A Better Way to Attend: Attention with Trees for Video Question Answering

Hongyang Xue, Wenqing Chu, Zhou Zhao, and Deng Cai*, Member, IEEE

Abstract—We propose a new attention model for video question answering. The main idea of the attention models is to locate on the most informative parts of the visual data. The attention mechanisms are quite popular these days. However, most existing visual attention mechanisms regard the question as a whole. They ignore the word-level semantics where each word can have different attentions and some words need no attention. Neither do they consider the semantic structure of the sentences. Although the Extended Soft Attention (E-SA) model for video question answering leverages the word-level attention, it performs poorly on long question sentences. In this paper, we propose the heterogeneous tree-structured memory network (HTreeMN) for video question answering. Our proposed approach is based upon the syntax parse trees of the question sentences. The HTreeMN treats the words differently where the visual words are processed with an attention module and the verbal ones not. It also utilizes the semantic structure of the sentences by combining the neighbors based on the recursive structure of the parse trees. The understandings of the words and the videos are propagated and merged from leaves to the root. Furthermore, we build a hierarchical attention mechanism to distill the attended features. We evaluate our approach on two datasets. The experimental results show the superiority of our HTreeMN model over the other attention models especially on complex questions. Our code is available on github.

Index Terms—Video question answering, attention model, scene understanding.

I. INTRODUCTION

Visual question answering is an important task towards scene understanding [1], which is one of the ultimate goals of computer vision. The majority of the existing works focus only on static images [35], [33], [19] that are far from the real-world applications. The more ubiquitous task of video question answering, which is closer to the dynamic real-world scenes, has been recently addressed in [41]. In [41], four models are extended from the previous Image-QA and video captioning tasks [23], [11], [36], [30] which are named as E-MN, E-VQA, E-SA, and E-SS. Among them the E-SA model extended from [36], which is an attention-based model, shows the best performance.

The attention mechanisms have become popular in many tasks recently, particularly in visual captioning [34], [39], [38], [10] and visual question answering [35], [33], [19], [23], [32].

Despite the success of the current attention mechanisms, we found that the existing attention mechanisms have several drawbacks:

- First, most current attention approaches usually regard the question as a whole. The question is encoded into an embedding vector. During this process, much information might be lost. When facing complex and long questions, the performance of these models drops significantly.
- Although the E-SA [41] model leverages the word-level attention, it only encodes the questions employing LSTM

In image question answering (Visual-QA), the basic idea of the attention mechanism is to dynamically focus on different fine-grained regions of the images based on the questions [35]. Since the proposal of the SAN [35] model which showed the effectiveness of the attention mechanism, the attention-based models have become major parts of the Visual-QA approaches [33], [19], [23], [32]. In video question answering, the E-SA model [41] further extends the spatial attention to the temporal dimension for videos and leverage the word-level attention mechanisms. All these question-guided attentions have been proved to be quite effective.

![Fig. 1: Overview of our HTreeMN model. The tree is the parse tree of the question What is the woman cooking in the pot?. The leaves are the words. S is the root. The nodes with double circles like pot indicates the need of attentions. The intermediate nodes such as NP and VP are the intermediate representations of the sentence. We compute joint question-video embeddings from bottom to the top. The answer meat is generated in the end.](image-url)
in a linear chain manner. The syntactic properties of the natural language are discarded, as a result of which, the performance significantly deteriorates. (See our experiments.)

- Some words may relate to concrete ideas, such as objects and actions, which have strong meanings in the attention procedure. We name these words visual words. However, some words are verbal, such as the linking verbs is, are, and the conjunctions. These words are hard to correlate with regions or clips of the visual data. They are non-visual and belong to the underlying language models.

Current approaches do not distinguish the two kinds.

To remedy these drawbacks, we propose the Heterogeneous Tree-Structured Memory Network Model (HTreeMN). The HTreeMN model is built upon the syntax parse trees of the sentences (An example is shown in Figure 1. Similar ideas have been employed for NLP tasks such as sentiment classification). The leaves are the words in the question sentence. We divide these words into two types: the visual and the verbal words. The visual words are combined with the visual feature through a temporal attention module while the verbal words remain their semantic meanings. These two types of words are then put through different transformations before passing into the trees. Then the joint word-video representations are merged and passed bottom-up to the root. Finally the answer is decoded from the root’s state.

The HTreeMN model not only utilizes the word-level attentions, distinguishing the visual and verbal words, but also adopts the syntactic tree-structured property of the natural language which combines words into phrases and then into sentences. The understandings of the words and videos are propagated bottom-up in the manner how a sentence is understood.

There are also many intermediate nodes in the parse tree such as NP, VP and SQ. These intermediate nodes contain the intermediate representations of the joint word-video embeddings. To distill the encoded features, we also propose a hierarchical framework where the intermediate nodes of the parse tree are also equipped with attention modules. Whether an intermediate node needs attention depends on its children. Details will be explained in the algorithm section.

We summarize our contribution as follows:

- We propose a tree-structured memory network based on the parse tree of the question sentence. The parse trees are generated following. The tree models better utilize the syntactic property of the natural language compared with the traditional linear chain models. Our model can perform very well on complex questions while the previous attention models cannot.

- The tree-structured memory network enables word-level attention. We distinguish the visual words and the verbal words so that the attention computation is more reasonable and meaningful.

- We propose a hierarchical attention framework to distill the encoded features. We propose an algorithm that decides whether an intermediate node needs attention. The final encoded features are more related to the questions and answers.

In the end, we evaluate our approach on two video question answering datasets. We compare our model with the E-SA model and show a significant improvement. We also leave out some of our model’s features. The results show the effectiveness of all the proposed components.

The following sections are organized as follows: We first briefly review the work related to ours. Then we present the details of our proposed models. Finally, we describe the design of the experiments and analyze the results.

II. RELATED WORK

A. Video-QA

The video question answering (Video-QA) is a recently proposed task. In, they propose a dataset with the help of question generation approaches. They also extend four models in Image-QA to Video-QA. As reported in, the attention-based model (E-SA) achieves the best result. The E-SA model is a simple word-level attention model where each word attends to the video frames and all the words’ attentions are accumulated with an LSTM.

Prior to, there have also been some works similar to Video-QA. The Movie-QA task combines movies with plots, subtitles, and scripts to generate answers. The “fill-in-the-blank” task requires a blank of a sentence to be filled with a word given the videos.

B. Attentions in Visual-QA

The attention models have a long history. Researchers first apply it for image recognition. Recently the attention mechanisms make their way into NLP and vision tasks with the recurrent neural networks and the more general memory networks. Since our paper deals with video question answering tasks, we pay our attention mainly to the models in visual question answering.

The visual question answering tasks (on images) have been intensively studied. Since the proposal of the SAN model, a lot of attention-based models for Visual-QA have been developed. In, a convolutional attention network is proposed for visual question answering. A question-image co-attention mechanism is proposed where the phrase-level attention is addressed. In, an attention model based on edge boxes is proposed. Different from the SAN which separates the image uniformly, relies on the edge boxes to generate region proposals and then compute the attentions on the proposals. In, a dynamic memory network is proposed to allow the usage of supporting facts in question answering. In, a memory network with attention mechanism is used for Visual-QA. More recently, a focused dynamic attention model is proposed and research on the gap between human attention and deep networks are also studied.

Those previous works seldom consider word-level attentions or only accumulate the word-level attention sequentially. They do not distinguish words that need or do not need attention. They usually encode the question with a linear chain RNN or memory network. Although the LSTM and memory networks can keep the information of a long sequence, their encoding
of the sentences loses the syntactic property of the natural language which can be modeled with parse trees.

C. Tree Networks

There have been some works on tree-structured LSTMs for NLP tasks. In [26], a Tree-LSTM is proposed for modeling semantic relatedness and sentiment classification for natural language sentences. Sequential and tree LSTMs are later integrated for natural language inference in [51, 17] also shows that tree structures are usually necessary for deep learning NLP tasks. Later a bidirectional Tree-LSTM is proposed in [28]. In [37], a tree-structured recursive neural network is applied for visual-textual sentiment analysis. The tree structures have also been utilized for machine translation in [8] and for object localization in [14]. Our model differs from them in the following aspects: We introduce a tree-structured memory network which contains the heterogeneous nodes. We also build a hierarchical attention mechanism on the network. The information propagated on our HTreeMN is the combined visual and textual features. And we utilize the final encoding to answer the questions on videos.

III. HETEROGENEOUS TREE-STRUCTURED MEMORY NETWORK

In video question answering, we need to encode the videos and the questions. The features from different domains should be combined to decode the answer. In this paper, we address the open-ended question answering task following [41].

Given a video \( v = \{v_1, \ldots, v_T\} \) and a question \( q = \{q_1, \ldots, q_N\} \), where \( v_i \in \mathbb{R}^m \) are the \( m \)-dimensional visual feature of the frames and \( q_i \in \mathbb{R}^n \) are the \( n \)-dimensional embeddings of the tokens, our purpose is to produce the answer from the answer list \( A \). The answer selection scheme is very similar to a classification problem. While most previous works encode the questions with linear chain models, in our work, we model it with a tree-structured model. To avoid confusion about notations, we will use \( v \) to denote the matrix in \( \mathbb{R}^{T \times m} \) where the \( i \)th row of \( v \) is the feature vector of the \( i \)th frame \( v_i \).

Before diving into the details of our models, we first review the E-SA model [41] and the TreeLSTMs [26], since they are closely related to our proposed models.

A. Previous Works

In this subsection, we briefly review the E-SA [41] and the TreeLSTM [26] models.

The E-SA [41] model is an attention-based model which consists of two-layer LSTMs (see Figure 3). The question sentences are padded to a fixed length and then processed by the first layer LSTM. After that, the outputs are processed by the second layer LSTM augmented with attention mechanisms. In the end, the outputs of both LSTMs are combined to produce the result.

TreeLSTMs (Figure 4) are improvements over traditional LSTMs since it enables the underlying LSTM structure as a complex tree rather than a linear chain. The difference between an ordinary LSTM and a TreeLSTM is that in a TreeLSTM, the inputs to a cell comes from several child cells. As a result, while computing the current cell value, the cell values of children should be summed after multiplying their respective forget gate.

The E-SA [41] model only employs an ordinary LSTM of the linear chain structure while the TreeLSTM [26] enables an underlying structure of a tree. The TreeLSTM can better capture the complex syntactic properties that would naturally combine words to phrases. However, the E-SA model [41] is augmented with an attention mechanism while the TreeLSTM [26] has no attention mechanism. In this paper, an attention-based variety of TreeLSTM is proposed.
B. Attention with Tree-Structured Memory Network

From this section, we describe our proposed models.

The parse trees of different questions can have different topological structures. We build the parse trees for sentences using the StanfordParser tool [16]. Given the question $q$, we can build a parse tree for it (See Figure 2 for example). Its corresponding tree-structured memory network is shown in Figure 5. The nodes of the memory network can be categorized into three types: the leaves, the intermediate nodes, and the root. Denote the $n$-dimensional state vector of node $c_i$ as $h_i \in \mathbb{R}^n$, $h_i$ is computed from all its children which we will denote as $C(i)$. We now describe how we compute the state for every node in a bottom-up manner.

1) Attention Module: The attention module is similar to the E-SA model in [41]. The video frame features are first processed by a bi-directional LSTM. The processed features correspond to sentence intermediate representations such as the component of $h$ where $W$ are parameters which transform the visual and textual features into the joint space. $W_p \in \mathbb{R}^{T \times n}$ is a linear transformation.

It is the temporal attention over the video frames.

2) Leaf Node: The leaf nodes correspond to the words in the question sentence. The state of a leaf node does not come from its children (it has no children) but is computed from the word embedding and the attention over video frames. Initialize the state $h_j$ of a leaf node $c_j$ to be $h_j = q_j$, its new state is computed by:

$$h_{A,j} = \text{tanh}(W_Qh_i + W_Vv_j + b_V)$$
$$p = \text{softmax}(W_Ph_i)$$
$$\text{att}_i = \text{Attention}(h_i, v)$$
$$= \sum_{k=1}^{T} p_kv_k$$

where $h_{A,j}$ is the $j$th component of $h_A$, $p_k$ is the $k$th component of $p$, $W_Q \in \mathbb{R}^{n \times n}$, $W_V \in \mathbb{R}^{n \times m}$ and $b_V \in \mathbb{R}^n$ are parameters which transform the visual and textual features into the joint space. $W_p \in \mathbb{R}^{T \times n}$ is a linear transformation.

3) Intermediate Node: For the intermediate nodes which correspond to sentence intermediate representations such as $NP$, $VP$ and $NN$, the state $h_i$ of an intermediate node $c_i$ is computed from its children $C(i)$:

$$h_i = \sum_{j \in C(i)} W_B o_j + b_B$$

where $W_B \in \mathbb{R}^{n \times n}$, $b_B \in \mathbb{R}^n$ is the linear layer shared by all the intermediate nodes. Its output is $o_j = \text{tanh}(h_j)$.

The computation is performed from bottom to the top in a recursive manner. The information is propagated from the leaves to the parents and finally to the root. At the root $c_0$, its state $h_0$ can be computed following the computation of the intermediate nodes. Then a linear layer is placed followed by a softmax layer for answer classification:

$$y = \text{softmax}(Wh_0 + b)$$

where $W \in \mathbb{R}^{z \times n}$, $n \in \mathbb{R}^z$ and $z$ is the number of answers.

C. Heterogeneous Nodes

In the previous subsection, we describe how the tree-structured memory network is built and how computation is performed on the tree. All the leaf nodes which correspond to the words in the question sentence are equipped with an attention module. However, as discussed in the introduction, the visual words need attention while the verbal words do not need. This requires us to treat the leaf nodes differently.

The heterogeneous tree memory network is shown in Figure 6. The gray nodes are the leaf nodes which need attention computation while the green ones are the leaf nodes of the verbal type. We treat the visual and the verbal nodes differently.

1) Visual Node: For a visual node $c_i$, we need to compute the attention over the videos frames

$$\text{att}_i = \text{Attention}(q_i, v)$$

its state is computed

$$h_i = W_A \text{att}_i + b_A + q_i$$

Fig. 5: The tree-structured memory network (TreeMN) corresponds to the question in Figure 2. Double circle nodes are leaf nodes which are the words in the question. Each leaf node needs to compute the attention over the video frames. The encoded features are propagated from leaves to the root.
Hierarchical attention mechanisms, such as the stacked attention networks (SAN) [35], are designed to enable multi-step reasoning on the visual data. The multi-step reasoning gradually narrows down the focus on the related regions [35]. The hierarchical structure can result in better-attended features since the concentration will be more focused.

Since the parse tree of the sentence provides us a natural hierarchical structure, we also build a hierarchical attention mechanism in our tree-structured memory network. The nodes are the word-level and phrase-level hidden representations of the sentence. To enable multi-step reasoning, we also equip the intermediate nodes with the attention modules.

Like the leaf nodes, the intermediate nodes are also categorized into two types: visual and verbal nodes. Whether an intermediate node needs attention is defined recursively on the tree. An intermediate node is a verbal node if and only if all of its children are verbal nodes. Otherwise, it needs attention computation.

However, the intermediate nodes and leaf nodes are processed differently before they are passing into the parent node. The state of an intermediate node $c_k$ is computed from its children as

$$h_k = \sum_{u \in C_1(k)} W_{B_1} o_u + \sum_{w \in C_2(k)} W_{B_2} o_w + b_B$$  \hspace{1cm} (8)$$

where $C_1(k)$ and $C_2(k)$ are the children set of visual and verbal nodes respectively. $W_{B_1} \in \mathbb{R}^{n \times n}, W_{B_2} \in \mathbb{R}^{n \times n}, b_B \in \mathbb{R}^n$ is the linear layer.

E. Full Model

Our Heterogeneous Tree-Structured Memory Network (HTreeMN) consists of the following features:

1) Encoding question and visual features with the parse tree structure.
2) Four types of heterogeneous nodes: the visual leaf and intermediate nodes and the verbal leaf and intermediate nodes.
3) Hierarchical attention over the tree-structured networks.

The full model is shown in Figure 7. Green nodes are the verbal nodes that do not need attention computations. Red nodes indicate the visual nodes. Each self loop on a node indicates an attention computation.

IV. EXPERIMENTS

In this section, we describe our datasets and conduct all the experiments on them.

A. Datasets

In [41] the authors propose a dataset which is not totally available till now. We perform experiments on two video-QA datasets: the YouTube-QA and the TGIF-QA datasets. The two datasets are constructed from the annotated video clip data [39] and the annotated GIF data [18] with natural language descriptions. The two datasets are harvested following the method in [41]. We focus on questions starting with What, Who, Whose, Where, When and How many. These types of
questions can correspond to the questions on object, location, number and time. We then split the two datasets into the training, validation and testing sets. The split of the datasets is shown in Table I and Table II. The size of the training set is almost equal to the size of both the validation and the testing set. Like [41], we choose the top $K$ most frequent answers as possible candidates. The answer space $K$ of the two datasets are 500 and 1000 respectively.

Both our datasets and codes are now publicly available [6].

| Data Split | Question Type | All | Object | Location | Number | Time |
|------------|---------------|-----|--------|----------|--------|------|
| Train      | 24415         | 23452 | 99     | 758      | 106    |
| Validation | 7684          | 7420  | 45     | 169      | 50     |
| Test       | 16258         | 15528 | 88     | 573      | 69     |

TABLE I: The split of the YouTube-QA dataset.

| Data Split | Question Type | All | Object | Location | Number | Time |
|------------|---------------|-----|--------|----------|--------|------|
| Train      | 79967         | 74918 | 680    | 3494     | 875    |
| Validation | 19526         | 18379 | 170    | 782      | 195    |
| Test       | 63101         | 58962 | 520    | 2828     | 791    |

TABLE II: The split of the TGIF-QA dataset.

B. Data Preparation

In this section, we describe how we preprocess the datasets.

1) Visual Feature: Since the video clips in the datasets contain variable lengths, we sample them to have the same number of frames. We sample 60 frames for each video in the YouTube-QA dataset and 30 frames for the TGIF-QA dataset on account of the fact that videos in TGIF-QA dataset are shorter. To extract the visual feature of each frame, we first resize the frames to be of size $224 \times 224$ and employ the VGGNet [24] pre-trained on the ImageNet dataset for feature extraction. We take the 4096-dimensional feature vector of the first fully-connected layer as the visual feature.

2) Word Embedding: The question answering datasets contain a large amount of vocabulary. We apply the pre-trained GloVe model [22] to embed each word into a 300-dimensional vector. We do not take the other approach which joint learns the word embedding and the question answering task. To classify the words into visual and verbal, we refer to the result of [3] which contains the concreteness ratings of 40,000 English words and use 0.5 as the threshold.

C. Evaluation Metrics

We follow the evaluation metrics in [41] to evaluate the experimental results. We first use the accuracy to measure the results. Furthermore, we employ the WUPS score to measure the answer quality since many words have similar meanings. The score is defined as follows

$$WUPS@t(a, o) = \begin{cases} WUP(a, o) & WUP(a, o) \geq t \\ 0.1 \times WUP(a, o) & \text{otherwise} \end{cases}$$

where $WUP$ score [31] is a word-level similarity measure based on WordNet [9]. We use the WUPS score under two thresholds: the WUPS@0.0 and WUPS@0.9.
D. Comparison Methods

We compare our model with the E-SA model \cite{41} and also do ablation study on our proposed mechanisms.

1) Simple: We design an non-word-level attention model which is simpler than the E-SA \cite{41} model. This model encodes the questions first. Then the encoding of the question is used to attend the videos. The hidden representation of the features is of dimensionality 1024.

2) E-SA: The E-SA model (Figure 3) is the same as in \cite{41}. The hidden representation of the features is of dimensionality 1024. The E-SA model \cite{41} is extended from the model \cite{36} for video captioning. It computes the attention for each word and then accumulates the encoding with an LSTM.

3) E-SS: The E-SS model \cite{41} is a sequence-to-sequence model. They \cite{41} employed E-SS to first, encode a video; then, encode a question; finally, decode an answer. This model mimics how humans first watch a video; then, listen to a question; finally, answer the question \cite{41}. The hidden representation of the features is of dimensionality 1024.

4) TreeMN: The TreeMN model does not contain the heterogeneous nodes. The implementation of this model only follows the description in section 3.1. All the leaf nodes are computed with the attention modules. The hidden representation of features is of size 1024.

5) HTreeMN-non-hierarchical: This model follows the description of section 3.2. It adds the heterogeneous nodes to the TreeMN model. Compared with the full HTreeMN model, this model only computes attention on part of the leaf nodes. The intermediate nodes are processed without attention computations. We set the hidden representation size to be 1024. We will name this model as HTreeMN-noh for short.

6) HTreeMN: This is the full model. It adds the hierarchical attention mechanism to the previous model. The HTreeMN model contains 4 types of heterogeneous nodes. The visual and the verbal leaf nodes and the visual and the verbal intermediate nodes.

E. Implementation Details

1) Model Details: The visual features are encoded by a bi-directional LSTMs from dimensionality 4096 to 1024. The hidden representation of the networks is set to size 1024. In the HTreeMN model, all the linear layer of the tree, including \(W_{A_1}, W_{A_2}, W_{B_1}\), and \(W_{B_2}\), are of size 1024 \(\times\) 1024. On all the leaf nodes, the raw word embeddings are transformed by linear layer to dimensionality 1024.

2) Optimization: We train all our models using stochastic gradient descent (SGD) with the Adam strategy \cite{13}. At the \(t\)-th step, the parameter \(\Theta\) is updated by:

\[
\Theta_t = \Theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{(\sqrt{\hat{v}_t} + \epsilon)}
\]

where \(\hat{m}_t\) is the bias-corrected first moment estimate, \(\hat{v}_t\) is the biased second raw moment estimate. \(m_t\) and \(v_t\) are the biased first and second moment estimates. The initial learning rate is set to 0.0001. The exponential decay rates for the first and the second moment estimates are set to 0.1 and 0.001 respectively.

We use the mini-batch training strategy. The batch size is set to 64 which is the largest batch that can fit into our NVIDIA Titan X GPU’s memory. A gradient clipping scheme is utilized to clip gradient norms within 10. An early stopping strategy is applied to stop training when the validation accuracy no longer improves for 10 epochs. We select the final model according to the best validation accuracy.

F. Results

1) Overall: We list the overall results of all the models in Table III and IV. As we can see from the tables, Our full HTreeMN model performs the best on both the two datasets. Our tree-structured attention mechanism outperforms the simple attention, the E-SS \cite{41} and the E-SA \cite{41} models. Notice that under the WUPS@0.9 measure, our HTreeMN model achieves very high scores compared with all the other approaches, which suggests that the HTreeMN model produces answers of high quality.

In addition, we can notice that the E-SA and the E-SS methods have incremental improvements over the simplest attention model.

2) Detailed Analysis: We further report the results of the approaches on all types of questions based on the three evaluation criteria. These results are shown in table V to X. Table V to X are the accuracy, WUPS@0.0 and WUPS@0.9 scores of each type of questions on the YouTube-QA dataset. Table VII to X are the accuracy, WUPS@0.0 and WUPS@0.9 scores of each type of questions on the TGIF-QA dataset. We can see from the results that all the methods perform better on the questions about number and time than object and location. The attention mechanism with a tree improves the results significantly on questions about Number and Time.

The data amount of different types of questions is imbalanced as shown in Table II and III. Since the data on the questions about object occupies the majority of the dataset, the remaining types of questions only have few samples for training. The result shows that our model can be trained with a small number of data and achieves high performance on these types of questions.

Questions about locations are the most difficult. Our full HTreeMN model is not very good at such kind of questions while the TreeMN and the TreeMN-noh models achieve the best results on the TGIF-QA and the YouTube-QA dataset respectively. The TreeMN-noh method also performs the best on the number questions on the TGIF-QA dataset.

All the models achieve better performance on the TGIF-QA dataset. This is because the TGIF-QA dataset is much larger than the YouTube-QA dataset, which enables the models to be sufficiently trained. And it also can be seen that on the smaller dataset of YouTube-QA, our proposed approaches have more improvement over the E-SA, the E-SS, and the simple baseline models, which demonstrates that our approaches can better utilize the data since semantic structures are well considered.

In all the cases, our tree-structured networks achieve the best performance. This fact shows that the tree-structured attention mechanism that both enables the word-level attention and utilizes the syntactic property of the natural language sentences...
can improve the performance of video question answering. The HTreeMN and HTreeMN-noh have little performance discrepancy, which indicates that HTreeMN-noh is the best choice from the aspects of both accuracy and efficiency. Besides, TreeMN is the most efficient among the three models and its accuracy also significantly surpasses previous approaches.

3) Question Complexity: We further analyze the results of the models on the questions of different complexity. We measure the complexity of a question by its length. Thus we divide the questions into 5 sets according to their lengths. These sets are $\{0, 5\}$, $\{5, 10\}$, $\{10, 15\}$, $\{15, 20\}$ and $\{20, \infty\}$, where $(x, y)$ includes the questions whose length are greater than $x$ but no more than $y$. We report the accuracies of the questions of different lengths in Figure 8.

From the histograms in Figure 8, we observe several interesting points:

- The E-SA and the E-SS [41] models perform poorly on long questions. On long questions, they perform even worse than the Simple model which does not leverage word-level attentions while on short questions they perform better than the Simple model.

- Our tree-structured attention models perform quite well on long questions as well as short questions.

Although the E-SA model [41] leverages the word-level attention mechanism, it only considers the sequential structure of the question sentence. For a long question, the sequential encoding may be inaccurate. This can account for the reason why E-SA [41] performs better than the Simple model on short questions but becomes worse on long questions. The reason why E-SS [41] model performs worse as the question length increases is also that the E-SS [41] model encodes the semantic structures of the natural language sentences.

4) Examples: We display several examples of long questions where our model performs well while the E-SA model [41] fails. The examples are shown in Figure 9. In these cases, the structures of the questions are complex and a simple sentence embedding model cannot well encode the semantics. We also display several failure cases in Figure 10. As can be seen most errors are caused by recognition errors. For example, the baby lions are recognized as dogs; the mouse is recognized as rabbit; and lemon is recognized as orange. One of the causes
Question Length

Accuracy

0 − 5 5 − 10 10 − 15 15 − 20 > 20

(a) YouTube-QA: Accuracy on different lengths of questions.

(b) TGIF-QA: Accuracy on different lengths of questions.

Fig. 8: The accuracy on questions of different complexity. We divide the lengths into 5 sets 0 − 5, 5 − 10, 10 − 15, 15 − 20 and those larger than 20. The labels are ranges where 0 − 5 indicates length larger than 0 and smaller than or equal to 5.

Fig. 9: We display several examples of long questions where the E-SA model errs and our model succeeds.

for the recognition error may be that some objects are rare in the dataset. In all, the experiments reveal that the sequential word-level attention cannot well encode the sentence. Our proposed
attention mechanism with the tree-structured memory network can better leverage the syntactic property of the sentences and are more accurate in encoding long sentences. The experiments also show the effectiveness of the heterogeneous nodes which selectively attend on videos, and the effectiveness of the hierarchical attention mechanism on the tree.

5) Qualitative Examples: In this subsection, we qualitatively show the examples of attentions for verbal and visual words in Figure 11. The TreeMN computes the attention distributions for all the words while the TreeMN-noh approach only computes the attention distributions for visual words.

As can be seen in Figure 11, the attention distributions computed on the verbal words are almost meaningless. Besides, the computed attentions for verbal words introduce noises into the whole attended features. For example, the attention for the word *are* is focused at the end of the video while this fragment contributes little to answer the question. And so is the attention for the words *the* and *by*. The HTreeMN-noh approach avoids the redundant computation for the attentions on the verbal words. The attentions are more focused on the important parts. In this example (Figure 11), the attentions are focused on the fragment where the chicken attacked the rabbits.

Notice that for a few amount of data, for a single visual word, there can exist several frames of similar contents while the attention values on these frames are quite different. The reason is that the attention distribution tends to have peaks, since the model is trained not merely on single words but on the accumulated attentions of the whole sentences. If we are training a model which captures the visual-textual correspondence for a single word, the distributions should be equal on the related frames. However, the attention for a single word here actually is not the attention for the isolated word, and we are actually training models where the attention of a single word is affected by the other words in the question sentence. The frames informative to answer the question are usually very short. After the model is trained, the attention distributions become concentrated for all the words.

V. CONCLUSION

We propose the attention mechanism with the heterogeneous tree-structured memory network (HTreeMN) for video question answering. Our model computes the attention on the parse tree of the natural language sentence. It not only enables the word-level attention but also alleviates the defects of the E-SA model [41] that the performance is poor on long questions. We also propose the heterogeneous nodes which distinguish the visual and the verbal words where the former
needs attention and the latter is only part of the language model. We further propose a hierarchical attention mechanism on the tree. We achieve the state-of-the-art results on video question answering.

ACKNOWLEDGMENT
This work was supported in part by the National Nature Science Foundation of China (Grant Nos: 61702405 and Grant Nos: 61602405) and in part by the National Youth Top-notch Talent Support Program. The experiments are supported by Chengwei Yao in the Experiment Center of the College of Computer Science and Technology, Zhejiang University.

REFERENCES
[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In Proceedings of the IEEE International Conference on Computer Vision, pages 2425–2433, 2015.
[2] S. Bird, E. Klein, and E. Lopez. Natural language processing with Python: analyzing text with the natural language toolkit. ” O’Reilly Media, Inc.”, 2009.
[3] M. Brysbaert, A. B. Warriner, and V. Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. Behavior Research Methods, 46(3):904, 2014.
[4] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia. Abcnn: An attention based convolutional neural network for visual question answering. arXiv preprint arXiv:1511.05960, 2015.
[5] Q. Chen, X. Zhu, Z. Ling, S. Wei, and H. Jiang. Enhancing and combining sequential and tree lstm for natural language inference. arXiv preprint arXiv:1609.06038, 2016.
[6] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? arXiv preprint arXiv:1606.03556, 2016.
[7] M. Denil, L. Bazzani, H. Larochelle, and N. de Freitas. Learning where to attend with deep architectures for image tracking. Neural computation, 24(8):2151–2184, 2012.
[8] A. Eriguchi, K. Hashimoto, and Y. Tsurenoka. Tree-to-sequence attentional neural machine translation. arXiv preprint arXiv:1603.00675, 2016.
[9] C. Fellbaum. WordNet. Wiley Online Library, 1998.
[10] Z. Guo, L. Gao, J. Song, X. Xu, J. Shao, and H. T. Shen. Attention-based lstm with semantic consistency for videos captioning. In Proceedings of the 2016 ACM on Multimedia Conference, pages 357–361. ACM, 2016.
[11] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In Advances in Neural Information Processing Systems, pages 1693–1701, 2015.
[12] J. Ilievski, S. Yan, and J. Feng. A focused dynamic attention model for visual question answering. arXiv preprint arXiv:1604.04185, 2016.
[13] M. Iyyer, J. L., Boyd-Graber, L. M. B. Claudino, R. Socher, and H. Daumé III. A neural network for factoid question answering over paragraphs. In EMNLP, pages 633–644, 2014.
[14] Z. Jie, X. Liang, J. Feng, X. Jin, W. Lu, and S. Yan. Tree-structured reinforcement learning for sequential object localization. In Advances in Neural Information Processing Systems, pages 127–135, 2016.
[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[16] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, pages 423–430. Association for Computational Linguistics, 2003.
[17] J. Li, M.-T. Luong, D. Jurafsky, and E. Hovy. When are tree structures necessary for deep learning of representations? arXiv preprint arXiv:1503.00185, 2015.
[18] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. Tgif: A new dataset and benchmark on animated gif description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4641–4650, 2016.
[19] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In Advances In Neural Information Processing Systems, pages 289–297, 2016.
[20] A. Mazaheri, D. Zhang, and M. Shah. Video fill in the blank with merging lstms. arXiv preprint arXiv:1610.04062, 2016.
[21] V. Mnih, N. Heess, A. Graves, et al. Recurrent models of visual attention. In Advances in neural information processing systems, pages 2204–2212, 2014.
[22] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In EMNLP, volume 14, pages 1532–1543, 2014.
[23] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4613–4621, 2016.
[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[25] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks.
In Advances in neural information processing systems, pages 2440–2448, 2015.

[26] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075, 2015.

[27] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4631–4640, 2016.

[28] Z. Teng and Y. Zhang. Bidirectional tree-structured lstm with head lexicalization. arXiv preprint arXiv:1611.06788, 2016.

[29] Z. Teng and Y. Zhang. Bidirectional tree-structured lstm with head lexicalization. arXiv preprint arXiv:1611.06788, 2016.

[30] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S.-C. Zhu. Joint video and text parsing for understanding events and answering queries. IEEE MultiMedia, 21(2):42–70, 2014.

[31] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4631–4640, 2016.

[32] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pages 133–138. Association for Computational Linguistics, 1994.

[33] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. arXiv, 1603, 2016.

[34] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In European Conference on Computer Vision, pages 451–466. Springer, 2016.

[35] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In ICML, volume 14, pages 77–81, 2015.

[36] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 21–29, 2016.

[37] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4631–4639, 2016.

[38] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4631–4639, 2016.

[39] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4584–4593, 2016.

[40] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank description generation and question answering. In Proceedings of the IEEE International Conference on Computer Vision, pages 2461–2469, 2015.

[41] K.-H. Zeng, T.-H. Chen, C.-Y. Chuang, Y.-H. Liao, J. C. Niebles, and M. Sun. Leveraging video descriptions to learn video question answering. In AAAI, 2017.

[42] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4995–5004, 2016.