

A2A: Attention to Attention Reasoning for Movie Question Answering

Chao-Ning Liu¹, Ding-Jie Chen², Hwann-Tzong Chen¹, and Tyng-Luh Liu²

¹ Department of Computer Science, National Tsing Hua University, Taiwan

² Institute of Information Science, Academia Sinica, Taiwan

Abstract. This paper presents the Attention to Attention (A2A) reasoning mechanism to address the challenging task of movie question answering (MQA). By focusing on the various aspects of attention cues, we establish the technique of attention propagation to uncover latent but useful information to the underlying QA task. In addition, the proposed A2A reasoning seamlessly leads to effective fusion of different representation modalities about the data, and also can be conveniently constructed with popular neural network architectures. To tackle the out-of-vocabulary issue caused by the diverse language usages in nowadays movies, we adopt the GloVe mapping as a teacher model and establish a new and flexible word embedding based on character n-grams learning. Our method is evaluated on the MovieQA benchmark dataset and achieves the state-of-the-art accuracy for the “Video+Subtitles” entry.

1 Introduction

We aim to address a specific problem of Visual Question Answering (VQA) that is coined as Movie Question Answering (MQA). For a model to deal with a question answering (QA) task, it is expected to have the ability of analyzing visual and textual contents and inferring the most plausible answer to a given question. The MQA task is deemed to be challenging in that the correct answering requires a comprehensive understanding of not only the recognition sub-task (who, where, and when) but also the reasoning sub-task (what, why, and how) via associating the visual content with the textual and vice versa. So far, the popular content analysis approaches mainly comprise word embedding [20, 24] and image embedding [9, 28], and the inferring approaches usually are based on memory networks [16, 18, 21, 23, 25, 29, 31–33] and attention models [2, 6, 15, 19]. We instead design a new attention based model that is able to propagate attention across different segments in a movie sequence to address the MQA problem.

The MovieQA dataset provides online testing for benchmark evaluation of VQA models. We compare our model with others on this collection. The dataset contains 408 movies with standard subtitles, and 140 of them are accompanied with video clips. To analyze and understand such long video sequences, previous strategies commonly rely on adopting the per-frame visual content, successive-frame temporal dependencies, and the subtitles. In contrast, the proposed model

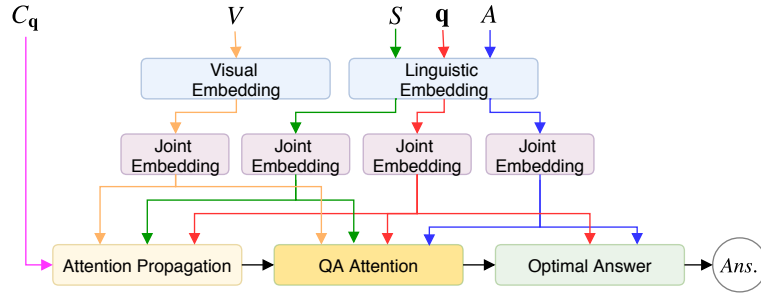


Fig. 1: An overview of the proposed network architecture for MQA. With the provided video data V and subtitle data S , our model leverages the A2A reasoning mechanism, namely, “Attention Propagation” and “QA Attention” to decide, from a set A of five candidate answers, the best answer to question \mathbf{q} .

of attention-to-attention (A2A) reasoning focuses on exploring higher-level attention information about the questions and answers for QA video analysis.

The proposed method aims to explore high-level and multimodality attention mechanisms for addressing QA tasks of movie understanding. We illustrate the overall network architecture of our method in Fig. 1 and characterize the main contributions as follows.

- We propose the attention-to-attention (A2A) reasoning mechanism to distill more attention information for answering questions. The implications are twofold. First, it enables attention propagation to uncover neglected information that may be useful for MQA. Second, the distilled attention aggregates and associates the visual with the textual information from subtitles, questions, and answers.
- We adopt the GloVe mapping [24] as the teacher model to design a new word embedding approach for tackling the out-of-vocabulary issue in MQA.
- We establish a joint embedding approach that simplifies the association learning between the visual and textual modalities.
- Our model achieves the state-of-the-art performance on the “Video+Subtitles” entry of MovieQA benchmark.

2 Related Work

We start with an overview of several popular datasets for visual captioning and question answering, and then briefly discuss the two main trends of solving QA tasks, *i.e.*, memory network and attention model.

2.1 Visual Captioning and Question Datasets

A number of comprehensive datasets have been created for evaluating the machine learning methods that are designed to tackle integrated visual-textual tasks such as visual captioning and visual question answering.

COCO [17] and LSMDC [27] are two widely used datasets for studying the visual captioning problems. The COCO dataset provides up to 330K images with five captions per image for image captioning tasks. For video description tasks, the LSMDC dataset contains 200 movies with aligned description sentences for exploring the way to generate descriptions for movies.

There exist several datasets for studying the question answering tasks concerning text, image, or video. Regarding question answering tasks of pure text, bAbI [32] contains various tasks for evaluating the performance of a question answering system, and SQuAD [25] consists of hundred thousand QAs and 500 articles for studying the reading comprehension. For question answering tasks conducted on images, CLEVR [14], VQA-v1.0 [1], and VQA-v2.0 [8] can be considered. The CLEVR dataset comprises many questions that are explicitly relational, and hence it requires rich relational reasoning to analyze the data. VQA-v1.0 and VQA-v2.0 include images and their corresponding QAs from COCO dataset. The VQA-v1.0 dataset provides evaluation in a multiple-choice setting with additional candidate answers per question. VQA-v2.0 balances the answers to each question for minimizing the effects of dataset-prior learning.

TGIF-QA [13] and MarioQA [22] are conducted for studying video question answering tasks that require temporal reasoning to answer the questions. TGIF-QA dataset provides question answering tasks concerning not only single-image inputs but also spatio-temporal frames. The data are collected from animated Tumblr GIFs. MarioQA dataset is built upon Super Mario video gameplays. The dataset provides videos with multiple events and event-centric questions. There is no extra information to reason answers while analyzing temporal relationship between events in the MarioQA dataset.

This work focuses on another video question answering dataset, MovieQA [30], which provides several data modalities such as video clips, questions, subtitles, descriptive video service (DVS), plot synopsis, and scripts. The tasks in MovieQA is challenging because several questions are about the story and it needs the ability of long-term temporal reasoning, natural language understanding, and scene understanding. Table 1 illustrates a QA example from the MovieQA dataset.

2.2 Memory Network

Storing long sequential information is one key factor for dealing with the MQA problem. Memory networks usually perform read-write operations on an internal representation. Instead of using the traditional recurrent neural networks (RNNs) [10] that store and update the given information into fixed-size hidden units, another solution is to leverage an external memory network [21, 29, 33] that directly “memorizes” much earlier temporal information. Several state-of-the-art approaches, such as bAbI [32], SQuAD [25], and GMemN2N [18] have adopted memory networks to address pure-text question answering tasks.


		
You can see here the Death Star...	the Death Star does have a strong defense mechanism.	The shield must be deactivated...
q: How is the Death Star protected from attack? a₁: By an army of soldiers. a₂: <i>By nuclear weapons.</i> a₃: By an energy shield. a₄: By starfighters. a₅: By Emperor Palpatine’s army.		C_q: a Boolean vector indicating the video frames (subtitles) that are relevant to the task of question answering w.r.t q .

Table 1: Example of MovieQA benchmark. The answer marked in green is the ground truth. The first row presents example images from the movie, the second row shows the corresponding subtitles, and the third row lists the corresponding question, candidate answers, and the answer-required frame barcode C_q .

The approaches LMN [31], DEMN [16], RWMN [23], and SC-MemN2N [4] show the state-of-the-art accuracy on the task of video question answering, LMN proposes static word memory and dynamic subtitle memory. The static word memory stores visual word with static size, and the dynamic subtitle memory use static word memory to generate clips-level representation with subtitle. They also use multiple computational steps (hops) mechanism [29] to refine the memory. DEMN uses a long-term memory component to embed both visual and textual features for storing, and query the features with respect to the question and the answers sequentially. RWMN adopts a convolution based read-write network for allowing highly-capable and flexible read-write operations to construct the long-term memory. Its visual features come from the last average pooling layer of ResNet-152 [9]. SC-MemN2N includes an end-to-end memory network that leverages visual, textual, and acoustic modalities with several grammatical and acoustic constraints in a unified optimization framework.

Notice that, the dynamic subtitle update rule of LMN [31] is similar to the attention mechanism of our approach. The difference is that we use both question and answer for guiding the attention of subtitle, while LMN uses clip-level semantic representation to do the refinement.

2.3 Attention Model

When global features are used to represent the visual contents, irrelevant or noisy information may affect the reasoning. It is possible to use an attention model to address this issue by assigning different importance weights to local features corresponding to partial contents.

Attention models are widely adopted in the task of image question answering [2, 6, 15, 19]. Lu *et al.* [19] design a co-attention model to reason the attention

of image and question jointly. The co-attention model hierarchically reasons the important part of both image and question via a one-dimensional convolutional neural network. Fukui *et al.* [6] propose to use multimodal compact bilinear (MCB) pooling to combine multimodal features. For visual question answering, the MCB module is adopted twice, one for predicting attention over spatial features and the other for combining the question representation with attended representation. Kazemi *et al.* [15] embed image via a ResNet and embed tokenized question via a multi-layer LSTM. The embedded image features and question are then concatenated to predict multiple attention distributions over image features. Anderson *et al.* [2] propose a mechanism that combines bottom-up and top-down attention for calculating object-level attention and salient-region attention. The bottom-up mechanism extracts image regions with associated feature vectors, and the top-down decides the feature weightings.

Attention mechanisms can be used to identify *where to notice* before further reasoning to answer questions in VQA task. The proposed A2A reasoning mechanism guides our question answering model to notice not only inter-segment relations in the spatial-temporal domain but also the association between questions and answers in sentence domain.

3 Our Method

This study considers the MovieQA dataset [30], comprising a set of movies, for evaluating the QA performance. Since for each particular movie the analysis of our method to perform question answering is the same, it is sufficient to restrict the discussion of the proposed formulation and notations for an arbitrary movie, unless explicitly stated otherwise. To begin, the following notations are used to express the various aspects about the data. We decompose a given movie into a set of video clips, $V = \{v_1, v_2, \dots, v_{|V|}\}$ where the collection of corresponding subtitle sets is denoted as $S = \{s_1, s_2, \dots, s_{|S|}\}$ and $|S| = |V|$. For a presented question \mathbf{q} about the movie, the QA task is to choose the *best* answer from a set of five candidates, denoted as $A = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_5\}$. In addition, a Boolean mask $C_{\mathbf{q}}$ is provided to indicate those video clips (and hence the corresponding subtitle sets) that are relevant to carry out the task of question answering with respect to the specific question \mathbf{q} .

3.1 Visual and Linguistic Embedding

We represent visual information of each frame with its B most “salient” objects. To do so, we use Faster-RCNN [26] with the ResNet-101 [9] pre-trained model from TensorFlow object detection API [12] to select those object bounding boxes with the B highest scores among all detected candidates, and extract a feature vector of dimension $d_v = 2048$ for each bounding box from the last average pooling layer in the second stage of Faster-RCNN. The derivation yields $V \in \mathbb{R}^{d_v \times N \times B}$ where N is the total number of frames (also subtitles) over all the video clips of a particular movie, and our current implementation assumes $B = 6$.

To model the linguistic input, we leverage the technique of word embedding to achieve the intended mapping. However, in tackling question answering with movies, out-of-vocabulary (OOV) could be a legitimate concern as the provided subtitles may contain slangs, special names and terms. We resolve the OOV issue by learning a more flexible word embedding based on character n -grams, using GloVe [24] as a teacher model. Denote the collection of words in GloVe as Ω . For each word $\mathbf{w} \in \Omega$, the resulting set of character n -grams is denoted as $G_{\mathbf{w}} = G_{\mathbf{w}}^1 \cup G_{\mathbf{w}}^3 \cup G_{\mathbf{w}}^6$, where 1-gram, 3-gram, and 6-gram tokens are considered in our formulation. Notice that in constructing $G_{\mathbf{w}}^3$ and $G_{\mathbf{w}}^6$, \mathbf{w} is first augmented by adding the special token “<” at the beginning and “>” at the end. Thus assuming \mathbf{w} is of length ℓ , we have $|G_{\mathbf{w}}^3| = \ell$ and $|G_{\mathbf{w}}^6| = \ell - 3$.

Now let the proposed word embedding be ϕ and the one by GloVe be $\tilde{\phi}$. Using the latter as a teacher model, we train a multi-layer perceptron to realize ϕ by minimizing the following loss function:

$$L(\phi) = \sum_{\mathbf{w} \in \Omega} D(\tilde{\phi}(\mathbf{w}), \phi(\mathbf{w})) \quad (1)$$

$$= \sum_{\mathbf{w} \in \Omega} D(\tilde{\phi}(\mathbf{w}), \sum_{\mathbf{g} \in G_{\mathbf{w}}} \phi(\mathbf{g})) \quad (2)$$

where D is defined to be the cosine distance function, *i.e.*, $D(\mathbf{x}, \mathbf{y}) = 1 - \cos(\mathbf{x}, \mathbf{y})$ and $\mathbf{g} \in G_{\mathbf{w}}$ is any of the n -gram tokens yielded by \mathbf{w} . From (1) and (2), we see that the new embedding $\phi(\mathbf{w})$ is obtained by summing over the embeddings of all n -gram tokens of \mathbf{w} . For $\mathbf{w} \in \Omega$, the proposed word embedding ϕ behaves like the GloVe embedding $\tilde{\phi}$. More importantly, it alleviates the OOV problem by integrating the embeddings of n -gram tokens via (2). To achieve sentence embedding, we simply divide each sentence in all provided S, \mathbf{q}, A from the MovieQA dataset to words, and apply ϕ to each of those words. We can then employ *Smooth Inverse Frequency Weighting* scheme [3] to obtain sentence embeddings. With the embedding (linguistic) dimension d_ℓ set to 300, we have $S \in \mathbb{R}^{d_\ell \times N}$, $\mathbf{q} \in \mathbb{R}^{d_\ell \times 1}$, $A \in \mathbb{R}^{d_\ell \times 5}$, and $C_{\mathbf{q}} \in \{0, 1\}^{N \times 1}$, where $C_{\mathbf{q}}$ is a mask indicating those frames of the video clips relevant to \mathbf{q} .

3.2 Joint Embedding

Once we have respectively obtained the visual and linguistic representations, it is useful to investigate the association between the two modalities for more effectively solving the QA task. To this end, we reduce exploring the two embeddings to learning the relatedness of the representations in a *common space*, where a similar idea can be found in addressing image captioning [34] or VQA [11]. Using the notation “dnorm2” to represent taking “ L_2 -normalization” and then “dropout,” we design the following *normalized affine transform* $\mathcal{J} : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2}$ such that

$$\mathcal{J}(\mathbf{x}; d_1, d_2) = \text{dnorm2}((\delta(d_1, d_2)I + W_{\mathbf{x}}) \cdot \mathbf{x} + \mathbf{b}_{\mathbf{x}}) \quad (3)$$

where $\mathbf{x} \in \mathbb{R}^{d_1 \times 1}$, $\mathbf{b}_{\mathbf{x}} \in \mathbb{R}^{d_2 \times 1}$, $W_{\mathbf{x}} \in \mathbb{R}^{d_2 \times d_1}$ and $\delta(d_1, d_2)$ is the Kronecker delta, namely, $\delta(d_1, d_2) = 1$ when $d_1 = d_2$ and 0, otherwise. We implement the

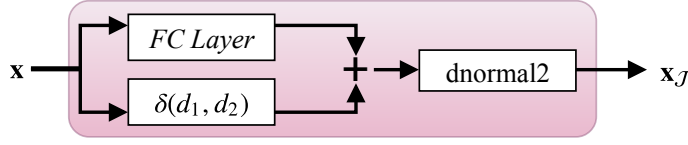


Fig. 2: Normalized affine transform \mathcal{J} : The dropout layer is performed after the L_2 -normalization in the “dnormal2” block. We denote the normalized affine mapping by $\mathbf{x} \xrightarrow{\mathcal{J}} \mathbf{x}_{\mathcal{J}}$ and therefore $X \xrightarrow{\mathcal{J}} X_{\mathcal{J}}$.

transform \mathcal{J} as a single fully-connected layer. In particular, when $d_1 = d_2$, \mathcal{J} does not alter the feature dimension so we can add a short-cut connection in the network to boost the performance. Also note that in (3), dropout and L_2 -normalization are used to regularize the transform \mathcal{J} . Fig. 2 shows the network architecture of the affine transform \mathcal{J} .

With \mathcal{J} in (3), the joint embedding of visual and linguistic representations can be achieved by transforming the visual dimension from $d_v = 2048$ to $d_\ell = 300$. We first apply \mathcal{J} to each visual feature \mathbf{v} of the visual tensor V and obtain the transformed visual tensor as

$$\mathbf{v}_{\mathcal{J}} = \mathcal{J}(\mathbf{v}; d_v, d_\ell) \Rightarrow V_{\mathcal{J}} = \mathcal{J}(V; d_v, d_\ell) \in \mathbb{R}^{d_\ell \times N \times B} \quad (4)$$

where the parameters of \mathcal{J} to be learned are $W_{\mathbf{v}}$ and $\mathbf{b}_{\mathbf{v}}$. On the other hand, the transform \mathcal{J} (now with a short-cut connection) can be applied to the linguistic data S , \mathbf{q} and A , respectively. That is, we consider in turn the three types of linguistic data. For each $\mathbf{x} \in \{\mathbf{s}$ (subtitle), \mathbf{q} (question), \mathbf{a} (answer) $\}$, the transformed linguistic tensors are derived as follows:

$$\mathbf{x}_{\mathcal{J}} = \mathcal{J}(\mathbf{x}; d_\ell, d_\ell) \Rightarrow X_{\mathcal{J}} = \mathcal{J}(X; d_\ell, d_\ell). \quad (5)$$

The above mappings would yield $S_{\mathcal{J}} \in \mathbb{R}^{d_\ell \times N}$, $\mathbf{q}_{\mathcal{J}} \in \mathbb{R}^{d_\ell \times 1}$, and $A_{\mathcal{J}} \in \mathbb{R}^{d_\ell \times 5}$. Analogously, for transforming the linguistic representations with (3), the parameters to be learned in each case are $W_{\mathbf{x}}$ and $\mathbf{b}_{\mathbf{x}}$, for $\mathbf{x} \in \{\mathbf{s}, \mathbf{q}, \mathbf{a}\}$.

The transformed visual tensor $V_{\mathcal{J}} \in \mathbb{R}^{d_\ell \times N \times B}$ accounts for B objects in each image frame. To obtain the final visual representation, denoted as $U_{\mathcal{J}} \in \mathbb{R}^{d_\ell \times N}$, we compute the attention cue $\alpha_{\mathbf{q}}^o$ for each detected object o by

$$\alpha_{\mathbf{q}}^o[j, k] = \text{drelu}(\mathbf{q}_{\mathcal{J}}^\top \cdot V_{\mathcal{J}}[j, k]), \quad \text{for } 1 \leq j \leq N, 1 \leq k \leq B \quad (6)$$

where we use the notation “drelu” to denote the application of two consecutive operations, namely, first taking “relu” and then “dropout”. Then, by weighted summing the B object features in each frame according to the resulting attention $\alpha_{\mathbf{q}}^o$, we have

$$U_{\mathcal{J}}[i, j] = \sum_{k=1}^B V_{\mathcal{J}}[i, j, k] \times \alpha_{\mathbf{q}}^o[j, k]. \quad (7)$$

We now give an interpretation of (7). The attention matrix $\alpha_{\mathbf{q}}^o[j, k]$ reflects the relatedness of the detected object o_k in image frame j to the given question \mathbf{q} . Thus, the final visual representation $U_{\mathcal{J}}[i, j]$ highlights the association between feature i and frame j concerning \mathbf{q} .

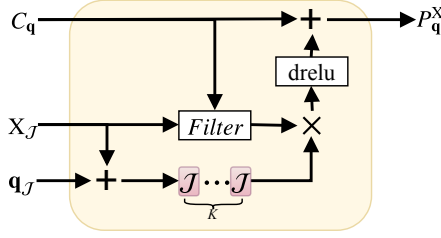


Fig. 3: A2A scheme: Attention propagation is applied to uncover latent but useful information from the data to help answer question \mathbf{q} . “Filter” means the operation to select the subset of the input indicating by mask $C_{\mathbf{q}}$. $X \in \{V, S\}$

3.3 Attention Propagation

For each question \mathbf{q} pertaining to a specific movie in the MovieQA dataset, the provided mask $C_{\mathbf{q}} \in \{0, 1\}^{N \times 1}$ indicates those image frames relevant to answering the question \mathbf{q} . We observed that $C_{\mathbf{q}}$ might not always yield sufficient information to answer the question. It is constructive to augment the provided clues by including other useful information from those neglected by $C_{\mathbf{q}}$. We thus propose an effective A2A scheme called *attention propagation* to augment $C_{\mathbf{q}}$. Without loss of generality, the following discussion focuses on the subtitle information in that the steps for dealing with visual information are similar.

Attention propagation for the linguistic information takes as input a specific transformed question $\mathbf{q}_{\mathcal{J}}$, the transformed subtitles $S_{\mathcal{J}}$, and the clue mask $C_{\mathbf{q}}$ to uncover the propagated subtitle mask $P_{\mathbf{q}}^S$. The propagation process starts by incorporating $\mathbf{q}_{\mathcal{J}}$ into $S_{\mathcal{J}}$. The affine transform \mathcal{J} in (3) is then repeatedly applied to the question-augmented subtitle tensor so that a more flexible representation can be learned, which will be used to compute $P_{\mathbf{q}}^S$. We illustrate the mechanism of attention propagation in Fig. 3, and summarize the steps as follows.

1. Initialize and iterate the question-augmented subtitle tensor with

$$\hat{S}_{\mathbf{q}}^1 = S_{\mathcal{J}} + \mathbf{q}_{\mathcal{J}} \cdot \mathbf{1}^{1 \times N} \text{ and } \hat{S}_{\mathbf{q}}^k = \mathcal{J}(\hat{S}_{\mathbf{q}}^{k-1}; d_{\ell}, d_{\ell}), k = 2, \dots, K.$$

To simplify the notation, the resulting $\hat{S}_{\mathbf{q}}^K$ will be written as $\hat{S}_{\mathbf{q}}$, while the iteration parameter K will be discussed in the experiments.

2. Let $N_{\mathbf{q}}$ be the number of relevant subtitles (frames) filtered by $C_{\mathbf{q}}$, and $S_{\mathbf{q}}^+ \in \mathbb{R}^{d_{\ell} \times N_{\mathbf{q}}}$ be the collection of relevant subtitles. We define the relatedness $F_{\mathbf{q}} \in \mathbb{R}^{N \times N_{\mathbf{q}}}$ of each subtitle to the relevant ones by $F_{\mathbf{q}} = \text{drelu}(\hat{S}_{\mathbf{q}}^{\top} \cdot S_{\mathbf{q}}^+)$.
3. The propagated mask $P_{\mathbf{q}}^S \in \mathbb{R}^{N \times 1}$ can now be computed by

$$P_{\mathbf{q}}^S[i] = \min \left\{ 1, C_{\mathbf{q}}[i] + \frac{1}{N_{\mathbf{q}}} \sum_{j=1}^{N_{\mathbf{q}}} F_{\mathbf{q}}[i, j] \right\} \text{ for } 1 \leq i \leq N. \quad (8)$$

The propagated visual mask $P_{\mathbf{q}}^V \in \mathbb{R}^{N \times 1}$ for the given \mathbf{q} and $C_{\mathbf{q}}$ can be obtained analogously. With the two propagated masks we are now ready to complete our method based on the augmented visual and linguistic information.

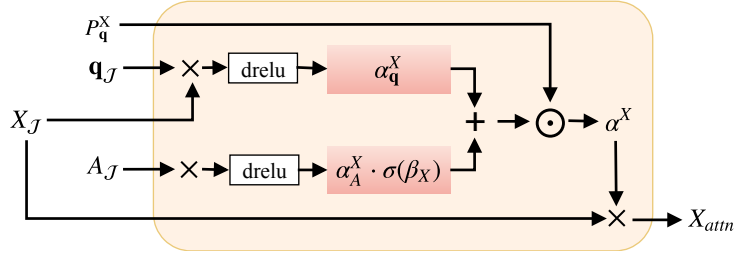


Fig. 4: A2A scheme: QA attention can be obtained via fusing multimodality attention and exploring the augmented mask P_q^X yielded by attention propagation.

3.4 QA Attention

With the visual tensor $U_{\mathcal{J}}$, we can evaluate its question attention $\alpha_{\mathbf{q}}^V \in \mathbb{R}^{N \times 1}$ and the answer attention $\alpha_A^V \in \mathbb{R}^{N \times 5}$. Similarly, we could also compute the question attention $\alpha_{\mathbf{q}}^S$ and the answer attention α_A^S for the subtitle tensor $S_{\mathcal{J}}$. Specifically, we have

$$\alpha_{\mathbf{q}}^V = \text{drelu}(U_{\mathcal{J}}^{\top} \cdot \mathbf{q}_{\mathcal{J}}), \quad \alpha_A^V = \text{drelu}(U_{\mathcal{J}}^{\top} \cdot A_{\mathcal{J}}), \quad (9)$$

$$\alpha_{\mathbf{q}}^S = \text{drelu}(S_{\mathcal{J}}^{\top} \cdot \mathbf{q}_{\mathcal{J}}), \quad \alpha_A^S = \text{drelu}(S_{\mathcal{J}}^{\top} \cdot A_{\mathcal{J}}). \quad (10)$$

We add up the tiled question attention and the answer attention, scaled by the sigmoid output of a learnable variable, and then element-wise multiply the aggregated attention by the respective propagated mask $P_{\mathbf{q}}$. Thus, we can obtain the visual attention $\alpha^V \in \mathbb{R}^{N \times 5}$ and subtitle attention $\alpha^S \in \mathbb{R}^{N \times 5}$ by

$$\alpha^V = (\alpha_{\mathbf{q}}^V \cdot \mathbf{1}^{1 \times 5} + \sigma(\beta_V) \cdot \alpha_A^V) \odot (P_{\mathbf{q}}^V \cdot \mathbf{1}^{1 \times 5}) \quad (11)$$

$$\alpha^S = (\alpha_{\mathbf{q}}^S \cdot \mathbf{1}^{1 \times 5} + \sigma(\beta_S) \cdot \alpha_A^S) \odot (P_{\mathbf{q}}^S \cdot \mathbf{1}^{1 \times 5}) \quad (12)$$

where \odot is the element-wise multiplication operator for tensors, β_V and β_S are two learnable variables, and $\sigma(\cdot)$ is the sigmoid function. (See Fig. 4 for illustration of the architecture.) Finally, we can derive the attention-weighted visual representation V_{attn} and subtitle representation S_{attn} by

$$V_{attn} = U_{\mathcal{J}} \cdot \alpha^V \in \mathbb{R}^{d_{\ell} \times 5}, \quad (13)$$

$$S_{attn} = S_{\mathcal{J}} \cdot \alpha^S \in \mathbb{R}^{d_{\ell} \times 5}. \quad (14)$$

3.5 Optimal Answer Response

By fusing the two modalities of visual and linguistic information, we obtain the overall movie representation $M_{\mathbf{q}} \in \mathbb{R}^{d_{\ell} \times 5}$ with respect to a specific question \mathbf{q} . (Also see Fig. 5 for the network architecture.) We then use “dnormal2” specified in formulating (3) to compute $M_{\mathbf{q}}$ as follows.

$$M_{\mathbf{q}} = \text{dnormal2}(\sigma(\gamma_1) \cdot S_{attn} + (1 - \sigma(\gamma_1)) \cdot \mathbf{q}_{\mathcal{J}} \cdot \mathbf{1}^{1 \times 5} + \sigma(\gamma_2) \cdot V_{attn}) \quad (15)$$

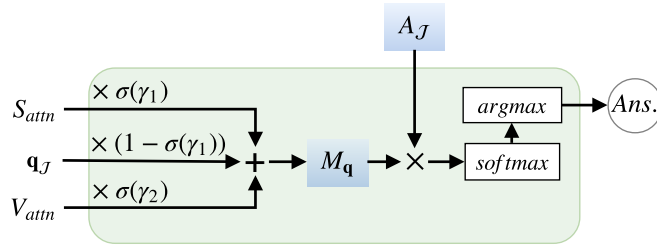


Fig. 5: Illustration of the network architecture to output the MQA responses.

where γ_1 and γ_2 are both learnable variables. Finally, the answer $\mathbf{a}_{j^*} \in A$ to the question \mathbf{q} by our method is given by finding the highest response $R[j^*]$, where

$$j^* = \arg \max_{1 \leq j \leq 5} R[j] = \sum_{i=1}^{d_\ell} M_{\mathbf{q}}[i, j] \times A_{\mathcal{J}}[i, j]. \quad (16)$$

4 Experiments and Discussions

All our experiments are carried out on the MovieQA benchmark dataset [30]. The evaluation of our method contains four parts, including *key components ablation*, *leader board comparison*, *model selection*, and *question types comparison*. Visualization examples of question answering are also included in the results.

Dataset Specification The MovieQA dataset contains 14,944 multi-choice questions related to 408 movies. Each question has five candidate answers of which only one is correct. We focus on the task of “Video+Subtitles,” which contains 6,462 QAs. These QAs are further split into 4,318, 886, and 1,258 for training, validation, and testing, respectively. All the results are measured in accuracy.

Implementation Details To train our model, we use softmax cross entropy as our loss function between response R and one-hot vector \mathbf{a}_{gt} . For general hyperparameter setting, we set the dropout keep rate to 0.9 and the scale of L2-regularizer to 0.01. We use “powersign-ld” [5] as our neural optimizer with 128 decay epochs and a batch size of 1. All model parameters are initialized with Glorot normal initialization [7] and the learning rate is 10^{-3} with linear cosine decay [5]. In validating and testing our method, we use the same ensemble strategy as RWMN [23], which independently trains multiple models for answering, to mitigate the potential overfitting issue on MovieQA due to relatively small dataset size and highly difficult task. To report our results, we average the best accuracy of 10 models with different random initializations on the validation set. As for the test set, we use majority voting by 20 models with different random initializations as an ensemble model, and submit our result to the official test server¹. Both validation and test set are held out from training.

¹ http://movieqa.cs.toronto.edu/new_submission/

Method	Validation (%)	Method	Validation (%)	Test (%)
A2A-noJE	26.44 ± 0.78	A2A-noDropout	41.19 ± 0.40	-
A2A-noProp	40.58 ± 0.30	A2A-noScale	40.01 ± 0.52	-
A2A-noQAattn	33.71 ± 0.31	A2A-noSubtProp	40.28 ± 0.41	-
A2A-noVis	41.22 ± 0.14	A2A-noVisProp	41.05 ± 0.69	41.65
A2A-noSubt	28.53 ± 0.28	A2A	41.66 ± 0.25	41.97
A2A-noL2norm	20.00			

Table 2: Ablation comparison for key components of our A2A method on the validation and test sets of MovieQA benchmark. For validation set, we report accuracy results within the 95% confidence interval. A variant that is not evaluated is marked by (-). Details of the model abbreviations are described in section 4.1.

4.1 Ablation Study on Key Components

We perform an in-depth ablation study on the key components in our method, and report the results in Table 2. The experiment includes 11 variants of A2A.

- (i) (A2A-noJE) model: replacing normalized affine transformation with identity transformation in joint embedding.
- (ii) (A2A-noProp) model: replacing propagated attention with mask C_q .
- (iii) (A2A-noQAattn) model: skipping the use of QA attention.
- (iv) (A2A-noVisual) model: skipping the visual input.
- (v) (A2A-noSubtitle) model: skipping the subtitle input.
- (vi) (A2A-noL2norm) model: skipping L_2 -normalization after each layer.
- (vii) (A2A-noDropout) model: skipping dropout after L_2 -normalization.
- (viii) (A2A-noScale) model: skipping the use of sigmoid scaling.
- (ix) (A2A-noSubtProp) model: attention propagation using only visual input.
- (x) (A2A-noVisProp) model: attention propagation using only subtitle input.
- (xi) (A2A) model: using all components described in our method.

In Table 2, we find that the full A2A model achieves the best performance on both the validation and test sets among all the other variants. It implies that all of our model components are essential. For instance, A2A-noJE has a performance gap in comparison with A2A model. A reasonable explanation is that the normalized affine transform \mathcal{J} plays a key role in jointly embedding different modalities to the same semantic space, and enables the subsequent model components to explore correlations among inputs. Further, A2A-noQAattn also has a large gap in performance which indicates the QA attention mechanism is crucial to the localization of the most relevant content in video context.

We further investigate the outcome of source ablation. While A2A-noVis is the second best among all variants, we notice that it is better than A2A-noVisProp and A2A-noSubtProp. A possible reason is that the propagated attention of single modality of either the visual or subtitle input may impinge on the QA attention due to the connection between the two is left out.

Methods	Validation	Test	Methods	Validation	Test
A2A (ours)	41.66 %	41.97 %	RWMN [23]	38.67 %	36.25 %
A2A-noVisProp	41.05 %	41.65 %	DEMNI [16]	44.70 %	29.97 %
LMN [31]	42.50 %	39.03 %	SSCB [30]	21.90 %	–
SC-MemN2N [4]	–	38.16 %	MemN2N [30]	34.20 %	–

Table 3: Performance comparison among the proposed A2A method, others from MovieQA leader board, and the two baselines, SSCB and MemN2N, in the original MovieQA paper [30]. “–” means the method is not evaluated.

Variants	Validation
A2A-GloVe	41.12 ± 0.594 %
A2A-U+SAttn	41.06 ± 0.318 %
A2A-SoftmaxAttn	28.32 ± 0.087 %
A2A-FeatRelu	31.92 ± 0.410 %
A2A	41.66 ± 0.406 %

Table 4: Comparison with 4 implementation variants of A2A.

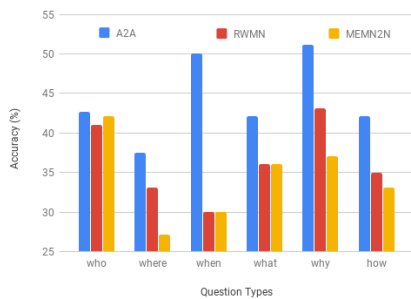


Fig. 6: Performance comparison on different question types.

Last but not least, we observe from the performance of A2A-noL2norm that L_2 -normalization on features is critical to training our method for solving the MQA task. Without using L_2 -normalization, the training would eventually fail. The phenomenon is caused by gradient vanishing due to small output values. On the other hand, A2A-noDropout and A2A-noScale are slightly inferior to A2A model, which suggests dropout operation and sigmoid scaling are needed.

4.2 Leader board Comparison

In Table 3, we compare the A2A models with those from the MovieQA leader board² and the baselines used in Tapaswi *et al.* [30]. Our method achieves the best performance on the test set among all others. Compared with LMN [31], A2A-noVisProp improves by 2.62% in accuracy and A2A improves by 2.94%. The results also indicate that our visual attention propagation is effective.

4.3 Model Selection

To search the best combination for MovieQA benchmark, we experiment on different parameter settings and implementation variants. Due to the limited

² <http://movieqa.cs.toronto.edu/leaderboard/>

space allowed, we omit the detailed experimental results and discuss our findings. The model selection is examined by varying different combinations of the number of layers in the normalized affine transform, visual propagated attention layers, and subtitle propagated attention layers. First, we observe that increasing the number of normalized affine transform layers in joint embedding does not yield performance gain, and indeed one layer is sufficient to achieve relatively good performance. Second, increasing the numbers of subtitle and visual propagated attention layers is beneficial to the performance. However, it becomes worse when the number of layers is over three/five for visual/subtitle propagated attention. The reason might be due to model overfitting while increasing the parameters. We next compare four implementation variants of A2A as reported in Table 4.

- (i) (A2A-GloVe) model: using GloVe [24] for word embedding. Each unknown token is mapped to the average vector of whole GloVe embedding.
- (ii) (A2A-U+SAttn) model: constructing V_{attn} and S_{attn} with the summation of visual α^V attention, and subtitle α^S attention.
- (iii) (A2A-SoftmaxAttn) model: replacing relu with softmax to yield attention.
- (iv) (A2A-FeatRelu) model: adding relu function to every fully-connected layer.

From Table 4, we find A2A-GloVe is slightly worse than the full A2A model. It suggests that the advantage of using our new word embedding is noticeable but subtle. Presumably, the main reason might be that only a small proportion of movies (e.g. fiction movies which usually have characters with unusual names.) incurs the OOV problem. For A2A-U+SAttn, it performs a bit worse than A2A. It is because the summation of both attention cues may cause the final representation M_q to get some irrelevant information from visual and subtitle representations. Moreover, A2A-softmaxAttn and A2A-FeatRelu models degrade drastically, due to the use of softmax to account for attention fusion.

4.4 Question Types Comparison

In MovieQA benchmark [30], questions can be classified into six different question types: *Who*, *Where*, *When*, *What*, *Why*, and *How*. Usually, answering *Where*, *When*, *What* questions (e.g., “Where does Bruce go after revealing himself to Vicki as batman?, When does Forrest discover that he can run really fast for the first time?, What does Gandalf retrieve from Saruman?”) requires localizing relevant information in context, and answering *Why* and *How* questions (e.g., “Why does Gollum ask Frodo to leave Sam behind?, How does Bruce survive the Joker’s bullets?”) requires abridging the context and relational reasoning. As for *Who* questions (e.g., “Who attacks Zachry, Adam, and Zachry’s nephew?”), it needs name entity matching.

Figure 6 compares the accuracy of A2A, RWMN [23], and MeMN2N [30] on different question types. We find that A2A outperforms all the other methods on every question type. For *Who* questions, A2A works slightly better than the others. The performance improvement of A2A on every question type except *Who* is over 4.5%. It indicates that our method is quite capable of summarizing

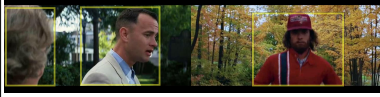
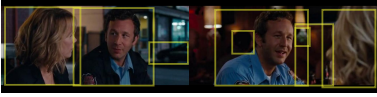
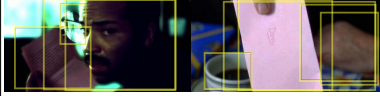
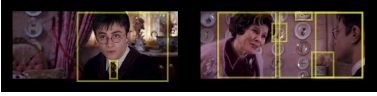
	
<p>q: Why does Forrest undertake a three-year marathon?</p> <p>a₁: He wants to find Jenny.</p> <p>a₂: <i>Upset that Jenny left he decides to go for a run one morning and just keeps running.</i></p> <p>a₃: He wants to get back in shape.</p> <p>a₄: He wants to raise awareness for cancer.</p> <p>a₅: He wants to travel the country but doesn't want to drive.</p>	<p>q: How does Annie react when Rhodes suggests that she opens up a new bakery?</p> <p>a₁: She asks him for help.</p> <p>a₂: <i>She refuses.</i></p> <p>a₃: She slaps him.</p> <p>a₄: <i>She ignores him.</i></p> <p>a₅: She agrees.</p>
	
<p>q: What does Don find, when he returns home from the trip?</p> <p>a₁: Another anonymous pink letter.</p> <p>a₂: A letter from Lolita.</p> <p>a₃: Sherry waiting for him.</p> <p>a₄: <i>A pink letter from Sherry.</i></p> <p>a₅: A young man waiting to see him.</p>	<p>q: What kind of relationship do Harry and Umbridge have?</p> <p>a₁: A friendly one.</p> <p>a₂: <i>A mentor-student type of relationship.</i></p> <p>a₃: A mother-son type of relationship.</p> <p>a₄: <i>None; they do not get along at all.</i></p> <p>a₅: A romantic one.</p>

Table 5: The examples of our results. The answer with green color is the ground truth, and one with Italic style is the predicted answer.

and reasoning in *How* and *Why* questions. Furthermore, our method can extract the key part of context in *Where*, *When*, and *What* questions.

5 Conclusions

We have shown that the proposed Attention to Attention (A2A) reasoning effectively addresses the problem of movie question answering. With the A2A reasoning mechanism, our method distills attention cues to aggregate and to associate the different representation modalities for answering questions. Besides, we establish a flexible n-grams word embedding for tackling the out-of-vocabulary issue. The experimental results show the state-of-the-art performance on the “Video+Subtitles” entry of MovieQA benchmark dataset.

Acknowledgement. This work was supported in part by MOST Grants 107-2634-F-001-002 and 106-2221-E-007-080-MY3 in Taiwan.

References

1. Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C.L., Parikh, D., Batra, D.: VQA: visual question answering - www.visualqa.org. *International Journal of Computer Vision* **123**(1), 4–31 (2017)
2. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: *CVPR* (2018)
3. Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: *ICLR* (2017)
4. Azab, M., Wang, M., Smith, M., Kojima, N., Deng, J., Mihalcea, R.: Speaker naming in movies. In: *NAACL-HLT*. pp. 2206–2216 (2018)
5. Bello, I., Zoph, B., Vasudevan, V., Le, Q.V.: Neural optimizer search with reinforcement learning. In: *ICML*. pp. 459–468 (2017)
6. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multi-modal compact bilinear pooling for visual question answering and visual grounding. In: *EMNLP*. pp. 457–468 (2016)
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: *AISTATS*. pp. 249–256 (2010)
8. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In: *CVPR* (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385* (2015)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Computation* **9**(8), 1735–1780 (1997)
11. Hu, H., Chao, W.L., Sha, F.: Learning answer embeddings for visual question answering. In: *CVPR* (2018)
12. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: *CVPR*. pp. 3296–3297 (2017)
13. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: TGIF-QA: toward spatio-temporal reasoning in visual question answering. In: *CVPR* (2017)
14. Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Zitnick, C.L., Girshick, R.B.: Inferring and executing programs for visual reasoning. In: *ICCV*. pp. 3008–3017 (2017)
15. Kazemi, V., Elqursh, A.: Show, ask, attend, and answer: A strong baseline for visual question answering. *CoRR* **abs/1704.03162** (2017)
16. Kim, K., Heo, M., Choi, S., Zhang, B.: Deepstory: Video story QA by deep embedded memory networks. In: *IJCAI*. pp. 2016–2022 (2017)
17. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: *ECCV*. pp. 740–755 (2014)
18. Liu, F., Perez, J.: Gated end-to-end memory networks. In: *EACL* (2017)
19. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: *NIPS*. pp. 289–297 (2016)
20. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *CoRR* (2013)
21. Miller, A.H., Fisch, A., Dodge, J., Karimi, A., Bordes, A., Weston, J.: Key-value memory networks for directly reading documents. In: *EMNLP*. pp. 1400–1409 (2016)

22. Mun, J., Seo, P.H., Jung, I., Han, B.: Marioqa: Answering questions by watching gameplay videos. In: ICCV. pp. 2886–2894 (2017)
23. Na, S., Lee, S., Kim, J., Kim, G.: A read-write memory network for movie story understanding. In: ICCV (2017)
24. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. pp. 1532–1543 (2014)
25. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100, 000+ questions for machine comprehension of text. In: EMNLP. pp. 2383–2392 (2016)
26. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017)
27. Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., Courville, A., Schiele, B.: Movie description. *International Journal of Computer Vision* (2017)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
29. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks. In: NIPS (2015)
30. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: CVPR (2016)
31. Wang, B., Xu, Y., Han, Y., Hong, R.: Movie question answering: Remembering the textual cues for layered visual contents. In: AAAI (2018)
32. Weston, J., Bordes, A., Chopra, S., Mikolov, T.: Towards ai-complete question answering: A set of prerequisite toy tasks. CoRR **abs/1502.05698** (2015)
33. Weston, J., Chopra, S., Bordes, A.: Memory networks. CoRR **abs/1410.3916** (2014)
34. Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A.R., van den Hengel, A.: Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding* **163**, 21–40 (2017)