wikipedia*. We crawled *imsdb* for scripts, which were available for 49% (199) of our movies. A fraction of our movies (60) come with DVS transcriptions provided by [32].

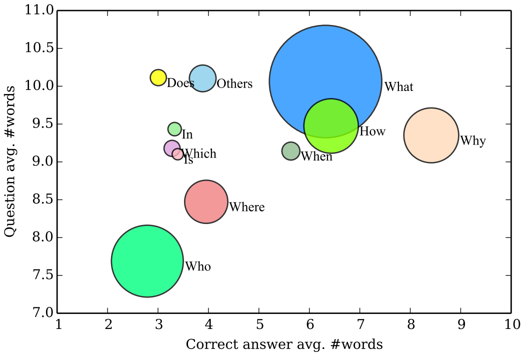

Figure 3: Average number of words in MovieQA dataset based on the first word in the question. Area of a bubble indicates #QA.

Plot synopses are movie summaries that fans write after watching the movie. Synopses widely vary in detail and range from one to 20 paragraphs, but focus on describing content that is directly relevant to the story. They rarely contain detailed visual information (*e.g*. character appearance), and focus more on describing the movie events and character interactions. We exploit plots to gather our quizzes.

Videos and subtitles. An average movie is about 2 hours in length and has over 198K frames and almost 2000 shots. Note that video alone contains information about e.g., “Who” did “What” to “Whom”, but may be lacking in information to explain why something happened. Dialogs play an important role, and only both modalities together allow us to fully understand the story. Note that subtitles do not contain speaker information. In our dataset, we provide video clips rather than full movies.

DVS is a service that narrates movie scenes to the visually impaired by inserting relevant descriptions in between dialogs. These descriptions contain sufficient “visual” information about the scene that they allow visually impaired audience to follow the movie. DVS thus acts as a proxy for a perfect vision system, and is another source for answering.

Scripts. The scripts that we collected are written by screenwriters and serve as a guideline for movie making. They typically contain detailed descriptions of scenes, and, unlike subtitles, contain both dialogs and speaker information. Scripts are thus similar, if not richer in content to DVS+Subtitles, however are not always entirely faithful to the movie as the director may aspire to artistic freedom.

3.1. QA Collection method

Since videos are difficult and expensive to provide to annotators, we used plot synopses as a proxy for the movie. While creating quizzes, our annotators only referred to the story plot and were thus automatically coerced into asking story-like questions. We split our annotation efforts into two primary parts to ensure high quality of the collected data.

	Txt	Img	Vid	Goal	Data source	AType	#Q	AW
MCTest [31]	✓	-	-	reading comprehension	Children stories	MC (4)	2,640	3.40
bAbI [44]	✓	-	-	reasoning for toy tasks	Synthetic	Word	20×2,000	1.0
CNN+DailyMail [12]	✓	-	-	information abstraction	News articles	Word	1,000,000*	1*
DAQUAR [23]	-	✓	-	visual: counts, colors, objects	NYU-RGBD	Word/List	12,468	1.15
Visual Madlibs [47]	-	✓	-	visual: scene, objects, person, ...	COCO+Prompts	FITB/MC (4)	2×75,208*	2.59
VQA (v1) [1]	-	✓	-	visual understanding	COCO+Abstract	Open/MC (18)	764,163	1.24
MovieQA	✓	✓	✓	text+visual story comprehension	Movie stories	MC (5)	14,944	5.29

Table 2: A comparison of various QA datasets. First three columns depict the modality in which the story is presented. AType: answer type; AW: average # of words in answer(s); MC (N): multiple choice with N answers; FITB: fill in the blanks; *estimated information.

Q and correct A. Our annotators were first asked to select a movie from a large list, and were shown its plot synopsis one paragraph at a time. For each paragraph, the annotator had the freedom of forming any number and type of questions. Each annotator was asked to provide the correct answer, and was additionally required to mark a minimal set of sentences within the plot synopsis paragraph that can be used to both frame the question and answer it. This was treated as ground-truth for localizing the QA in the plot.

In our instructions, we asked the annotators to provide context to each question, such that a human taking the quiz should be able to answer it by watching the movie alone (without having access to the synopsis). The purpose of this was to ensure questions that are localizable in the video and story as opposed to generic questions such as “What are they talking?”. We trained our annotators for about one to two hours and gave them the option to re-visit and correct their data. The annotators were paid by the hour, a strategy that allowed us to collect more thoughtful and complex QAs, rather than short questions and single-word answers.

Multiple answer choices. In the second step of data collection, we collected multiple-choice answers for each question. Our annotators were shown a paragraph and a question at a time, but not the correct answer. They were then asked to answer the question correctly as well as provide 4 wrong answers. These answers were either deceiving facts from the same paragraph or common-sense answers. The annotator was also allowed to re-formulate or correct the question. We used this to sanity check all the questions received in the first step. All QAs from the “val” and “test” set underwent another round of clean up.

Time-stamp to video. We further asked in-house annotators to align each sentence in the plot synopsis to the video by marking the beginning and end (in seconds) of the video that the sentence describes. Long and complicated plot sentences were often aligned to multiple, non-consecutive video clips. Annotation took roughly 2 hours per movie. Since we have each QA aligned to a sentence(s) in the plot synopsis, the video to plot alignment links QAs with video clips. We provide these clips as part of our benchmark.

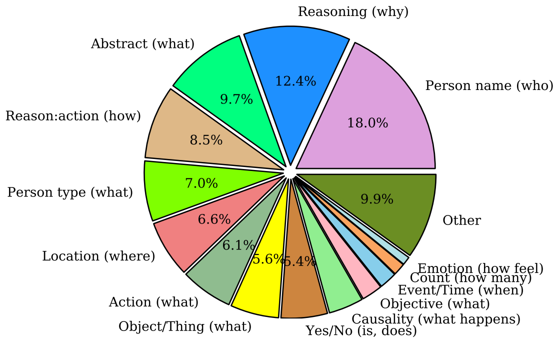

Figure 4: Stats about MovieQA questions based on answer types. Note how questions beginning with the same word may cover a variety of answer types: *Causality*: What happens ... ?; *Action*: What did X do? *Person name*: What is the killer’s name?; *etc*.

3.2. Dataset Statistics

In the following, we present some statistics of our MovieQA dataset. Table 2 presents an overview of popular and recent Question-Answering datasets in the field. Most datasets (except MCTest) use very short answers and are thus limited to covering simpler visual/textual forms of understanding. To the best of our knowledge, our dataset not only has long sentence-like answers, but is also the first to use videos in the form of movies.

Multi-choice QA. We collected a total of 14,944 QAs from 408 movies. Each question comes with one correct and four deceiving answers. Table 1 presents an overview of the dataset along with information about the train/val/test splits, which will be used to evaluate automatically trained QA models. On average, our questions and answers are fairly long with about 9 and 5 words respectively unlike most other QA datasets. The video-based answering split for our dataset, supports 140 movies for which we aligned plot synopses with videos. Note that the QA methods needs to look at a long video clip (∼200s) to answer the question.

Fig. 3 presents the number of questions (bubble area) split based on the first word of the question along with information about number of words in the question and answer. Of particular interest are “Why” questions that require verbose answers, justified by having the largest average number of words in the correct answer, and in contrast, “Who” questions with answers being short people names.

Text type	# Movies	# Sent. / Mov.	# Words in Sent.
Plot	408	35.2	20.3
Subtitle	408	1558.3	6.2
Script	199	2876.8	8.3
DVS	60	636.3	9.3

Table 3: Statistics for the various text sources used for answering.

Instead of the first word in the question, a peculiar way to categorize QAs is based on the answer type. We present such an analysis in Fig. 4. Note how reasoning based questions (Why, How, Abstract) are a large part of our data. In the bottom left quadrant we see typical question types that can likely be answered using vision alone. Note however, that even the reasoning questions typically require vision, as the question context provides a visual description of a scene (*e.g*., "Why does John run after Mary?").

Text sources for answering. In Table 3, we summarize and present some statistics about different text sources used for answering. Note how plot synopses have a large number of words per sentence, hinting towards the richness and complexity of the source.

4. Multi-choice Question-Answering

We now investigate a number of intelligent baselines for QA. We also study inherent biases in the data and try to answer the quizzes based simply on answer characteristics such as word length or within answer diversity.

Formally, let S denote the story, which can take the form of any of the available sources of information – *e.g.* plots, subtitles, or video shots. Each story S has a set of questions, and we assume that the (automatic) student reads one question q^S at a time. Let $\{a_j^S\}_{j=1}^M$ be the set of multiple choice answers (only one of which is correct) corresponding to q^S, with $M = 5$ in our dataset.

The general problem of multi-choice question answering can be formulated by a three-way scoring function $f(S, q^S, a^S)$. This function evaluates the "quality" of the answer given the story and the question. Our goal is thus to pick the best answer a^S for question q^S that maximizes f:

$$j^* = \arg \max_{j=1 \ldots M} f(S, q^S, a_j^S) \tag{1}$$

Answering schemes are thus different functions f. We drop the superscript $(\cdot)^S$ for simplicity of notation.

4.1. The Hasty Student

We first consider f which ignores the story and attempts to answer the question directly based on latent biases and similarities. We call such a baseline as the "Hasty Student" since he/she is not concerned to read/watch the actual story.

The extreme case of a hasty student is to try and answer the question by only looking at the answers. Here, $f(S, q, a_j) = g_{H1}(a_j|\mathbf{a})$, where $g_{H1}(\cdot)$ captures some properties of the answers.

Answer length. We explore using the number of words in the multiple choices to find the correct answer and explore biases in the dataset. As shown in Table 1, correct answers are slightly longer as it is often difficult to frame long deceiving answers. We choose an answer by: (i) selecting the longest answer; (ii) selecting the shortest answer; or (iii) selecting the answer with the most different length.

Within answer similarity/difference. While still looking only at the answers, we compute a distance between all answers based on their representations (discussed in Sec. 4.4). We then select our answer as either the most similar or most distinct among all answers.

Q and A similarity. We now consider a hasty student that looks at both the question and answer, $f(S, q, a_j) = g_{H2}(q, a_j)$. We compute similarity between the question and each answer and pick the highest scoring answer.

4.2. The Searching Student

While the hasty student ignores the story, we consider a student that tries to answer the question by trying to locate a subset of the story S which is most similar to both the question and the answer. The scoring function f is

$$f(S, q, a_j) = g_I(S, q) + g_I(S, a_j)\,. \tag{2}$$

a factorization of the question and answer similarity. We propose two similarity functions: a simple windowed cosine similarity, and another using a neural architecture.

Cosine similarity with a sliding window. We aim to find the best window of H sentences (or shots) in the story S that maximize similarity between the story and question, and story and answer. We define our similarity function:

$$f(S, q, a_j) = \max_l \sum_{k=l}^{l+H} g_{ss}(s_k, q) + g_{ss}(s_k, a_j)\,, \tag{3}$$

where s_k denotes a sentence (or shot) from the story S. We use $g_{ss}(s, q) = x(s)^T x(q)$ as a dot product between the (normalized) representations of the two sentences (shots). We discuss these representations in detail in Sec. 4.4.

Searching student with a convolutional brain (SSCB). Instead of factoring $f(S, q, a_j)$ as a fixed (unweighted) sum of two similarity functions $g_I(S, q)$ and $g_I(S, a_j)$, we build a neural network that learns such a function. Assuming the story S is of length n, *e.g.* n plot sentences or n video shots, $g_I(S, q)$ and $g_I(S, a_j)$ can be seen as two vectors of length n whose k-th entry is $g_{ss}(s_k, q)$. We further combine all $[g_I(S, a_j)]_j$ for the 5 answers into a $n \times 5$ matrix. The vector $g_I(S, q)$ is replicated 5-times, and we stack the question and answer matrix together to obtain a tensor of size $n \times 5 \times 2$.

Our neural similarity model is a convnet (CNN), shown in Fig. 5, that takes the above tensor, and applies couple layers of $h = 10$, 1×1 convolutions to approximate a family

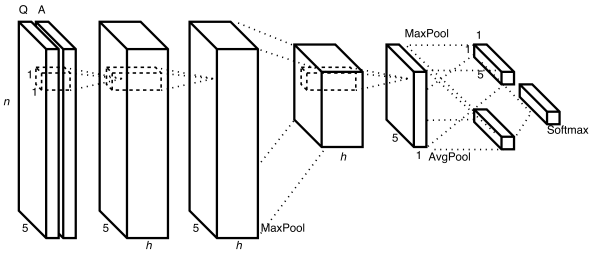

Figure 5: Our neural similarity architecture (see text for details).

of functions $\phi(g_I(S, q), g_I(S, a_j))$. Additionally, we incorporate a max pooling layer with kernel size 3 to allow for scoring the similarity within a window in the story. The last convolutional output is a tensor with shape $(\frac{n}{3}, 5)$, and we apply both mean and max pooling across the storyline, add them, and make predictions using softmax. We train our network using cross-entropy loss and the Adam optimizer [14].

4.3. Memory Network for Complex QA

Memory Networks were originally proposed for text QA and model complex three-way relationships between the story, question and answer. We briefly describe MemN2N proposed by [36] and suggest simple extensions to make it suitable for our data and task.

The input of the original MemN2N is a story and question. The answering is restricted to single words and is done by picking the most likely word from the vocabulary $\mathcal{V}$ of 20-40 words. Note that this is not directly applicable to MovieQA, as our data set does not have perform vocabulary-based answering.

A question q is encoded as a vector $u \in \mathbb{R}^d$ using a word embedding $B \in \mathbb{R}^{d \times |\mathcal{V}|}$. Here, d is the embedding dimension, and u is obtained by mean-pooling the representations of words in the question. Simultaneously, the sentences of the story s_l are encoded using word embeddings A and C to provide two different sentence representations m_l and c_l, respectively. m_l, the representation of sentence l in the story, is used in conjunction with u to produce an attention-like mechanism which selects sentences in the story most similar to the question via a softmax function:

$$p_l = \text{softmax}(u^T m_l)\,. \tag{4}$$

The probability p_l is used to weight the second sentence embedding c_l, and the output $o = \sum_l p_l c_l$ is obtained by pooling the weighted sentence representations across the story. Finally, a linear projection $W \in \mathbb{R}^{|\mathcal{V}| \times d}$ decodes the question u and the story representation o to provide a soft score for each vocabulary word

$$a = \text{softmax}(W(o + u))\,. \tag{5}$$

The top scoring word $\hat{a}$ is picked from a as the answer. The free parameters to train are the embeddings B, A, C, W for different words which can be shared across different layers.

Due to its fixed set of output answers, the MemN2N in the current form is not designed for multi-choice answering with open, natural language answers. We propose two key modifications to make the network suitable for our task.

MemN2N for natural language answers. To allow the MemN2N to rank multiple answers written in natural language, we add an additional embedding layer F which maps each multi-choice answer a_j to a vector g_j. Note that F is similar to embeddings B, A and C, but operates on answers instead of the question or story. To predict the correct answer, we compute the similarity between the answers g, the question embedding u and the story representation o:

$$a = \text{softmax}((o + u)^T g) \tag{6}$$

and pick the most probable answer as correct. In our general QA formulation, this is equivalent to

$$f(S, q, a_j) = g_{M1}(S, q, a_j) + g_{M2}(q, a_j), \tag{7}$$

where g_{M1} attends to parts of the story using the question, and a second function g_{M2} directly considers similarities between the question and the answer.

Weight sharing and fixed word embeddings. The original MemN2N learns embeddings for each word based directly on the task of question-answering. However, to scale this to large vocabulary data sets like ours, this requires unreasonable amounts of training data. For example, training a model with a vocabulary size 14,000 (obtained just from plot synopses) and $d = 100$ would entail learning 1.4M parameters for each embedding. To prevent overfitting, we first share all word embeddings B, A, C, F of the memory network. Nevertheless, even one embedding is still a large number of parameters.

We make the following crucial modification that allows us to use the Memory Network for our dataset. We drop B, A, C, F and replace them by a fixed (pre-trained) word embedding $Z \in \mathbb{R}^{d_1 \times |\mathcal{V}|}$ obtained from the Word2Vec model and learn a shared linear projection layer $T \in \mathbb{R}^{d_2 \times d_1}$ to map all sentences (stories, questions and answers) into a common space. Here, d_1 is the dimension of the Word2Vec embedding, and d_2 is the projection dimension. Thus, the new encodings are

$$u = T \cdot Zq;\; m_l, c_l = T \cdot Zs_l;\; \text{and}\; g_j = T \cdot Za_j. \tag{8}$$

Answer prediction is performed as before in Eq. 6.

We initialize T either using an identity matrix $d_1 \times d_1$ or using PCA to lower the dimension from $d_1 = 300$ to $d_2 = 100$. Training is performed using stochastic gradient descent with a batch size of 32.

4.4. Representations for Text and Video

TF-IDF is a popular and successful feature in information retrieval. In our case, we treat plots (or other forms of text) from different movies as documents and compute a weight for each word. We set all words to lower case, use stemming, and compute the vocabulary $\mathcal{V}$ which consists of words w that appear more than θ times in the documents. We represent each sentence (or question or answer) in a bag-of-words style with an TF-IDF score for each word.

Word2Vec. A disadvantage of TF-IDF is that it is unable to capture the similarities between words. We use the skip-gram model proposed by [25] and train it on roughly 1200 movie plots to obtain domain-specific, 300 dimensional word embeddings. A sentence is then represented by mean-pooling its word embeddings. We normalize the resulting vector to have unit norm.

SkipThoughts. While the sentence representation using mean pooled Word2Vec discards word order, SkipThoughts [16] use a Recurrent Neural Network to capture the underlying sentence semantics. We use the pre-trained model by [16] to compute a 4800 dimensional sentence representation.

Video. To answer questions from the video, we learn an embedding between a shot and a sentence, which maps the two modalities in a common space. In this joint space, one can score the similarity between the two modalities via a simple dot product. This allows us to apply all of our proposed question-answering techniques in their original form.

To learn the joint embedding we follow [49] which extends [15] to video. Specifically, we use the GoogLeNet architecture [37] as well as hybrid-CNN [48] to extract frame-wise features, and mean-pool the representations over all frames in a shot. The embedding is a linear mapping of the shot representation and an LSTM on word embeddings on the sentence side, trained using the ranking loss on the MovieDescription Dataset [32] as in [49].

5. QA Evaluation

We present results for question-answering with the proposed methods on our MovieQA dataset. We study how various sources of information influence the performance, and how different levels of complexity encoded in f affects the quality of automatic QA.

Protocol. Note that we have two primary tasks for evaluation. (i) **Text-based**: the story takes the form of various texts – plots, subtitles, scripts, DVS; and (ii) **Video-based**: story is the video, and with/without subtitles.

Dataset structure. The dataset is divided into three disjoint splits: *train*, *val*, and *test*, based on unique movie titles in each split. The splits are optimized to preserve the ratios between #movies, #QAs, and all the story sources at 10:2:3 (*e.g.* about 10k, 2k, and 3k QAs). Stats for each split are presented in Table 1. The *train* set is to be used for training automatic models and tuning any hyperparameters. The *val* set should not be touched during training, and may be used to report results for several models. The *test* set is a held-out set, and is evaluated on our MovieQA server. For this paper, all results are presented on the *val* set.

Answer length		longest	shortest	different
		25.33	14.56	20.38
Within answers		TF-IDF	SkipT	w2v
	similar	21.71	28.14	25.43
	distinct	19.92	14.91	15.12
Question-answer		TF-IDF	SkipT	w2v
	similar	12.97	19.25	24.97

Table 4: The question-answering accuracy for the "Hasty Student" who tries to answer questions without looking at the story.

Metrics. Multiple choice QA leads to a simple and objective evaluation. We measure *accuracy*, the number of correctly answered QAs over the total count.

5.1. The Hasty Student

The first part of Table 4 shows the performance of three models when trying to answer questions based on the answer length. Notably, always choosing the longest answer performs better (25.3%) than random (20%). The second part of Table 4 presents results when using within-answer feature-based similarity. We see that the answer most similar to others is likely to be correct when the representations are generic and try to capture the semantics of the sentence (Word2Vec, SkipThoughts). The most distinct answers performs worse than random on all features. In the last section of Table 4 we see that computing feature-based similarity between questions and answers is insufficient for answering. Especially, TF-IDF performs worse than random since words in the question rarely appear in the answer.

Hasty Turker. To analyze the deceiving nature of our multi-choice QAs, we tested humans (via AMT) on a subset of 200 QAs. The turkers were not shown the story in any form and were asked to pick the best possible answer given the question and a set of options. We asked each question to 10 turkers, and rewarded each with a bonus if their answer agreed with the majority. We observe that without access to the story, humans obtain an accuracy of 27.6%. We suspect that the bias is due to the fact that some of the QAs reveal the movie (e.g., "Darth Vader") and the turker may have seen this movie. Removing such questions, and re-evaluating on a subset of 135 QAs, lowers the performance to 24.7%. This shows the genuine difficulty of our QAs.

5.2. Searching Student

Cosine similarity with window. The first section of Table 5 presents results for the proposed cosine similarity using different representations and text stories. Using the plots to answer questions outperforms other sources (subtitles, scripts, and DVS) as the QAs were collected using plots and annotators often reproduce words from the plot.

We show the results of using Word2Vec or SkipThought representations in the following rows of Table 5.

Method	Plot	DVS	Subtitle	Script
Cosine TFIDF	47.6	24.5	24.5	24.6
Cosine SkipThought	31.0	19.9	21.3	21.2
Cosine Word2Vec	46.4	26.6	24.5	23.4
SSCB TFIDF	48.5	24.5	27.6	26.1
SSCB SkipThought	28.3	24.5	20.8	21.0
SSCB Word2Vec	45.1	24.8	24.8	25.0
SSCB Fusion	**56.7**	24.8	27.7	28.7
MemN2N (w2v, linproj)	40.6	**33.0**	**38.0**	**42.3**

Table 5: Accuracy for Text-based QA. **Top**: results for the Searching student with cosine similarity; **Middle**: Convnet SSCB; and **Bottom**: the modified Memory Network.

SkipThoughts perform much worse than both TF-IDF and Word2Vec which are closer. We suspect that while SkipThoughts are good at capturing the overall semantics of a sentence, proper nouns – names, places – are often hard to distinguish. Fig. 6 presents a accuracy breakup based on the first word of the questions. TF-IDF and Word2Vec perform considerably well, however, we see a larger difference between the two for "Who" and "Why" questions. "Who" questions require distinguishing between names, and "Why" answers are typically long, and mean pooling destroys semantics. In fact Word2Vec performs best on "Where" questions that may use synonyms to indicate places. SkipThoughts perform best on "Why" questions where sentence semantics help improve answering.

SSCB. The middle rows of Table 5 show the result of our neural similarity model. Here, we present additional results combining all text representations (*SSCB fusion*) via our CNN. We split the *train* set into 90% train / 10% dev, such that all questions and answers of the same movie are in the same split, train our model on train and monitor performance on dev. Both *val* and *test* sets are held out. During training, we also create several model replicas and pick the ones with the best validation performance.

Table 5 shows that the neural model outperforms the simple cosine similarity on most tasks, while the fusion method achieves the highest performance when using plot synopses as the story. Ignoring the case of plots, the accuracy is capped at about 30% for most modalities showing the difficulty of our dataset.

5.3. Memory Network

The original MemN2N which trains the word embeddings along with the answering modules overfits heavily on our dataset leading to near random performance on *val* (∼20%). However, our modifications help in restraining the learning process. Table 5 (bottom) presents results for MemN2N with Word2Vec initialization and a linear projection layer. Using plot synopses, we see a performance closer to SSCB with Word2Vec features. However, in the case of longer stories, the attention mechanism in the network is able to sift through thousands of story sentences and performs well on DVS, subtitles and scripts. This shows that complex three-way scoring functions are needed to tackle such QA sources. In terms of story sources, the MemN2N performs best with scripts which contain the most information (descriptions, dialogs and speaker information).

Method	Video	Subtitle	Video+Subtitle
SSCB all clips	21.6	22.3	21.9
MemN2N all clips	**23.1**	**38.0**	**34.2**

Table 6: Accuracy for Video-based QA and late fusion of Subtitle and Video scores.

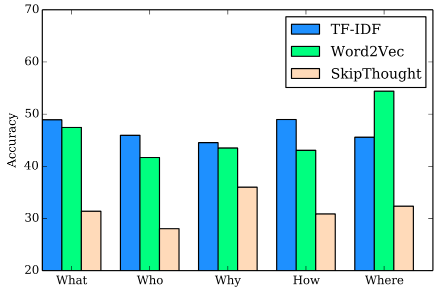

Figure 6: Accuracy for different feature representations of plot sentences with respect to the first word of the question.

5.4. Video baselines

We evaluate SSCB and MemN2N in a setting where the automatic models answer questions by "watching" all the video clips that are provided for that movie. Here, the story descriptors are shot embeddings.

The results are presented in Table 6. We see that learning to answer questions using video is still a hard problem with performance close to random. As visual information alone is insufficient, we also perform and experiment combining video and dialog (subtitles) through late fusion. We train the SSCB model with the visual-text embedding for subtitles and see that it yields poor performance (22.3%) compared to the fusion of all text features (27.7%). For the memory network, we answer subtitles as before using Word2Vec.

6. Conclusion

We introduced the MovieQA data set which aims to evaluate automatic story comprehension from both video and text. Our dataset is unique in that it contains several sources of information – video clips, subtitles, scripts, plots and DVS. We provided several intelligent baselines and extended existing QA techniques to analyze the difficulty of our task. Our benchmark with an evaluation server is online at http://movieqa.cs.toronto.edu.

Acknowledgment. We thank the Upwork annotators, Lea Jensterle, Marko Boben, and Soča Fidler for data collection, and Relu Patrascu for infrastructure support. MT and RS are supported by DFG contract no. STI-598/2-1, and the work was carried out during MT's visit to U. of T. on a KHYS Research Travel Grant.

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. 1, 3, 4

[2] M. Baeuml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised Learning with Constraints for Person Identification in Multimedia Data. In *CVPR*, 2013. 2

[3] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J. Siskind, J. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang. Video-In-sentences Out. In *UAI*, 2012. 2

[4] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding Actors and Actions in Movies. *ICCV*, pages 2280–2287, 2013. 2

[5] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 3

[6] X. Chen and C. L. Zitnick. Learning a Recurrent Visual Representation for Image Caption Generation. In *arXiv:1411.5654*, 2014. 2

[7] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/Script: Alignment and Parsing of Video and Text Transcription. In *ECCV*, 2008. 2

[8] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A Thousand Frames in Just a Few Words: Lingual Description of Videos through Latent Topics and Sparse Object Stitching. *CVPR*, 2013. 2

[9] P. Das, C. Xu, R. F. Doell, and J. J. Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, 2013. 3

[10] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In *arXiv:1411.4389*, 2014. 2

[11] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every Picture Tells a Story: Generating Sentences for Images. In *ECCV*, 2010. 2

[12] K. M. Hermann, T. Kočisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching Machines to Read and Comprehend. In *arXiv:1506.03340*, 2015. 2, 3, 4

[13] A. Karpathy and L. Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *CVPR*, 2015. 2

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 6

[15] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. *TACL*, 2015. 2, 7

[16] R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-Thought Vectors. *NIPS*, 2015. 7

[17] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What are you talking about? Text-to-Image Coreference. In *CVPR*, 2014. 3

[18] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama. Generating Natural-Language Video Descriptions Using Text-Mined Knowledge. In *AAAI*, July 2013. 2

[19] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. Berg, and T. Berg. Baby Talk: Understanding and Generating Simple Image Descriptions. In *CVPR*, 2011. 2

[20] P. Liang, M. Jordan, and D. Klein. Learning dependency-based compositional semantics. In *Computational Linguistics*, 2013. 2

[21] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual Semantic Search: Retrieving Videos via Complex Textual Queries. *CVPR*, 2014. 2

[22] 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. 2, 3

[23] M. Malinowski and M. Fritz. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In *NIPS*, 2014. 1, 2, 3, 4

[24] M. Malinowski, M. Rohrbach, and M. Fritz. Ask Your Neurons: A Neural-based Approach to Answering Questions about Images. In *ICCV*, 2015. 2

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 7

[26] V. Ordonez, G. Kulkarni, and T. Berg. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *NIPS*, 2011. 2

[27] H. Pirsiavash, C. Vondrick, and A. Torralba. Inferring the Why in Images. *arXiv.org*, jun 2014. 1

[28] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking People in Videos with "Their" Names Using Coreference Resolution. In *ECCV*. 2014. 2

[29] V. Ramanathan, P. Liang, and L. Fei-Fei. Video Event Understanding using Natural Language Descriptions. In *ICCV*, 2013. 2

[30] M. Ren, R. Kiros, and R. Zemel. Exploring Models and Data for Image Question Answering. *arXiv:1505.02074*, 2015. 2

[31] M. Richardson, C. J. Burges, and E. Renshaw. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *EMNLP*, 2013. 3, 4

[32] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A Dataset for Movie Description. In *CVPR*, 2015. 1, 2, 3, 7

[33] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating Video Content to Natural Language Descriptions. In *ICCV*, 2013. 2

[34] P. Sankar, C. V. Jawahar, and A. Zisserman. Subtitle-free Movie to Script Alignment. In *BMVC*, 2009. 2

[35] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?" - Learning person specific classifiers from video. *CVPR*, pages 1145–1152, 2009. 2

[36] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-To-End Memory Networks. In *arXiv:1503.08895*, 2015. 2, 6

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv:1409.4842*, 2014. 7

[38] M. Tapaswi, M. Bauml, and R. Stiefelhagen. Book2Movie: Aligning Video scenes with Book chapters. In *CVPR*, 2015. 2

[39] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Aligning Plot Synopses to Videos for Story-based Retrieval. *IJMIR*, 4:3–16, 2015. 2

[40] R. Vedantam, X. Lin, T. Batra, C. L. Zitnick, and D. Parikh. Learning Common Sense Through Visual Abstraction. In *ICCV*, 2015. 2

[41] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. J. Mooney, and K. Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. *CoRR abs/1312.6229*, cs.CV, 2014. 2

[42] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and Tell: A Neural Image Caption Generator. In *arXiv:1411.4555*, 2014. 2

[43] H. Wang, M. Bansal, K. Gimpel, and D. McAllester. Machine Comprehension with Syntax, Frames, and Semantics. In *ACL*, 2015. 2

[44] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *arXiv:1502.05698*, 2014. 4

[45] Y. Yang, C. L. Teo, H. Daumé, III, and Y. Aloimonos. Corpus-guided Sentence Generation of Natural Images. In *EMNLP*, pages 444–454, 2011. 2

[46] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *TACL*, 2014. 2, 3

[47] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual Madlibs: Fill in the blank Image Generation and Question Answering. In *ICCV*, 2015. 1, 3, 4

[48] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning Deep Features for Scene Recognition using Places Database. In *NIPS*, 2014. 7

[49] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In *ICCV*, 2015. 1, 2, 7

[50] C. Zitnick, R. Vedantam, and D. Parikh. Adopting abstract images for semantic scene understanding. *PAMI*, PP, 2014. 2, 3