A Read-Write Memory Network for Movie Story Understanding

Seil Na[1], Sangho Lee[1], Jisung Kim[2], Gunhee Kim[1]
[1]Seoul National University, [2]SK Telecom
{seil.na, sangho.lee}@vision.snu.ac.kr, joyful.kim@sk.com, gunhee@snu.ac.kr
https://github.com/seilna/RWMN

Abstract

We propose a novel memory network model named Read-Write Memory Network (RWMN) to perform question and answering tasks for large-scale, multimodal movie story understanding. The key focus of our RWMN model is to design the read network and the write network that consist of multiple convolutional layers, which enable memory read and write operations to have high capacity and flexibility. While existing memory-augmented network models treat each memory slot as an independent block, our use of multi-layered CNNs allows the model to read and write sequential memory cells as chunks, which is more reasonable to represent a sequential story because adjacent memory blocks often have strong correlations. For evaluation, we apply our model to all the six tasks of the MovieQA benchmark [24], and achieve the best accuracies on several tasks, especially on the visual QA task. Our model shows a potential to better understand not only the content in the story, but also more abstract information, such as relationships between characters and the reasons for their actions.

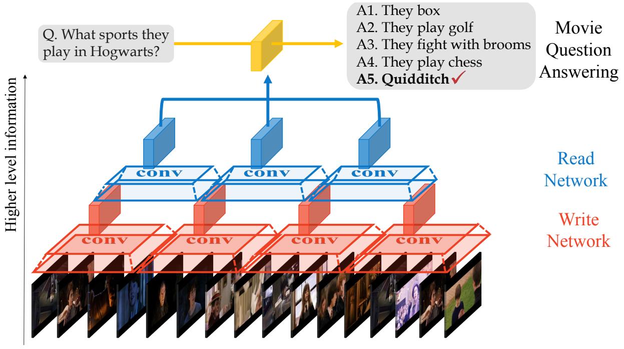


Figure 1. The intuition of the RWMN (*Read-Write Memory Network*) model for movie question and answering tasks. Using read/write networks of multi-layered CNNs, it abstracts a given series of frames stepwise to capture higher-level sequential information and stores it into memory slots. It eventually helps answer complex questions of movie QAs.

1. Introduction

For many problems of video understanding, including video classification [1, 14], video captioning [28, 29] and MovieQA [24], it is key to success for models to correctly process, represent, and store long sequential information. In the era of deep learning, one prevailing approach to model sequential input is to use recurrent neural networks (RNNs) [16] which store the given information into a hidden memory and update it over time. However, RNNs accumulate information in a single fixed-length memory regardless of the length of an input sequence, thus tend to fail to utilize far-distant information due to a vanishing gradient problem, which is still not fully solved even with advanced models such as LSTM [12] and GRU [3].

As another recent alternative to resolve this issue, many studies attempt to leverage an external memory structure for neural networks, often referred to as *neural memory networks* [8, 9, 10, 15, 23, 27]. One key benefit of external memory is to enable a neural model to cache sequential inputs in memory slots, and explicitly utilize even far early information. Such ability is particularly powerful to solve *question and answering* (QA) problems, which often require models to memorize a large amount of information, and correctly access the most relevant information to a given question. For this reason, memory networks have been popularly applied as state-of-the-art approaches to many QA tasks, such as bAbI task [26], SQuAD [21], and LSMDC [22].

MovieQA [24] is another challenging visual QA dataset, in which models need to understand movies over two hours long, and solve QA problems related to movie content and plots. The MovieQA benchmark consists of six tasks according to which sources of information is usable to solve the QA problems, including videos, subtitles, DVS, scripts, plot synopses, and open-end information. Understanding a movie is a highly challenging task; it is necessary not only to understand the content of individual video frames such as a characters' actions, places of events, but also to infer more abstract and high-level knowledge such as reasons of a characters' behaviors, and relationships between them. For instance, in the *Harry Potter* movie, to answer a question (*Q. What does Harry trick Lucius into doing? A. Freeing Dobby*), models need to realize that *Dobby was a Lucius's house elf, wanted to escape from him, had a positive relationship with Harry, and Harry helped him*. Some of such information is visually or textually observable in the movie, but much information like relationships between characters and correlations between events should be deduced.

Our objective is to propose a novel memory network model to perform QA tasks for large-scale, multimodal movie story understanding. That is, the input to the model can be very long (*e.g*. videos more than two hours long), or be multimodal (*e.g*. text-only or video-text pairs). The key focus of our novel memory network named *Read-Write Memory Networks* (RWMN) is on defining the memory read/write operations to have high capacity and flexibility, for which we propose the *read* and *write networks* that consist of multiple convolutional layers. Existing neural memory network models treat each memory slot as an independent block. However, adjacent memory blocks often have strong correlations, which are the case to represent a sequential story. That is, when human understands a story, the entire story is often recognized as a sequence of closely-interconnected abstract events. Hence, preferably memory networks need to read and write sequential memory cells as chunks, which are implemented by multiple convolutional layers of the read and write network.

To conclude introduction, we summarize the contributions of this work as follows.

1. We propose a novel memory network named RWMN that enables the model to flexibly read and write more complex and abstract information into memory slots through read/write networks. To the best of our knowledge, it is the first attempt to leverage multi-layer CNNs for read/write operations of a memory network.

2. The RWMN shows the best accuracies on several tasks of MovieQA benchmark [24]; as of the ICCV2017 submission deadline (March 27, 2017 23:59 GMT), our RWMN achieves the best performance for *four* out of five tasks in the validation set, and *four* out of six tasks in the test set. Our quantitative and qualitative evaluation also assures that the read/write networks effectively utilize higher-level information in the external memory, especially on the visual QA task.

2. Related Work

Neural Memory Networks. Recently, much research has been done to model sequential data using explicit memory architecture. The memory access of existing memory network models can be classified into *content-based* addressing and *location-based* addressing [8]. The content-based addressing (*e.g*. [9, 27, 18]) lets the controller to generate a key vector and measure its similarity with each memory cell to find out which cells are to be *attended* as the relevant cells to the key vector. Location-based addressing (*e.g*. [8]), on the other hand, enables simple arithmetic operations that find out the addresses to store or retrieve information, regardless of the content of the key vector.

Neural Turing Machine (NTM) [8] and its extensions of DNC [9], D-NTM [10], focus on learning the entire process of memory interaction (read/write operations), and thus the degree of freedom (or capability) of the model is high in solving a given problem. They have been successfully applied to complex tasks such as sorting, sequence copying, and graph traversal. The memory networks of [15, 23, 27] address the QA problems using continuous memory representation similar to the NTM. However, while the NTM leverages both content-based and location-based addressing, they use only the former (content-based) memory interaction. They apply the concept of multi-hops to recurrently read the memory, which results in performance improvement in solving QA problems that require causal reasoning. The work of [18, 30] proposes a key-value memory network that stores information in the form of (key, value) pairs into the external knowledge base. These methods are good at solving QA problems that focus on the content or facts in a context such as WikiMovies [18] and bAbI dataset [26].

The work of [2, 20] deals with how to make the read/write operations scalable with extremely large amount of memory. Chandar *et al*. [2] propose to organize memory hierarchically, and Rae *et al*. [20] make read and write operations sparse, thereby increasing scalability and reducing the cost of operations.

Compared to all the previous models, our RWMN model is explicitly equipped with learnable read/write networks of CNNs, which are specialized in storing and utilizing more abstract information, such as relationships between characters, reasons for characters' specific behaviors, as well as understanding of facts in a given story.

Models for MovieQA. Among the models applied to the MovieQA benchmark [24], the end-to-end memory network [23] is the state-of-the-art approach. It splits each movie into shot subshots, and constructs memory slots with video and subtitle features. It then uses content-based addressing to attend on the information relevant to a given question. Recently, Wang and Jiang [25] present the compare-aggregate framework for word-level matching to measure the similarity of sentences. However, it is applied to only a single task (plot synopses) of MovieQA.

There have been also several studies to solve Video QA tasks in other datasets, such as LSMDC [22], MSR-VTT [28], and TGIF-QA [13], which mainly focus on understanding short video clips, and answering about factual elements in the clips. Yu *et al*. [29] achieve compelling performance in video captioning, video QA, and video retrieval by constructing an end-to-end trainable concept-word-detector along with vision-to-language models.

3. Read-Write Memory Network (RWMN)

Figure 2 shows the overall structure of our RWMN. The RWMN is trained to store the movie content with proper representation in the memory, extract relevant information from memory cells in response to a given query, and select correct answer from five choices.

Based on the QA format of MovieQA dataset [24], the input of the model is (i) a sequence of video segment and subtitle pairs $S_{movie} = \{(v_1, s_1), ..., (v_n, s_n)\}$ for the whole movie, which takes about 2 hours ($n \sim 1,558$ on average), (ii) a question q for the movie, and (iii) five answer candidates $a = \{a_1, ..., a_5\}$. In the video+subtitle task of MovieQA, for example, each s_i is a dialog sentence of a character, and $v_i = \{v_{i1}, ..., v_{im}\}$ is a video subshot (*i.e*. a set of frames) sampled at 6 fps that are temporally aligned with s_i. The output is a confidence score vector over the five answer candidates.

In the following, we explain the architecture according to information flow, from movie embedding to answer selection via write/read networks.

3.1. Movie Embedding

We convert each subshot v_i and text sentence s_i into feature representation as follows. For each frame $v_{ij} \in v_i$, we first obtain its feature $\mathbf{v}_{ij}$ by applying the ResNet-152 [11] pretrained on ImageNet [4]. We then mean-pool over all frames as $\mathbf{v}_i = \sum_j \mathbf{v}_{ij} \in \mathbb{R}^{7\times 7\times 2,048}$, as a representation of the subshot v_i. For each sentence s_i, we first divide the sentence into words, apply the pretrained Word2Vec [17], and then mean-pool with the position encoding (PE) [23] as $\mathbf{s}_i = \sum_j \mathrm{PE}(\mathbf{s}_{ij}) \in \mathbb{R}^{300}$.

Finally, to obtain a multimodal space embedding of $\mathbf{v}_i$ and $\mathbf{s}_i$, we use the Compact Bilinear Pooling (CBP) [6] as

$$\mathbf{E}[i] = \mathrm{CBP}(\mathbf{v}_i, \mathbf{s}_i) \in \mathbb{R}^{4,096}. \tag{1}$$

We perform this procedure for all n pairs of subshots and text, resulting in a 2D movie embedding matrix $\mathbf{E} \in \mathbb{R}^{n\times 4,096}$, which is the input of our *write network*.

3.2. The Write Network

The write network takes a movie embedding matrix $\mathbf{E}$ as an input and generates a memory tensor $\mathbf{M}$ as output. The write network is motivated by that when human understands a movie, she does not remember it as a simple sequence of speech and visual content, but rather ties together several adjacent utterances and scenes in a form of events or episodes. That is, each memory cell needs to associate neighboring movie embeddings, instead of storing each of n movie embedding separately. To implement this idea of jointly storing adjacent embeddings into every slot, we exploit a convolutional neural network (CNN) as the write network. We experimentally confirm the following CNN design after thorough tests, by varying the dimensions, depths, strides of convolution layers.

To the movie embedding $\mathbf{E} \in \mathbb{R}^{n\times 4,096}$, we first apply a fully connected (FC) layer with parameter $\mathbf{W}_c \in \mathbb{R}^{4,096\times d}$, $\mathbf{b}_c \in \mathbb{R}^d$ to project each $\mathbf{E}[i]$ into a d-dimensional vector. The FC layer reduces the dimension of $\mathbf{E}$ in order to equalize the dimensions of query embedding and answer embedding, which is also beneficial to reduce the number of required convolution operations later. We then use a convolution layer consisting of a filter $\mathbf{w}^w_{conv} \in \mathbb{R}^{f^w_v \times f^w_h \times 1 \times f^w_c}$, whose vertical and horizontal filter size is $f^w_v = 40$, $f^w_h = d$, the number of filter channel is $f^w_c = 3$ and strides are $s^w_v = 30$ and $s^w_h = 1$, respectively:

$$\mathbf{M} = \mathrm{ReLU}(\mathrm{conv}((\mathbf{E}\mathbf{W}_c + \mathbf{b}_c), \mathbf{w}^w_{conv}, \mathbf{b}_w)) \tag{2}$$

where conv (input, filter, bias) indicates the convolution layer, $\mathbf{b}_w \in \mathbb{R}^{f^w_c}$ is a bias, and ReLU indicates the element-wise ReLU activation [19]. Finally, the generated memory is $\mathbf{M} \in \mathbb{R}^{m\times d\times 3}$, where $m = \lfloor((n-1)/s^w_v + 1)\rfloor$.

Note that the write network can employ multiple convolutional layers. If the number of layers is ν_w, then we obtain $\mathbf{M}$ by recursively applying

$$\mathbf{M}^{(l+1)} = \mathrm{ReLU}(\mathrm{conv}(\mathbf{M}^{(l)}, \mathbf{w}^{w(l)}_{conv}, \mathbf{b}^{(l)}_w)) \tag{3}$$

from $l = 1 \ldots, \nu_w - 1$. In section 4, we will report the result of ablation study to find out the best-performing ν_w.

3.3. The Read Network

The *read network* takes a question q and then generate answer from a compatibility between q and $\mathbf{M}$.

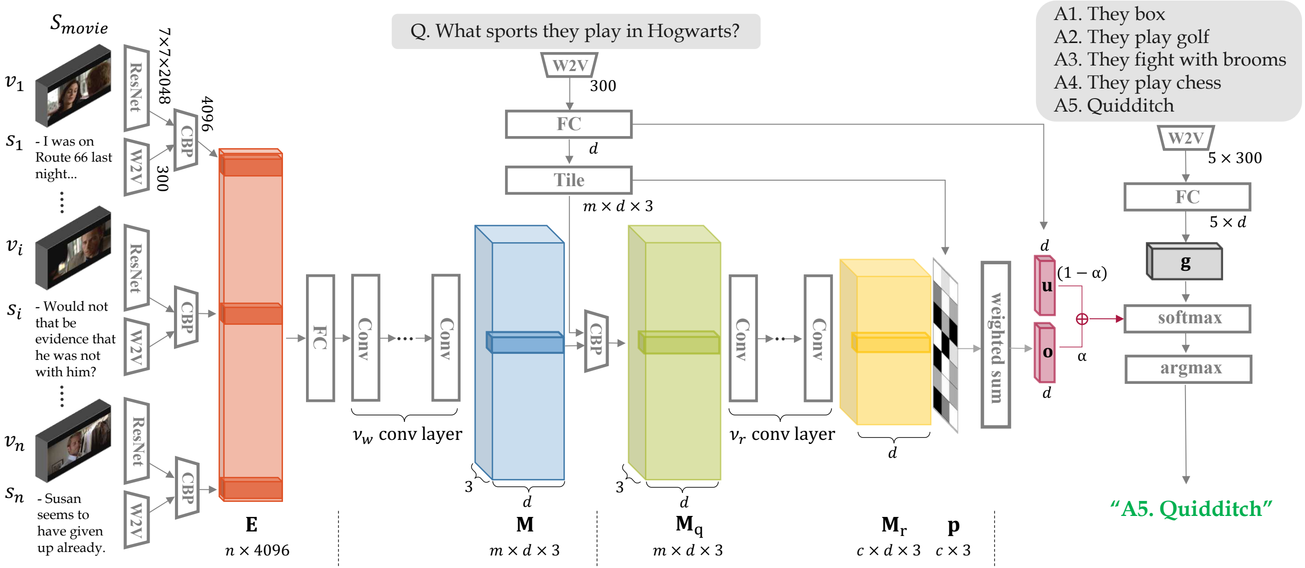
Question embedding. We embed the question sentence q as follows. We first obtain the Word2Vec vector [17] $\mathbf{q}$ as done in section 3.1, and then project it as follows.

$$\mathbf{u} = \mathbf{W}_q\mathbf{q} + \mathbf{b}_q \tag{4}$$

where parameters are $\mathbf{W}_q \in \mathbb{R}^{d\times 300}$ and $\mathbf{b}_q \in \mathbb{R}^d$.

Next the read network takes the memory $\mathbf{M}$ and the query embedding $\mathbf{u}$ as input, and generates the confidence score vector $\mathbf{o} \in \mathbb{R}^d$ as follows.

Query-dependent memory embedding. We first transform the memory $\mathbf{M}$ to be query-dependent. Its intuition is that, according to the query, different types of information must be retrieved from the memory slots. For example, for the *Harry Potter* movie, suppose that one memory slot contains the information about a particular scene where *Harry is chanting magic spells*. This memory slot should be read



(a) Movie Embedding (b) Write Network (c) Read Network (d) Question Answering

Figure 2. Illustration of the proposed *Read-Write Network*. (a) The multimodal movie embedding $\mathbf{E}$ is obtained using the ResNet feature and the Word2Vec representation from movie subshots and subscripts (section 3.1). (b) The write memory $\mathbf{M}$ abstracts higher-level sequential information through multiple convolution layers (section 3.2). (c) The query-dependent memory $\mathbf{M}_q$ is obtained via the Compact Bilinear Pooling (CBP) between the query and each slot of $\mathbf{M}$, and then the read memory $\mathbf{M}_r$ is constructed through convolution layers (section 3.3). (d) Finally, the answer with the highest confidence score is chosen out of five candidates (section 3.4).

differently according to two different questions Q_1: *What color is Harry wearing?* and Q_2: *Why is Harry chanting magic spells?* In section 4, we will empirically show the effectiveness of this question-dependent memory update.

To transform the memory $\mathbf{M}$ into a query-dependent memory $\mathbf{M}_q \in \mathbb{R}^{m\times d\times 3}$, we apply the CBP [6] between each memory cell of $\mathbf{M}$ and the query embedding $\mathbf{u}$ as

$$\mathbf{M}_q[i,:,j] = \mathrm{CBP}(\mathbf{M}[i,:,j],\mathbf{u}) \tag{5}$$

for all $i = 1,\cdots,m$, and $j = 1,2,3$.

Convolutional memory read. As done in the write network, we also leverage a CNN to implement the read network. Our intuition is that, for correctly answering the question of movie understanding, it is important to connect and relate a series of scenes as a whole. Therefore, we use the CNN architecture to access chunks of sequential memory slots. We obtain the reconstructed memory $\mathbf{M}_r$ by applying convolutional layers with a filter $\mathbf{w}^r_{conv} \in \mathbb{R}^{f^r_v\times f^r_h\times 3\times f^r_c}$ whose vertical and horizontal filter size is $f^r_v = 3, f^r_h = d$, the number of filter channel is $f^r_c = 3$ and strides are $s^r_v = 1$, $s^r_h = 1$, respectively. Finally, the reconstructed memory is $\mathbf{M}_r \in \mathbb{R}^{c\times d\times 3}$ with $c = \lfloor(m-1)/s^r_v + 1\rfloor$:

$$\mathbf{M}_r = \mathrm{ReLU}(\mathrm{conv}(\mathbf{M}_q,\mathbf{w}^r_{conv},\mathbf{b}_r)) \tag{6}$$

where $\mathbf{b}_r \in \mathbb{R}^3$ is a bias term. As in the write network, the read network can also have a ν_r number of stacks of convolutional layers; the formulation is the same with Eq.(3) only except replacing $\mathbf{M},\mathbf{w}^w_{conv},\mathbf{b}_w$ with $\mathbf{M}_r,\mathbf{w}^r_{conv},\mathbf{b}_r$, respectively. We will also report the results of ablation study about different ν_r in section 4.

3.4. Answer Selection

Next we compute the attention matrix $\mathbf{p} \in \mathbb{R}^{c\times 3}$ through applying the softmax to the dot product between the query embedding $\mathbf{u}$ and each cell of memory $\mathbf{M}_r$:

$$\mathbf{p}[i,j] = \mathrm{softmax}(\mathbf{M}_r[i,:,j]\cdot\mathbf{u}) \tag{7}$$

where $\cdot$ indicates the dot product. Finally, the output vector $\mathbf{o} \in \mathbb{R}^d$ is obtained through a weighted sum between each memory cell of $\mathbf{M}_r$ and the attention vector $\mathbf{p}$:

$$\mathbf{o}[i] = \sum_{j=1}^{c}\sum_{k=1}^{3}\mathbf{M}_r[j,i,k]\mathbf{p}[j,k]. \tag{8}$$

Next we obtain the embedding of five answer candidate sentences $\{a\}$ as done for the question in Eq.(4) with sharing the parameters $\mathbf{W}_q$ and $\mathbf{b}_q$. As a result, we compute the embedding of answer candidates $\mathbf{g} \in \mathbb{R}^{5\times d}$.

We compute the confidence vector $\mathbf{z} \in \mathbb{R}^5$ by finding the similarity between $\mathbf{g}$ and the weighted sum of $\mathbf{o}$ and $\mathbf{u}$.

$$\mathbf{z} = \mathrm{softmax}((\alpha\mathbf{o} + (1-\alpha)\mathbf{u})^T\mathbf{g}), \tag{9}$$

Story sources	# movie	# QA pairs
Videos and subtitles	140	6,462
Subtitles	408	14,944
DVS	60	2,446
Scripts	199	7,810
Plot synopses	408	14,944

Table 1. The number of movies and QA pairs according to data sources in the MovieQA dataset [24].

where $\alpha \in [0, 1]$ is a trainable parameter. Finally, we predict the answer y with the highest confidence score: $y = \text{argmax}_{i\in[1,5]}(\mathbf{z}_i)$.

3.5. Training

For training of our model, we minimize the softmax cross-entropy between the prediction $\mathbf{z}$ and the groundtruth one-hot vector $\mathbf{z}_{gt}$. All training parameters are initialized with the Xavier method [7]. Experimentally, we select the Adagrad [5] optimizer with a mini-batch size of 32, a learning rate of 0.001, and an initial accumulator value of 0.1. We train our model up to 200 epochs, although we actively use the early stopping to avoid overfitting due to the small size of the MovieQA dataset. We repeat training each model with 12 different random initializations, and select the one with the lowest cost.

4. Experiments

We evaluate the proposed RWMN model for all the tasks of MovieQA benchmark [24]. We defer more experimental results and implementation details to the supplementary file.

4.1. MovieQA Tasks and Experimental Setting

As summarized in Table 1, MovieQA dataset [24] contains 408 movies and 14,944 multiple choice QA pairs, each of which consists of five answer choices with only one correct answer. The dataset provides with five types of story sources associated with the movies: videos, subtitles, DVS, scripts, and plot synopses, based on which the MovieQA challenge hosts 6 subtasks, according to which sources of information are differently used: (i) video+subtitle, (ii) subtitles only, (iii) DVS only, (iv) scripts only, (v) plot synopses only, and (vi) open-ended. That is, there are one video-text QA task, and four text-only QA tasks, and one open-end QA task with no restriction on additional story sources. We strictly follow the test protocols of the challenge, including training/validation/test split and evaluation metrics. More details of the dataset and rules are available in [24] and its homepage[1].

Among six tasks, we discuss our results with more focus on the video+subtitle task, because it is the only VQA task that requires both video and text understanding, whereas the other tasks are text-only. We weight less on the plot synopses only task, since plot synopses are given with a question, and all the QA pairs are generated from plot synopses, this task can be tackled using simple word/sentence matching algorithms (with little movie understanding), achieving a very high accuracy of 77.63%.

Methods	Video+Subtitle	
	val	test
OVQAP	–	23.61
Simple MLP	–	24.09
LSTM + CNN	–	23.45
LSTM + Discriminative CNN	–	24.32
VCFSM	–	24.09
DEMN	–	29.97
MEMN2N [24]	34.20	–
RWMN-noRW	34.20	–
RWMN-noR	36.50	–
RWMN-noQ	38.17	–
RWMN-noVid	37.20	–
RWMN	**38.67**	**36.25**
RWMN-bag	38.37	35.69
RWMN-ensemble	38.30	–

Table 2. Performance comparison for the video+subtitle task on MovieQA public validation/test dataset. (–) means that the method does not participate on the task. Baselines include DEMM (Deep embedded memory network), OVQAP (Only video question answer pairs) and VCFSM (Video clip features with simple MLP).

We solve the video+subtitle task using the proposed RWMN model in Figure 2. For the four text-only QA tasks, no visual sources $\{v_1, ..., v_n\}$ are given, thus we use $\{s_1, ..., s_n\}$ only to construct the movie embedding $\mathbf{E}$ of Eq.(1) without the CBP. Except this, we use the same RWMN model to solve four text-only QA tasks.

4.2. Baselines

We compare the performance of our approach with those of all the methods proposed in the original MovieQA paper [24] or in the official MovieQA leaderboard[2]. We describe the baseline names in the caption of each result table.

In order to measure the effects of key components of the RWMN, we experiment with five variants: (i) (RWMN-noRW) model without read/write networks, (ii) (RWMN-noR) model with only the write network, (iii) (RWMN-noQ) model without query-dependant memory embedding, (iv) (RWMN-noVid) model trained without using videos to quantify the importance of visual input, and (v) (RWMN) model with both write/read networks.

We also test two ensemble versions of our model. Since the MovieQA dataset size is relatively small compared to task difficulty (*e.g.* 4,318 training QA examples in video+subtitle category), models often suffer from severe

[1] http://movieqa.cs.toronto.edu/.

[2] http://movieqa.cs.toronto.edu/leaderboard/ as of the ICCV2017 submission deadline (March 27, 2017 23:59 GMT).

Method	Subtitle		Script		DVS		Plot Synopses		Open-end
	val	test	val	test	val	test	val	test	test
MEMN2N [24]	38.0	36.9	42.3	37.0	33.0	**35.0**	40.6	38.4	–
SSCB-W2V [24]	24.8	23.7	25.0	24.4	24.8	24.9	45.1	45.6	–
SSCB-TF-IDF [24]	27.6	26.5	26.1	23.9	24.5	23.3	48.5	47.4	–
SSCB Fusion [24]	27.7	–	28.7	–	24.8	–	56.7	56.7	–
CNN Word Matching [25]	–	–	–	–	–	–	**72.1**	72.9	–
Convnet Fusion (TF-IDF + Word2Vec)	–	–	–	–	–	–	–	**77.6**	–
Longest Answer	–	–	–	–	–	–	–	–	25.6
RWMN	**40.4**	**38.5**	**44.0**	**39.4**	**40.0**	34.2	37.0	34.8	**36.6**

Table 3. Performance comparison for all the tasks on MovieQA public validation/test dataset. (–) indicates that the method does not participate on the task. The description of baselines with no reference can be found in the MovieQA leaderboard.

overfitting, which the ensemble methods can mitigate. The first (RWMN-bag) is a bagged version of our approach, in which we independently learn RWMN models on 30 bootstrapped datasets, and obtain the averaged prediction. The second (RWMN-ensemble) is a simple ensemble, in which we independently train 20 models with different random initializations, and compute the average prediction.

4.3. Quantitative Results

We below report the results of each method on the validation and test sets, both of which are not used for training at all. While the original MovieQA paper [24] reports the results on the validation set only, the official leaderboard shows the performance on the test set only, for which groundtruth answers are not observable and the evaluation is performed through the evaluation server. The test submission to the server is limited to once every 72 hours.

As of the ICCV2017 submission deadline, our RWMN achieves the best performance for *four* out of five tasks in the validation set, and *four* out of six tasks in the test set.

Results of VQA task. Table 2 compares the performance of our RWMN model with those of baselines for the video+subtitle task. We observe that RWMN achieves the best performance on both validation and test sets. For example, in the test set, RWMN attains 36.25%, which is significantly better than the runner-up DEMN of 29.97%.

As expected, the RWMN with both read/write networks is the best among our variants on both validation and test sets. It implicates that read/write networks play a key role in improving movie understanding. For example, the RWMN-noR with only write network attains higher performance than the RWMN-noRW, which has similar or lower performance than other existing models. The RWMN-noQ without question-dependent memory embedding also underperforms the normal RWMN, which shows that the memory update according to the question is indeed helpful to select a more relevant answer to the question. Finally, the RWMN-noVid is not as good as the RWMN, meaning that our RWMN successfully exploits both full videos and subtitles for training. Interestingly, the ensemble methods of our model, RWMN-bag and RWMN-ensemble, slightly underperform the single model RWMN.

Results of text-only tasks. Table 3 shows the results on the validation and test sets for text-only categories (*i.e*. subtitle only, DVS only, script only, plot synopses only). For the open-end task, we simply use the plot synopses version of our method, which outperforms the only trivial baseline for the test set (*i.e*. selecting the longest answer choice).

Our RWMN achieves the best performance in all tasks except for DVS-test set and plot synopses task. We also observe that the ensemble methods hardly improve the performance of our method noticeably. As discussed before, the memory network approaches including our RWMN and MEMN2N are not outstanding in the plot synopses only category. It is mainly due to that the queries and answer choices are made directly from the plot sentences, and thus, this task can be tackled better by word/sentence matching methods with little story comprehension. In addition, each plot synopsis consists of about 35 sentences on average as a summary of a movie, which is much shorter than other data types, for examples, about 1,558 sentences of subtitles per movie. Therefore, the memory abstraction by our method becomes less critical to solve the problems in this category.

One important difference between the four text-only tasks is that each story source has a different n (*i.e*. the number of sentences), and thus the density of information contained in each sentence is also different. For example, the average n of the scripts is about 2,877 per movie, while the average n of DVS is about 636; thus, each sentence in the script contains low-level details, while each sentence in the DVS contain high-level and abstract content. Given that the performance improvement by our RWMN is more significant in the DVS only task (*e.g*. RWMN: 40.0 and MEMN2N: 33.0), it can be seen that our proposal to read/write networks may be more beneficial to understand and answer high-level and abstract content.

4.4. Ablation Results

We experiment the performance variation according to the structure of CNNs in the write/read networks. Among hyperparameters of the RWMN, the following three combinations have significant effects on the performance of the model; i) conv-filter/stride sizes of the write network (f^w_v, s^w_v, f^w_c), ii) conv-filter/stride sizes of the read network (f^r_v, s^r_v, f^r_c), and iii) number of read/write CNN layers ν_r, ν_w. Regarding the convolutions, the larger the convolution filter sizes, the more memories are read/written as a chunk. Also, as the stride size decreases or the number of output channels increases, the total number of memory blocks increases.

Table 4 summarizes the performance variation on the video+subtitle task according to different combinations of these three hyperparameters. We make several observations from the results. First, as the number of CNN layers in read/write network increases, the capacity of memory interaction may increase as well; yet the performance becomes worsen. Presumably, the main reason may be overfitting due to a relative small dataset size of MovieQA as discussed. It is hinted by our results that the two-layer CNN is the best for training performance, while the one-layer CNN is the best for validation. Second, we observe that there is no absolute magic number of how many memory slots should be read/written as a single chunk and how many strides the memory controller moves. If the stride height is too small or too large compared to the height of a convolution filter, the performance decreases. It means that the performance can be degraded when too much information is read/written as a single abstracted slot, when too much information is overlapped in adjacent reads/writes (due to a small stride), or when the information overlap is too coarse (due to a high stride). We present more ablation results to the supplementary file.

Figure 3 compares between the MEMN2N [24] and our RWMN model according to question types in the video+subtitle task. We examine the results of six question types, according to what starting word is used in the question: *Who, Where, When, What, Why*, and *How*. Usually, *Why* questions require abstraction and high-level reasoning to answer correctly (*e.g. Why did Harry end his relationship with Helen?, Why does Michael depart for Sicily?*). On the other hand, *Who* and *When* questions primarily deal with factual elements (*e.g. Who is Harry's girlfriend?, When does Grissom plan to set up Napier to be murdered?*). Compared to the MEMN2N [24], our RWMN shows higher performance enhancement in the questions starting with *Why*, which may implicate the superiority of the RWMN to deals with high-level reasoning questions.

# Layers		Write network	Read network	Acc.
ν_w	ν_r	$(f^w_{vi}, s^w_{vi}, f^w_{ci})$	$(f^r_{vi}, s^r_{vi}, f^w_{ri})$	
0	0	–	–	34.2
1	0	(40,7,1)	–	33.9
1	0	(40,30,3)	–	36.5
1	1	(40,30,3)	(3,1,1)	**38.6**
1	1	(40,60,3)	(3,1,1)	33.6
2	1	(40,10,3), (10,5,3)	(3,1,1)	37.2
2	1	(5,3,1), (5,3,1)	(3,1,1)	37.3
2	2	(4,2,1), (4,2,1)	(3,1,1), (3,1,1)	36.9
2	2	(4,2,1), (4,2,1)	(4,2,1), (4,2,1)	37.3
3	1	(10,3,3), (40,3,3), (100,3,3)	(3,1,1)	35.1
3	1	(40,3,3), (10,3,3), (10,3,3)	(3,1,1)	37.9
3	1	(40,3,3), (40,3,3), (40,3,3)	(3,1,1)	35.7
3	1	(100,3,3), (40,3,3), (10,3,3)	(3,1,1)	35.8

Table 4. Performance of the RWMN on the video+subtitle task, according to the structure parameters of write/read networks. $\nu_{w/r}$: the number of layers for write/read networks, $(f^{w/r}_{vi}, s^{w/r}_{vi}, f^{w/r}_{ci})$: the height and the stride of convolution filters, and the number of output channels.

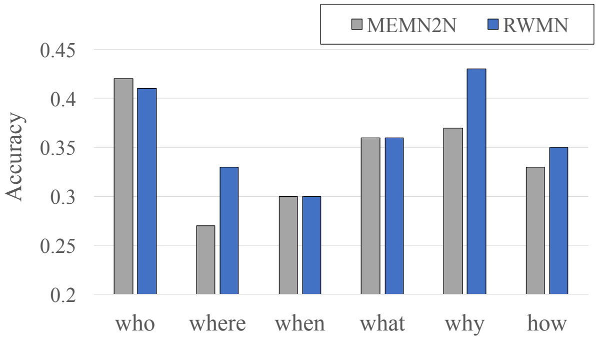
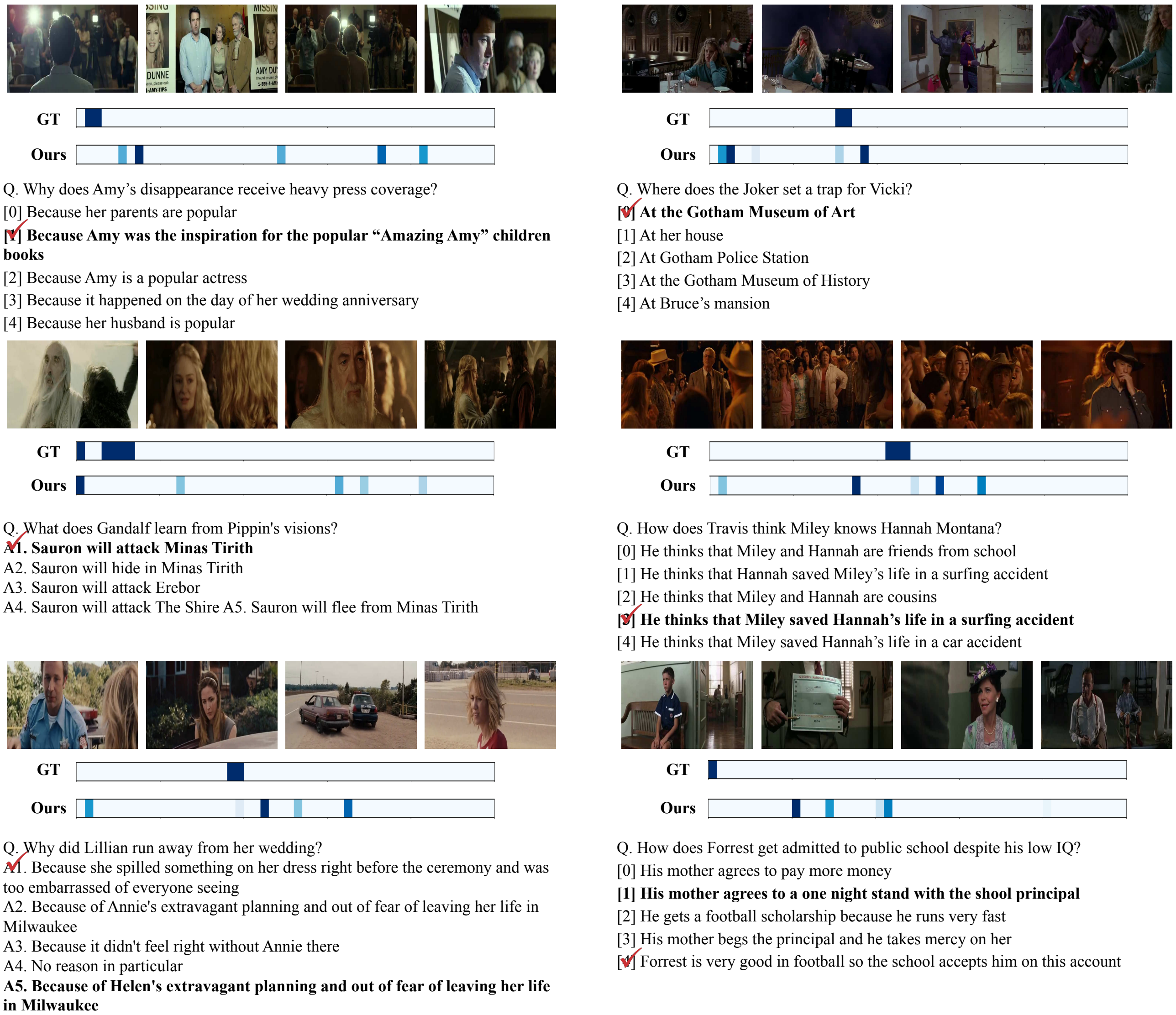


Figure 3. Accuracy comparison between RWMN and the MEMN2N [24] baseline on the video+subtitle task according to question types. The RWMN leads higher improvement for *Why* questions that often require abstract and high-level understanding.

4.5. Qualitative Results

Figure 4 illustrates selected qualitative examples of video+subtitle problems solved by our methods, including four success and two near-miss cases. In each example, we present a sampled query video, a question, and five answer choices in which groundtruth is in bold and our model's selection is red checked. We also show on which parts our RWMN attends over entire movies, along with the groundtruth (GT) attention maps indicating the temporal locations of the clips where the question is actually generated, provided by the dataset. As examples show, movie question answering is highly challenging, and sometimes is not easy even for human.

Our predicted attention often agrees well with the GT; the RWMN can implicitly learn where to place its attention in a very long movie for answering, although such information is not available for training. However, sometimes the RWMN can find correct answers even with the attention mismatch with the GT. It is due to that the MovieQA dataset also includes many questions that are hardly solvable with only attending on the GT parts. That is, some questions re-



Q. Why does Amy's disappearance receive heavy press coverage?
[0] Because her parents are popular
✓ **[1] Because Amy was the inspiration for the popular "Amazing Amy" children books**
[2] Because Amy is a popular actress
[3] Because it happened on the day of her wedding anniversary
[4] Because her husband is popular

Q. Where does the Joker set a trap for Vicki?
✓ **[0] At the Gotham Museum of Art**
[1] At her house
[2] At Gotham Police Station
[3] At the Gotham Museum of History
[4] At Bruce's mansion

Q. What does Gandalf learn from Pippin's visions?
✓ **A1. Sauron will attack Minas Tirith**
A2. Sauron will hide in Minas Tirith
A3. Sauron will attack Erebor
A4. Sauron will attack The Shire A5. Sauron will flee from Minas Tirith

Q. How does Travis think Miley knows Hannah Montana?
[0] He thinks that Miley and Hannah are friends from school
[1] He thinks that Hannah saved Miley's life in a surfing accident
[2] He thinks that Miley and Hannah are cousins
✓ **[3] He thinks that Miley saved Hannah's life in a surfing accident**
[4] He thinks that Miley saved Hannah's life in a car accident

Q. Why did Lillian run away from her wedding?
✓ A1. Because she spilled something on her dress right before the ceremony and was too embarrassed of everyone seeing
A2. Because of Annie's extravagant planning and out of fear of leaving her life in Milwaukee
A3. Because it didn't feel right without Annie there
A4. No reason in particular
A5. Because of Helen's extravagant planning and out of fear of leaving her life in Milwaukee

Q. How does Forrest get admitted to public school despite his low IQ?
[0] His mother agrees to pay more money
[1] His mother agrees to a one night stand with the shool principal
[2] He gets a football scholarship because he runs very fast
[3] His mother begs the principal and he takes mercy on her
✓ [4] Forrest is very good in football so the school accepts him on this account

Figure 4. Qualitative examples of MovieQA video+subtitle problems solved by our methods (success cases in the top two rows, and failure cases in the last row). Bold sentences are groundtruth answers and red check symbols indicate our model's selection. In each example, we also show on which parts our RWMN model attend over entire movie. The attention by the RWMN often matches well with the groundtruth (GT) where the question is actually generated.

quire understanding the relationship between characters or progress of event development, for which attending beyond GT parts is necessary.

5. Conclusion

We proposed a new memory network model named Read-Write Memory Network (RWMN), whose key idea is to propose the CNN-based read/write network that enable the model to have highly-capable and flexible read/write operations. We empirically validated that the proposed read/write networks indeed improve the performance of visual question answering tasks for large-scale, multimodal movie story understanding. Specifically, our approach achieved the best accuracies in multiple tasks of MovieQA benchmark, with a significant improvement on visual QA task. We believe that there are several future research directions that go beyond this work. First, we can apply our approach to other QA tasks that require complicated story understanding. Second, we can explore better video and text representation methods beyond ResNet and Word2Vec.

Acknowledgements. This research is partially supported by SK Telecom and Basic Science Research Program through National Research Foundation of Korea (2015R1C1A1A02036562). Gunhee Kim is the corresponding author.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8M: A Large-scale Video Classification Benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 1

[2] S. Chandar, S. Ahn, H. Larochelle, P. Vincent, G. Tesauro, and Y. Bengio. Hierarchical Memory Networks. *arXiv preprint arXiv:1605.07427*, 2016. 2

[3] 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. In *EMNLP*, 2014. 1

[4] 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. 3

[5] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *JMLR*, pages 2121–2159, 2011. 5

[6] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *EMNLP*, 2016. 3, 4

[7] X. Glorot and Y. Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *AISTATS*, 2010. 5

[8] A. Graves, G. Wayne, and I. Danihelka. Neural Turing Machines. *arXiv preprint arXiv:1410.5401*, 2014. 1, 2

[9] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, et al. Hybrid Computing Using a Neural Network with Dynamic External Memory. *Nature*, 538:471–476, 2016. 1, 2

[10] C. Gulcehre, S. Chandar, K. Cho, and Y. Bengio. Dynamic Neural Turing Machine with Soft and Hard Addressing Schemes. In *ICLR*, 2017. 1, 2

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 3

[12] S. Hochreiter and J. Schmidhuber. Long Short-term Memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[13] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. In *CVPR*, 2017. 2

[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. 1

[15] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. *ICML*, 2016. 1, 2

[16] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent Neural Network Based Language Model. In *Interspeech*, 2010. 1

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *NIPS*, 2013. 3

[18] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston. Key-value Memory Networks for Directly Reading Documents. In *EMNLP*, 2016. 2

[19] V. Nair and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*, 2010. 3

[20] J. Rae, J. J. Hunt, I. Danihelka, T. Harley, A. W. Senior, G. Wayne, A. Graves, and T. Lillicrap. Scaling Memory-Augmented Neural Networks with Sparse Reads and Writes. In *NIPS*, 2016. 2

[21] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ Questions for Machine Comprehension of Text. In *EMNLP*, 2016. 1

[22] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie Description. *IJCV*, 123(1):94–120, 2017. 1, 2

[23] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-End Memory Networks. In *NIPS*, 2015. 1, 2, 3

[24] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. MovieQA: Understanding Stories in Movies through Question-answering. In *CVPR*, 2016. 1, 2, 3, 5, 6, 7

[25] S. Wang and J. Jiang. A Compare-Aggregate Model for Matching Text Sequences. In *ICLR*, 2017. 2, 6

[26] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards AI-complete Question Answering: A Set of Prerequisite Toy Tasks. *arXiv preprint arXiv:1502.05698*, 2015. 1, 2

[27] J. Weston, S. Chopra, and A. Bordes. Memory Networks. *ICLR*, 2015. 1, 2

[28] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*, 2016. 1, 2

[29] Y. Yu, H. Ko, Jongwook, and G. Kim. End-to-end Concept Word Detection for Video Captioning, Retrieval, and Question Answering. In *CVPR*, 2017. 1, 2

[30] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic Key-Value Memory Network for Knowledge Tracing. In *WWW*, 2017. 2