XTQA: Span-Level Explanations for Textbook Question Answering

Jie Ma†, Member, IEEE, Qi Chai†, Jun Liu‡, Senior Member, IEEE, Qingyu Yin, Pinghui Wang†, Senior Member, IEEE, and Qinghua Zheng†, Member, IEEE

Abstract—Textbook question answering (TQA) is the task of correctly answering diagram or nondiagram (ND) questions given large multimodal contexts consisting of abundant essays and diagrams. In real-world scenarios, an explainable TQA system plays a key role in deepening humans’ understanding of learned knowledge. However, there is no work to investigate how to provide explanations currently. To address this issue, we devise a novel architecture toward span-level explanations for TQA (XTQA). In this article, spans are the combinations of sentences within a paragraph. The key idea is to consider the entire textual context of a lesson as candidate evidence and then use our proposed coarse-to-fine grained explanation extracting (EE) algorithm to narrow down the evidence scope and extract the span-level explanations with varying lengths for answering different questions. The EE algorithm can also be integrated into other TQA methods to make them explainable and improve the TQA performance. Experimental results show that XTQA obtains the best overall explanation result [mean intersection over union (mIoU)] of 52.38% on the first 300 questions of CK12-QA test splits, demonstrating the explainability of our method (ND: 150 and diagram: 150). The results also show that XTQA achieves the best TQA performance of 36.46% and 36.95% on the aforementioned splits. We have released our code in https://github.com/dr-majie/opentqa.

Index Terms—Explanation extracting (EE), question answering.

I. INTRODUCTION

Q

UESTION answering tasks, such as visual question answering (VQA) [1], [2] and machine reading comprehension (MRC) [3], [4], have attracted the extensive interest of researchers, due to their numerous real-world applications such as intelligent assistants. Recently, a new task called textbook question answering (TQA) [5] was proposed and it requires a system to answer diagram and nondiagram (ND) questions automatically given large multimodal contexts consisting of abundant essays and diagrams. Different from VQA and MRC, TQA uses both text and diagram inputs in the context and the question, which makes it a nontrivial task. Fig. 1 shows an example of the TQA task. In this example, a TQA system is required to provide the answers to questions for humans after learning the multimodal context of lesson “solids, liquids, gases, and plasmas” on the left. Humans will be perplexed in real-world education if they are only given the answers because they may not fully comprehend the knowledge involved in the questions. As a result, a desirable TQA system should provide answers as well as explanations for humans, allowing them to gain a better understanding of their learned knowledge. Although existing works [6], [7], [8] have made significant progress on the TQA performance, there is currently no work to investigate how to provide explanations to the best of our knowledge. A recent study [5] found that about 80% of the questions can be answered by using sentences in the context and we notice that these evidence spans (combinations of sentences in a paragraph) can also be regarded as explanations because they contain the key knowledge to answer the questions. For example, the span [9, 11] marked in green on the left of Fig. 1 can be provided for humans to explain why the TQA system chooses $D$ for

Fig. 1. Example of the TQA task. Questions with or without diagrams are shown on the right. Sentences marked in green on the left are the explanations for answering question 1. The number indicates the order of sentences within a passage.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
question 1, where 9 and 11 denote the start and end indices, respectively.

Inspired by this, we devise a novel architecture toward span-level explanations for TQA (XTQA), which considers the entire textual context of a lesson as candidate evidence and extracts span-level explanations using our proposed coarse-to-fine-grained explanation extracting (EE) algorithm. Concretely, we regard each paragraph of a lesson as a document and apply the embedding-based query expansion method [9] to choose the top $M$ paragraphs that are relevant to questions in the coarse-grained phase. In the fine-grained phase, the top $U$ span-level explanations are extracted from all candidate spans within the top $M$ paragraphs by computing the information gain of each span for questions. The larger information gain indicates the more uncertainty of questions reduced by spans. We consider the explanations extracted by EE to be the key context for assisting XTQA in predicting answers and improving the performance. Due to the lack of ground truth for explanations and their importance for TQA, we use the answer label to optimize EE. Furthermore, the EE algorithm can also be integrated into other methods to make them explainable and improve the TQA performance.

Experimental results show that XTQA obtains the best overall explanation result of 52.38% on the first 300 questions of the CK12-QA test split (ND questions: 150 and diagram questions: 150) [5]. The results also show that XTQA achieves the best TQA performance, improving the accuracy on the test split from 34.06% to 36.95%.

In summary, our contributions are mainly threefold.

1) We devise a novel TQA architecture, which considers the entire textual context of a lesson as candidate evidence and extracts span-level explanations with varying lengths for different questions. To the best of our knowledge, this article is the first work to investigate the explainability of TQA.

2) We propose the EE algorithm, which can also be integrated into other TQA methods to make them explainable and improve the TQA performance.

3) We conduct extensive experiments to explore how well XTQA and baselines + EE provide explanations and how much performance they can obtain on CK12-QA [5]. Extensive ablation studies and discussions are also carried out to analyze XTQA.

The remainder of this article is organized as follows. We introduce the related works in Section II. Section III describes the task formulation. We introduce the details of XTQA in Section IV. The experiments are discussed in Section V. Section VI introduces the conclusions.

II. RELATED WORK

In this section, we introduce the related works of three question answering tasks, including TQA, MRC, and VQA due to their similarities.

1 The TQA dataset is collected from http://www.ck12.org. In this article, we call the TQA dataset CK12-QA to distinguish TQA tasks from TQA datasets.

A. Textbook Question Answering

There have a few works [6], [7], [8] to study TQA. Li et al. [6] proposed instructor guidance with memory networks, which find contradictions between options and textual context to predict answers. Kim et al. [7] proposed a fusion graph convolutional network (GCN) to extract knowledge features and a self-supervised learning method to solve out-of-domain problems. Both the above papers did not report the accuracy of the test split and release their codes. Ma et al. [10] proposed a relation-aware fine-grained reasoning network, which builds diagram graphs based on dependency analyses and then applies question-guided attention mechanisms to reason over the graphs. Ma et al. [11] proposed a weakly supervised multitask learning framework for TQA to strengthen the diagram and text understanding. Gómez-Pérez and Ortega [8] proposed a pretrained TQA method ISAAQ based on transformer language models and top-down attention [12] to solve the multimodality understanding issue. They pretrained the textual ISAAQ on RACE [13], ARC-Easy, ARC-Challenge [14], and OpenBookQA [15] datasets and fine-tuned it on CK12-QA. Similarly, they pretrained the multimodal ISAAQ on VQA abstract scenes, VQA [16], and AI2D [17] datasets and fine-tuned it on CK12-QA. By comparison, XTQA tries to provide explanations for humans and it is trained only on CK12-QA.

B. Machine Reading Comprehension

It requires a machine to answer questions accurately given a textual context [18]. We classify MRC methods into two categories: single-hop and multihop reasoning.

Single-hop methods [19], [20], [21] use specific means such as attention mechanisms to perform interactions between queries and single paragraphs to predict answers. SeoKFH17 proposed the bidirectional attention flow network to learn query-aware context representations without the early summarization. Yuan et al. [20] reframed current static MRC environments as interactive and partially observed environments by restricting the context which a model observes at one time and used reinforcement learning to optimize the information-seeking agent. Zhang et al. [21] integrated the syntactic dependency of interest design into the self-attention network to strengthen the capacity of modeling the linguistic knowledge. However, the answers are very likely to be obtained from multiple paragraphs in real-life scenarios.

Multihop methods [3], [22], [23], [24] performs interactions between queries and multiple paragraphs to predict answers. Ding et al. [22] proposed CoqQA that builds a cognitive graph by an implicit extraction module and an explicit reasoning module to address the multihop question answering. Nie et al. [3] proposed a hierarchical pipeline model that reveals the importance of semantic retrieval to give general guidelines on the system design for MRC. Tang et al. [23] proposed a path-based GCN to perform multipath reasoning. Hu et al. [24] proposed a multitype multispan network, which combines a multitype answer predictor with a multispan extraction method to enhance the MRC performance. In comparison,
XTQA extracts evidence spans not only to enhance the TQA performance but also to provide span-level explanations for humans. The spans in this article are the combinations of sentences rather than words [24].

C. Visual Question Answering

It requires a machine to answer questions accurately given an image [16]. We classify VQA methods into three categories: joint embedding-based, attention mechanism-based, and explainable.

Joint embedding-based [25], [26], [27] methods use convolutional neural networks and recurrent neural networks to learn representations of images and questions, respectively, and then project them into a common space to predict answers. Fukui et al. [25] proposed the multimodal compact bilinear pooling method to learn joint input representations and project them into the answer space to predict answers. However, bilinear representations may limit the applicability to high-dimensional complex computation tasks. To address this issue, the low-rank bilinear pooling method using the Hadamard product [26] and factorized bilinear pooling [27] methods is proposed to learn multimodal representations efficiently. However, these methods may feed irrelevant or noisy information into the answer space.

Attention mechanism-based [2], [12], [28] methods assign different importance to input representations before information fusion by computing attention coefficients. Anderson et al. [12] proposed a combined bottom-up and top-down attention mechanism that computes attention at the level of salient image regions and objects. Li et al. [28] proposed a relation-aware graph attention network to learn question-adaptive multimodal representations. Khademi [2] devised a multimodal neural graph memory network to perform reasoning about the interactions of objects. Ma et al. [29] proposed a multitask learning framework to jointly optimize multimodal learning. However, these methods cannot provide explanations for humans.

Explainable methods [30], [31], [32], [33] give humans explanations with the help of specified means such as external knowledge and symbols. Fact-based visual question answering (FVQA) [30] queries the external knowledge base to obtain a supporting fact and predicts the answer. Park et al. [31] proposed a multimodal approach to explanations using post hoc justifications. Yi et al. [32] proposed a neural-symbolic VQA architecture that disentangles question and image understanding from reasoning. Based on this article, Mao et al. [33] proposed a neuro-symbolic reasoning module that executes generated programs on the latent scene representations to perform reasoning. The explanations of the above works are generated or extracted by complete supervision. By comparison, our model extracts span-level explanations with different lengths for different questions under the answer supervision rather than span supervision, i.e., weak supervision.

III. TASK FORMULATION

The TQA task can be classified into two categories: diagram question answering and ND question answering. In this section, we mainly introduce the task formulation of the diagram question answering due to their similarities.

Given a dataset $S$ consisting of $n$ quadruples $(c_i, q_i, d_i, A_i)$ with $c_i \in C$ representing multimodal contexts of a lesson, $q_i \in Q$ representing a question, $d_i \in D$ representing a diagram of $q_i$ and $A_i \in A$ representing candidate answers of $q_i$, the task can be denoted as follows:

$$\hat{a}_i = \arg \max_{a_{i,j} \in A_i} p(a_{i,j}|c_i, q_i, d_i; \theta)$$

where $\hat{a}_i$ is the predicted answer, $a_{i,j} \in A_i$ denotes the $j$th candidate answer of $q_i$, and $\theta$ denotes the trainable parameters. The dataset usually lacks the annotations for span explanations. Therefore, we apply the answer supervision $a_i$ to optimize the EE.

In this article, we only consider the textual context within $c_i$ due to the lack of visual context in some lessons. $N = (L(L + 1))/2$ is the number of candidate evidence spans supposing $c_i$ containing one paragraph with $L$ sentences. Candidate evidence span $e_{i,k}$ of $q_i$ is represented by its start $\text{START}(k)$ and end $\text{END}(k)$ indexes following [34], where $1 \leq k \leq N$ and $1 \leq \text{START}(k) \leq \text{END}(k) \leq L$. For example, if a paragraph is consisting of three sentences with 1, 2, and 3 denoting their indexes, there have six candidate spans, including $[1, 1], [1, 2], [1, 2, 3], [2, 1], [2, 3], \text{ and } [3]$ under the condition of not limiting the widths of spans. The span $[1, 2, 3]$ is denoted as $[1, 3]$. We optimize $\theta$ to obtain not only the predicted answer $\hat{a}_i$ but also the span-level explanation $\hat{e}_i$ of $q_i$.

IV. METHOD

In this section, we first give an overview of our method and then introduce the details of each module.

A. Overview

The architecture of XTQA with four modules is shown in Fig. 2. XTQA first obtains the sentence-level representations $q_i^*$ and $a_{i,j}^*$ of the question, candidate answer $q_i$ and $a_{i,j}$, respectively, in question/answer representing. Then, XTQA considers the entire textual context of a lesson as candidate evidence and obtains the representations $e_{i,k}^*$ of the top $U$ span-level explanations of $q_i$ and their indexes [START($k$), END($k$)] using our proposed EE algorithm in EE. Third, contrastive learning is first used in the TQA task to learn effective diagram representations $d_i'$ of $d_i$ in diagram representing. Finally, XTQA gives humans not only the predicted answer $\hat{a}_i$ but also the span-level explanation $\hat{e}_i$ to choose it after fusing the above multimodal information in answer predicting.

B. Question/Answer Representing

We use unidirectional gated recurrent units (GRUs) to obtain the $r_1$-dimensional word-level representations $q_i' \in \mathbb{R}^{X \times r_1}$ and $a_{i,j}' \in \mathbb{R}^{Y \times r_1}$ of $q_i$ and $a_{i,j}$, respectively, as follows:

$$q_i' = \text{GRU}_i(\text{embedding}(q_i))$$
$$a_{i,j}' = \text{GRU}_i(\text{embedding}(a_{i,j}))$$

where $q_i$ denotes the $i$th question, $a_{i,j}$ denotes the $j$th candidate answer of $q_i$, $X$ and $Y$ denote the maximum length of
Changes of State

Sublimation

Changes directly to a solid without going through the liquid state. The ice crystals are called frost. The process in which a solid changes directly to a gas is called sublimation. It occurs when the particles of a solid absorb enough energy to completely overcome the force of attraction between them. Dry ice (solid carbon dioxide, CO₂) is an example of a solid that undergoes sublimation. This is most likely to occur on sunny winter days...

q_i and a_i,j respectively, and embedding() is used to learn the word embeddings.

To obtain the r_1-dimensional sentence-level representations 𝑞_i'' ∈ ℝ^r_1 of q_i, a learned attention mechanism is applied as follows:

\[ \alpha = \text{softmax}(\text{MLP}_s(𝑞_i')) \]
\[ 𝑞_i'' = \sum_{u=1}^{x} \alpha_u \circ 𝑞_i'u \]  

(3)

where \( \alpha \in \mathbb{R}^x \) is the learned attention weight matrix by multilayer perceptrons (MLPs), \( \circ \) denotes the elementwise product, and \( 𝑞_i'u \in \mathbb{R}^r_1 \) is the u-th word representations of \( 𝑞_i \).

Similarly, we also use the above learned attention mechanism to learn the r_1-dimensional sentence-level representations \( a_i'' \in \mathbb{R}^r_1 \) of \( a_i,j \)

\[ \alpha = \text{softmax}(\text{MLP}_s(a_i')) \]
\[ a_i'' = \sum_{u=1}^{y} \alpha_u \circ a_i'j,u \]  

(4)

where \( a_i',j,u \in \mathbb{R}^r_1 \) is the u-th word representations of \( a_i,j \).

C. Explanation Extracting

Although the multimodal context \( c_i \) contains abundant essays with an average length of 788 words in CK12-QA, only a subset of sentences are required to answer \( q_i \) and these sentences can also be regarded as the explanations for \( q_i \). Inspired by this, XTQA first considers the entire textual context of a lesson as candidate evidence and then extracts the top U evidence spans from it as explanations using our proposed EE algorithm.

In the coarse-grained phase, we regard each paragraph of a lesson as a document and apply the embedding-based query expansion method [9] to narrow down the scope of textual contexts from a lesson to top M paragraphs \( p_i \) relevant to \( q_i \). \( p_i \in \mathbb{R}^{M \times L \times O} \) can be denoted as follows:

\[ p_i = \text{Query}(qi, c_i) \]  

(5)

where \( L \) is the maximum number of sentences in each paragraph and \( O \) is the maximum length of each sentence. The shared GRU_j in (2) is used to obtain the r_1-dimensional word-level representation \( p_i' \in \mathbb{R}^{M \times L \times r_1} \) of \( p_i \). We also use the shared learned attention mechanism in (3) to obtain the r_1-dimensional sentence-level representations \( p_i'' \in \mathbb{R}^{M \times L \times r_1} \) of \( p_i \) to match the next phase.

In the fine-grained phase, the top U span-level explanations are extracted from all candidate spans within top M paragraphs by computing the information gain of each span for questions. Specifically, the representations at start(START(k)) and end(END(k)) indexes are concatenated to obtain the representation \( e'_i,k \in \mathbb{R}^{M \times N \times 2r_1} \) of the candidate evidence span \( e_i,k \) as follows:

\[ e'_i,k = [p_i'_{\text{START}(k)} ; p_i''_{\text{END}(k)}] \]  

(6)

where \( N = (L(L + 1))/2 \) is the number of candidate evidence spans within each paragraph, \( 1 \leq k \leq N, 1 \leq \text{START}(k) \leq \text{END}(k) \leq L \), and \([;]\) denotes the concatenation. To match the following steps, \( e''_i,k \in \mathbb{R}^{M \times N \times r_1} \) is obtained by the average pooling AP with kernel size 2 x 1 on \( e'_i,k \) as follows:

\[ e''_i,k = \text{AP}(e'_i,k) \]  

(7)

XTQA computes the information gain \( g(q_i, e_i,k) \) of each candidate evidence span \( e_i,k \) for \( q_i \) to obtain the top U span-level explanations. \( g(q_i, e_i,k) \) can be obtained as follows:

\[ g(q_i, e_i,k) = H(q_i) - H(q_i|e_i,k) \]  

(8)

where \( H(q_i) \) is the entropy of \( q_i \), i.e., the uncertainty of \( q_i \), and \( H(q_i|e_i,k) \) is the conditional entropy of \( q_i \) given \( e_i,k \),
i.e., the uncertainty of $q_i$ given $e_{i,k}$. The larger information gain indicates the more uncertainty of $q_i$ reduced by $e_{i,k}$. $H(q_i)$ can be obtained as follows:

$$H(q_i) = \mathbb{E}[-\log(p(q_i))]$$

$$p(q_i) = \sigma(\text{MLP}_{\theta}(q_i'))$$

(9)

where $\mathbb{E}$ is the expected value operator, $p(q_i)$ denotes the probability of $q_i$ being answered accurately, $q_i'$ denotes the sentence-level representations of $q_i$, and $\sigma$ is the sigmoid function. $H(q_i|e_{i,k})$ can be obtained as follows:

$$H(q_i|e_{i,k}) = \mathbb{E}[-\log(p(q_i, e_{i,k}))]$$

$$p(q_i, e_{i,k}) = \sigma(\text{MLP}_{\theta}(\text{AP}([q_i'', e_{i,k}'']))))$$

(10)

where $p(q_i, e_{i,k})$ is the probability of $q_i$ being answered accurately given $e_{i,k}$ and AP is the average pooling with kernel size $2 \times 1$. A formal description about the algorithm is shown in Algorithm 1.

After obtaining the top $U$ span-level explanations and their representations $e''_i \in \mathbb{R}^{U \times n'}$, the learned attention mechanism in (3) is used to obtain the global span-level explanation representation $e''_i \in \mathbb{R}^n$.

**Algorithm 1 EE**

**Input:** question $q_i$, multi-modal context $e_i$.

**Output:** representation $e''_i$ of span-level explanation $e_i$ and its index.

1. Choose top $M$ paragraphs relevant to $q_i$ using Equation 5;
2. Construct the possible span according to the way described in Section III;
3. Obtain the candidate evidence span representation using Equation 6;
4. Obtain the global representation of the span using Equation 7;
5. Compute the entropy of $q_i$ using Equation 9;
6. Compute the conditional entropy of $q_i$ given each candidate evidence span $e_{i,k}$ using Equation 10;
7. Compute the information gain of each span for $q_i$ using Equation 8;
8. Select top $U$ span-level explanations according to the gain.

**D. Diagram Representing**

Effective diagram representations play a key role in improving the TQA performance. However, there has no annotation for diagrams in CK12-QA. Recently, self-supervised learning methods such as SimCLR [35] have made significant progress in image classification, which shows that they can learn the deep understanding of images. Inspired by this, we first pretrain CNNs such as ResNet on the diagrams within CK12-QA by contrastive learning [35] and fine-tune this module on the TQA task to learn the $r_2$-dimensional representation $d_i' \in \mathbb{R}^{r_2}$ of $d_i$ as follows:

$$d_i' = \text{CNNs}(d_i).$$

(11)

**E. Answer Predicting**

After the above modules’ processing, we obtain the word and sentence-level representations $q_i'$ and $q_i''$ of $q_i$, the diagram representation $d_i'$ of $d_i$, the sentence-level representation $a_{i,j}'$ of $a_{i,j}$, the span-level explanation representation $e_{i,k}''$ of $e_i$, and the indexes [START($k$), END($k$)] of spans. In general, multigrained or multilevel representations are beneficial for obtaining effective multimodality features. Therefore, they are used to obtain the global fusion feature $g_{i,j} \in \mathbb{R}^{9n}$ with $j$th candidate answer as follows:

$$g_{i,j} = [q_i''; d_i'; a_{i,j}''; e_i''; q_i''; a_{i,j}''; g_{i,j}''; g_{i,j}''; g_{i,j}''; s_{i,j}^\gamma]$$

$$g_{i,j}^\beta = \text{BAN}(g_{i,j}', d_i'), \quad g_{i,j}^\gamma = Wq_i'' \circ Wa_{i,j}''$$

$$s_{i,j} = Wq_i'' \circ Wa_{i,j}'' \circ s_{i,j}^\gamma$$

(12)

where BAN is the bilinear attention mechanism [36], $W \in \mathbb{R}^{n \times n'}$ is the learned weight matrix, $g_{i,j}^\beta, g_{i,j}^\gamma, g_{i,j}'' \in \mathbb{R}^n$ denote the pairwise similarity, and $g_{i,j}'' \in \mathbb{R}^n$ is the triplywise similarity.

To obtain the scores of candidate answers $s_i \in \mathbb{R}^{|A_i|}$, $g_i \in \mathbb{R}^{|A_i| \times 9n}$ is projected as follows:

$$s_i = \text{MLP}_c(g_i)$$

(13)

where $|A_i|$ denotes the number of candidate answers of $q_i$.

We regard the TQA task as a multiclass classification. Therefore, XTQA is optimized by the multiclass cross-entropy function as follows:

$$\mathcal{L} = - \sum_{i=1}^{n} y_i \log \hat{y}_i$$

(14)

where $n$ denotes the number of questions, $y_i \in \{0, 1\}^{|A_i|}$ denotes the true answers of $q_i$, $\hat{y}_i \in [0, 1]^{|A_i|}$ denotes the predicted probability of candidate $\hat{y}_i$, and softmax denotes the softmax function.

Eventually, not only the predicted answer $\hat{a}_i$ but also the span-level explanation $e_i$ is provided for humans.

**V. EXPERIMENTS**

In this section, we first introduce the experimental setups such as datasets and implement details. Then, we describe the results, including explanation and TQA accuracy. Third, ablation studies and discussions are introduced. Finally, we give a case study of XTQA.

**A. Datasets and Evaluations**

Currently, there exist two TQA datasets, including CK12-QA [5] and AI2D [17]. Most of the previous works [6, 7, 37] are only evaluated on CK12-QA except ISAAQ [8]. Due to the lack of multimodal contexts of AI2D, XTQA cannot be applicable to this dataset to extract explanations. We will explore how to generate explanations only given questions and diagrams in the future. Following the previous works, we evaluate XTQA on CK12-QA [5] that consists of
1076 lessons with 78,338 sentences and 3455 diagrams. The lessons are obtained from the physical science, life science, and Earth science textbooks of the middle school online curricula. The dataset is split into a training set with 666 lessons and 15,154 questions, a validation dataset with 200 lessons and 5309 questions, and a test set with 210 lessons and 5797 questions. Among the total 26,260 questions, 12,567 of them have an accompanying diagram. There are four candidate answers for each diagram question. The ND questions can be classified into two categories: true/false (T/F) with two candidate answers and multiple choice (MC) with 4–7 candidate answers.

To estimate the results of span-level explanations, we employ the mean intersection over union (mIoU) that is always used in object detection [38] and segmentation [39] as metrics. The TQA accuracy is obtained by checking whether the prediction is the same as the ground truth.

### B. Implementation Details

In question/answer representing, we use BERT [40] to obtain 768-D word embeddings and apply unidirectional one-layer GRUs with \( r_1 = 1024 \) hidden units to encode questions and candidate answers. The shared MLPs (FC(1024)-Dropout(0.2)-FC(1)) is used to learn attention coefficients. In EE, the pytcune is used to conduct paragraphs indexing and searching. The maximum number of paragraphs \( M \), the maximum number of sentences within each paragraph \( L \), the maximum length of each sentence \( O \), and the maximum number of span-level explanations \( U \) are set to 1/1, 5/15, 20/15, and 1/1 for ND/diagram question answering, respectively. We set the maximum widths \( W \) of candidate evidence spans to 2. In diagram representing, we resize the diagrams to 224 due to the different sizes of them in the dataset. To obtain \( r_2 = 2048 \)-D diagram representations, we first train the SimCLR on the diagrams in CK12-QA with default hyperparameters and then fine-tune the pretrained model by the task-specific supervision (TQA). In answer predicting, the MLPc (FC(2048)-ReLU-Dropout(0.2)-FC(1)) is used to obtain the candidate answer scores.

XTQA is trained by the Adam optimizer with \( \beta_1 = 0.9 \) and \( \beta_2 = 0.98 \). The base learning rate is \( \min(2.5 r^{-4}, 1\cdot e^{-4}) \), where \( r \) is the current epoch. The rate is decayed by 0.1 after eight epochs. XTQA converges at the end of the 10th epoch with the batch size 2. Parameters of XTQA are initialized by the Pytorch default initialization with the fixed seed 666. All the experiments are run on one NVIDIA’s Tesla V100 card.

### C. Explanation Results

To the best of our knowledge, XTQA is the first method to explore TQA explanations. We select five previous state-of-the-art methods that focus on multimodality fusion and match well with our proposed EE algorithm as baselines. The introductions of them are given as follows.

1) MFB [27] is a multimodal factorized bilinear pooling approach, which aims at addressing the high dimensionality of the output features and the huge number of parameters caused by bilinear pooling-based models [41].

2) MUTAN [42] is a multimodal tensor-based decomposition approach with a low-rank matrix constraint. It also aims at addressing the huge dimensionality issue.

3) BAN [36] is a bilinear attention network that aims at learning effective interactions between images and questions using the proposed bilinear attention mechanism.

4) MCAN [1] is a deep modular coattention network. It aims at obtaining sufficient multimodality interactions by modularly composing the self-attention of questions and images, as well as the question-guided attention of images.

5) CMR [43] is a cross-modality relevance network, which learns the relevance representations between entities of input modalities and models the higher order relevance between entity relations to perform language and vision reasoning. It is the current state-of-the-art method for VQA.

To explore how well XTQA and baselines + EE provide explanations for humans, we manually annotate span-level explanations for the first 150 ND and diagram questions of the validation split and test split, respectively. Note that we do not use these annotations to train XTQA and baselines + EE. We apply mIoU to evaluate their performance. For example, if the indexes of a predicted span are (3, 5), the IoU value is 0.67. We do not consider whether the question is accurately answered here.

Tables I and II show the results on the validation and test split, respectively. It can be seen that each method provides more accurate explanations for T/F questions compared with the results on MC questions within the validation split. However, the results do not have the same trend on the test split.

| Model       | ND T/F | ND MC | ND All | Diagram | All   |
|-------------|--------|-------|--------|---------|-------|
| MFB [27]+EE | 50.96  | 40.25 | 44.46  | 45.68   | 45.07 |
| MUTAN [42]+EE | 52.36 | 43.39 | 46.92  | 43.36   | 45.14 |
| BAN [36]+EE | 56.38  | 42.66 | 48.06  | 47.20   | 47.63 |
| MCAN [1]+EE | 53.21  | 43.58 | 47.37  | 48.69   | 48.03 |
| CMR [43]+EE | 51.82  | 41.20 | 45.38  | 51.23   | 48.30 |
| XTQA        | 58.69  | 40.59 | 47.71  | 52.41   | 50.06 |

1 EE denotes our proposed explanation extracting algorithm.
2 ND All = ND T/F \( \lor \) ND MC and All = ND All \( \lor \) Diagram.

| Model       | ND T/F | ND MC | ND All | Diagram | All   |
|-------------|--------|-------|--------|---------|-------|
| MFB [27]+EE | 44.02  | 46.98 | 45.93  | 51.05   | 48.49 |
| MUTAN [42]+EE | 48.89 | 48.52 | 48.88  | 51.35   | 49.12 |
| BAN [36]+EE | 41.86  | 51.89 | 49.06  | 51.56   | 50.31 |
| CMR [43]+EE | 50.11  | 47.25 | 48.26  | 52.86   | 50.56 |
| XTQA        | 55.75  | 49.88 | 51.95  | 52.80   | 52.38 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
This may be caused by the different data distributions between the validation and the test split. For example, the knowledge of lesson Earth science and its branches is mutually exclusive. XTQA outperforms the best baseline MFB [27] by 1.29% and CMR [43] + EE achieves the best results. For ND questions, XTQA outperforms the best baseline MCAN [1] and the best baselines + EE by 1.36% and 0.17%, respectively. For diagram questions, XTQA outperforms the best baseline MFB [27] by 1.29% and CMR [43] + EE achieves the best results. For ND questions, XTQA outperforms the best baseline MCAN and the best baselines + EE by 1.22% and 0.55%, respectively. For T/F questions, XTQA outperforms the best baseline MFB and the best baselines + EE by 6.51% and 3.51%, respectively. XTQA has the worst performance on the MC questions, which may be caused by the different data distributions between T/F (two candidate answers) and MC (4–7 candidate answers) questions. In this article, we regard both the ND T/F and ND MC subtasks as the multiclass classification, i.e., padding the number of candidate answers of ND T/F questions into seven following previous works [6]. In this way, we do not need to devise a specific model for ND T/F and ND MC, respectively. Note that all the baselines perform information fusion between the top 1 paragraph and questions for ND questions considering the specificity of the TQA task.

In Table IV, we can see that XTQA outperforms the best baseline CMR [43] and the best baselines + EE by 2.89% and 0.54% on the total questions of the test split, respectively. For diagram questions, XTQA outperforms the best baseline CMR by 3.80% and CMR + EE achieves the best result. For ND questions, XTQA is superior to the best baselines + EE by 1% and achieves the best accuracy. For TF questions, XTQA outperforms the best baseline CMR and the best baselines + EE by 4.20% and 2.09%, respectively. For MC questions, XTQA outperforms the best baseline MCAN [1] and CMR by 0.25% and CMR + EE achieves the best result. In short, XTQA achieves the best performance on the two splits, which shows the effectiveness of our method. In addition, the results also demonstrate that EE can not only enhance the explainability of baselines but also the TQA performance of them.

To the best of our knowledge, existing TQA methods except ISAAQ and RAFR lack the results on the CK12-QA test split. ISAAQ achieves the current state-of-the-art results based on large dataset pretraining, large pretrained model fine-tuning, and ensemble learning. RAFR achieves the modest results with training only on CK12-QA and without large pretrained model fine-tuning and ensemble learning. To fairly compare with ISAAQ, we make the following minor changes for our method as follows.

1) Following ISAAQ, we apply RoBERTa to obtain the word-level representations \( q'_i \in \mathbb{R}^{X \times r_1} \) of \( q_i \), which is similar to (2). The word-level representation \( p'_i \in \mathbb{R}^{M \times L \times D \times r_1} \) of \( p_i \) is also learned by RoBERTa.

2) We concatenate \( q_i \) and \( a_{i,j} \) and apply RoBERTa to learn a joint sentence-level representation \( qa_{i,j} \in \mathbb{R}^{r_1} \), which is different from (3).

3) We do not use the multimodal fusing in (12). A question-explanation guided gate mechanism is proposed to learn the attended diagram representation \( d''_{i,j} \in \mathbb{R}^{r_1} \) of \( qa_{i,j} \). Then, we concatenate them to obtain the fusion information \( g_{i,j} \in \mathbb{R}^{r_1} \). The above steps can be denoted as follows:

\[
\begin{align*}
   a_{i,j} &= \sigma(W_a([qa_{i,j}; e''_{i,j}])) \\
   d''_{i,j} &= a_{i,j} \odot (W_d d'_i) \\
   g_{i,j} &= [qa_{i,j}; e''_{i,j}; d''_{i,j}]  
\end{align*}
\]  

\((15)\)
TABLE V
COMPARISON WITH CURRENT STATE-OF-THE-ART TQA METHODS ON THE VALIDATION SPLIT

| LMF  | Model      | ND T/F | ND MC | ND All | Diagram | All  |
|------|------------|--------|-------|--------|---------|------|
| No   | RAFF [10]  | 53.63  | 36.67 | 43.53  | 32.85   | 37.85|
| No   | XTQA       | 58.24  | 30.33 | 41.32  | 32.05   | 36.46|
| Yes  | ISAQA [8]  | 72.67  | 54.77 | 61.64  | 39.73   | 49.94|
| Yes  | XTQA-V2    | 76.65  | 57.65 | 65.15  | 46.85   | 55.56|

1 LMF denotes whether a specific method uses pre-trained language model fine-tuning. ND All = ND T/F ∪ ND MC and All = ND All ∪ Diagram.
2 RAFF and XTQA regard ND T/F and ND MC as a multi-class classification, which means the non-diagram questions do not be distinguished manually. XTQA-V2 and ISAQA regards ND T/F as a binary classification and ND MC as a multi-class classification.

TABLE VI
COMPARISON WITH CURRENT STATE-OF-THE-ART TQA METHODS ON THE TEST SPLIT

| LMF  | Model      | ND T/F | ND MC | ND All | Diagram | All  |
|------|------------|--------|-------|--------|---------|------|
| No   | RAFF [10]  | 52.75  | 34.38 | 41.03  | 30.47   | 35.04|
| No   | XTQA       | 55.22  | 33.40 | 41.67  | 33.34   | 36.95|
| Yes  | ISAQA [8]  | 72.57  | 55.74 | 62.00  | 35.18   | 46.80|
| Yes  | XTQA-V2    | 75.88  | 61.56 | 66.76  | 41.04   | 52.16|

TABLE VII
ABLATION RESULTS (% ACCURACY) ON THE VALIDATION SPLIT. Δ DENOTES THE ACCURACY REDUCTION WITHOUT THE SPECIFIC MODULE

| Models                        | ND All | Δ   | Diagram | Δ   | All  | Δ   |
|-------------------------------|--------|-----|---------|-----|------|-----|
| XTQA                          | 41.32  |      | 32.05   | 56.46|
| w/o contrastive learning      | -      |      | -       | -   | -    | -   |
| w/o fine-tuning ResNet        | -      | 2.03| -1.69   | 35.57| -0.89|
| w/o BERT embedding            | 41.20  | -0.12| 29.82  | -2.23| 35.24| -1.22|
| w/o span-level explanation    | 38.85  | -2.37| 30.36  | 1.09 | 34.45| -2.01|

1 ND All = ND T/F ∪ ND MC and All = ND All ∪ Diagram.

We call this method XTQA-V2. We train it and ISAQA only on CK12-QA to make a fair comparison. ISAQA [8] employs information retrieval, next sentence prediction, and nearest neighbors to extract the most related paragraphs of questions. Therefore, we use the average accuracy as the final result. The results on the validation and test splits are shown in Tables V and VI, respectively. We can see that RoBERTa-based models significantly outperform XTQA and RAFF that do not use the large language model fine-tuning. This is caused by large parameters and prior transfer knowledge.

E. Ablation Studies

We perform ablation studies on the validation split shown in Table VII to analyze the effectiveness of each module.

1) W/O Contrastive Learning: XTQA does not use contrastive learning to pretrain ResNet but uses the answer label to fine-tune it in the experiment. The accuracy on the diagram questions drops by 2.03%, which demonstrates that contrastive learning can be used to learn effective diagram representations except learning image representations.

where $\sigma$ is the sigmoid function, $W_a \in \mathbb{R}^{r_1 \times 2r_1}$ and $W_d \in \mathbb{R}^{r_1 \times r_2}$ are learned weight matrices, $\alpha_{i,j}$ denotes the gate weight, $d_i'$ is the diagram representation learned by (11), and $e_i''$ denotes the explanation representation obtained by (1).

We call this method XTQA-V2. We train it and ISAQA on only CK12-QA to make a fair comparison. ISAQA [8] employs information retrieval, next sentence prediction, and nearest neighbors to extract the most related paragraphs of questions. Therefore, we use the average accuracy as the final result. The results on the validation and test splits are shown in Tables V and VI, respectively. We can see that RoBERTa-based models significantly outperform XTQA and RAFF that do not use the large language model fine-tuning. This is caused by large parameters and prior transfer knowledge.

F. Algorithm Analyses

We conduct the experiments shown in Table VIII on the validation split to explore the effect of $M$, $U$, and span

Fig. 3. Explanation result (%) of XTQA under the condition of answering questions correctly and incorrectly. These questions are introduced in Section V-C.

Fig. 4. Statistics of instructional diagrams in all lessons. The diagram is the image included in multimodal contexts of lessons.

2) W/O Fine-Tuning ResNet: The ResNet is pretrained by contrastive learning but not fine-tuned using the answer label in the experiment. The accuracy on the diagram questions drops by 1.69%, which shows that effective diagram representation is important to improve the TQA performance.

3) W/O BERT Embedding: XTQA does not apply BERT embeddings to initialize the weight of embedding layers and applies the PyTorch default initialization in the experiment. The accuracy on the ND and diagram questions drops by 0.12% and 2.23%, respectively, which shows the significant performance difference of the BERT embeddings on the different task. This may be caused by unstable optimization on the diagram question answering because there exists a large vocabulary with 19,556 words but few training data with 6501 diagram questions.

4) W/O Span-Level Explanation: XTQA does not apply the fine-grained explanation but uses the coarse-grained top $M$ paragraphs in the experiment. The accuracy on ND and diagram questions drops by 2.37% and 1.69%, respectively, which shows the importance of span-level explanations. Moreover, the accuracy on the whole questions decreases the most compared with the ablation study of other modules.

In short, each component makes its contributions to the performance of XTQA and our proposed coarse-to-fine grained EE algorithm plays the biggest role.
Fig. 5. Case studies to show the strengths and weaknesses of XTQA. The left shows the multimodal context of lesson seafloor spreading. The middle shows the EE processes by our proposed algorithm. The right shows the questions of this lesson. These cases come from the validation split.

### TABLE VIII
**Algorithm Analysis (% Accuracy) on the Validation Split. △ denotes the Accuracy Reduction**

| M | U | W | ND All | Diagram | All |
|---|---|---|--------|---------|-----|
| 1 | – | – | 38.95  | 30.36   | 34.45 | 
| 1 | 1 | 1 | 40.52  | 31.85   | 35.97 | 
| 1 | 1 | 2 | 41.32  | 32.05   | 36.46 | 
| 1 | 2 | 2 | 41.47  | 31.60   | 36.30 | 
| 1 | 2 | 3 | 41.59  | 31.72   | 36.57 | 
| 2 | 1 | 3 | 39.93  | 30.68   | 34.38 | 
| 2 | 2 | 2 | 41.41  | 31.78   | 36.37 | 
| 2 | 2 | 3 | 41.35  | 31.38   | 36.13 | 
| 2 | 3 | 3 | 41.06  | 31.25   | 35.92 | 

1. Top M paragraphs, top U spans in each paragraph and span width with W.

width of Algorithm 1 on the TQA accuracy. All the other hyperparameters are the same in these experiments. It is clear that choosing the top \( M = 1 \) paragraph is the best way to answer ND and diagram questions. Our method performs better on ND questions but worse on diagram questions as the number of spans increases. The preceding description also applies to span width. We can conclude that the long textual explanations would interfere with answering diagram questions.

**G. Discussion**

We use the answer label to optimize the EE due to the lack of ground truth for explanations in CK12-QA, i.e., weakly supervised learning. To further analyze the effectiveness of this optimization, we conduct experiments on the first 150 ND and diagram questions. Our method performs better on ND questions but worse on diagram questions as the number of spans increases. The preceding description also applies to span width. We can conclude that the long textual explanations would interfere with answering diagram questions.

**H. Case Studies**

We conduct the case studies as shown in Fig. 5 to present the strengths and weaknesses of our method intuitively.

1. **Strengths**: XTQA can provide the explicit span-level explanations with different lengths for answering different questions. For example, XTQA provides the explanation of length 1 of the diagram question and provides the explanation of length 2 of the ND question for humans, as shown in the middle of Fig. 5.

2. **Weaknesses**: If the coarse grained algorithm makes errors, it will cause the failure to find span-level explanations.
in the fine-grained phase. For example, XTQA finds the wrong top 1 paragraph for the ND question as shown in the middle of Fig. 5, which causes the failure to find the explanations.

VI. CONCLUSION

In this article, we propose a novel architecture toward span-level XTQA. It considers the entire context of a lesson as level XTQA. It considers the entire context of a lesson as top 1 paragraph for the ND question as shown in the middle of the graph.

Experimental results show that XTQA obtains the best explanation result and improving their TQA performance. However, the proposed coarse-to-fine grained EE algorithm to extract span-level explanations with varying lengths for different questions. The results also demonstrate that the EE algorithm can be integrated into other TQA methods to enable them to have explainability and improve their TQA performance.

In the future, the following directions will be explored.

1) Error reduction of the coarse-grained algorithm may improve the accuracy of EE. For example, we can fine-tune the large pre-trained language model to retrieve the closest paragraph. We will explore how to devise an end-to-end architecture to optimize the process of the coarse-grained EE.

2) External knowledge helps to improve the performance of other tasks such as named entity recognition and VQA. An analysis about CK12-QA [5] also shows that about 10% of the questions require external knowledge to answer. We will investigate how to integrate the external knowledge into XTQA to improve its performance.

ACKNOWLEDGMENT

The authors are deeply indebted to Junlin Ji, Yi Huang, and Jianlong Zhou for their direct and indirect help.

REFERENCES

[1] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian, “Deep modular co-attention networks for visual question answering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6281–6290.

[2] M. Khademi, “Multimodal neural graph memory networks for visual question answering,” in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 7177–7188.

[3] Y. Nie, S. Wang, and M. Bansal, “Revealing the importance of semantic retrieval for machine reading at scale,” in Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP), 2019, pp. 2553–2566.

[4] A. Saxena, A. Tripathi, and P. Talukdar, “Improving multi-hop question answering over knowledge graphs using knowledge base embeddings,” in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 4498–4507.

[5] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi, “Are you smarter than a sixth grader? Textbook question answering for multimodal machine comprehension,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5376–5384.

[6] J. Li, H. Su, J. Zhu, S. Wang, and B. Zhang, “Textbook question answering under instructor guidance with memory networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3655–3663.

[7] D. Kim, S. Kim, and N. Kwak, “Textbook question answering with multi-modal context graph understanding and self-supervised open-set comprehension,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 3568–3584.

[8] J. M. Gómez-Pérez and R. Ortega, “ISAAQ-mastering textbook questions with pre-trained transformers and bottom-up and top-down attention,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020, pp. 5469–5479.

[9] S. Kuzi, A. Shokt, and O. Kurland, “Query expansion using word embeddings,” in Proc. 25th ACM Int. Conf. Inf. Knowl. Manage., Oct. 2016, pp. 1929–1932.

[10] J. Ma, J. Liu, Y. Wang, J. Li, and T. Liu, “Relation-aware fine-grained reasoning network for textbook question answering,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 1, pp. 15–27, Jan. 2023.

[11] J. Ma, Q. Chai, J. Huang, J. Liu, Y. You, and Q. Zheng, “Weakly supervised learning for textbook question answering,” IEEE Trans. Image Process., vol. 31, pp. 7378–7388, 2022.

[12] P. Anderson et al., “Bottom-up and top-down attention for image captioning and visual question answering,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jan. 2020, pp. 6077–6086.

[13] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, “RACE: Large-scale reading comprehension dataset from examinations,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2017, pp. 785–794.

[14] P. Clark et al., “Think you have solved question answering? Try ARC, the AI2 reasoning challenge,” 2018, arXiv:1803.05457.

[15] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, “Can a suit of armor conduct electricity? A new dataset for open book question answering,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 2381–2391.

[16] S. Antol et al., “VQA: Visual question answering,” in Proc. IEEE Int. Conf. Vis. (ICCV), Dec. 2015, pp. 2425–2433.

[17] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi, “A diagram is worth a dozen images,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 235–251.

[18] W. G. Lehnert, “The process of question answering,” Ph.D. dissertation, Dept. Comput. Sci., Yale Univ., New Haven, CT, USA, 1977.

[19] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” in Proc. 5th Int. Conf. Learn. Represent., 2017, pp. 1–13.

[20] X. Yuan, J. Fu, M.-A. Côte, Y. Tay, C. Pal, and A. Trischler, “Interactive machine comprehension with information seeking agents,” in Proc. 58th Annu. Meeting Assoc. for Comput. Linguistics, 2020, pp. 2325–2338.

[21] Z. Zhang, Y. Wu, J. Zhou, S. Duan, H. Zhao, and R. Wang, “SG-Net: Syntax-guided machine reading comprehension,” in Proc. 34th AAAI Conf. Artif. Intell., 2020, pp. 9305–9311.

[22] M. Ding, C. Zhou, Q. Chen, H. Yang, and J. Tang, “Cognitive graph for multi-hop reading comprehension at scale,” in Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 2694–2703.

[23] Z. Tang, Y. Shen, X. Ma, W. Xu, J. Yu, and W. Lu, “Multi-hop reading comprehension across documents with path-based graph convolutional network,” in Proc. 29th Int. Joint Conf. Artif. Intell., Jul. 2020, pp. 3905–3911.

[24] M. Hu, Y. Peng, Z. Huang, and D. Li, “A multi-type multi-span network for reading comprehension that requires discrete reasoning,” in Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP), 2019, pp. 1596–1606.

[25] 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 Proc. Conf. Empirical Methods Natural Lang. Process., 2016, pp. 457–468.

[26] J. Kim, K. W. On, W. Lim, J. Kim, J. Ha, and B. Zhang, “Hadamard product for low-rank bilinear pooling,” in Proc. 5th Int. Conf. Learn. Represent., 2017, pp. 1–9.

[27] Z. Yu, J. Yu, J. Fan, and D. Tao, “Multi-modal factorized bilinear pooling with co-attention learning for visual question answering,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1839–1848.

[28] L. Li, Z. Gan, Y. Cheng, and J. Liu, “Relation-aware graph attention network for visual question answering,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 10312–10321.

[29] J. Ma, J. Liu, Q. Lin, B. Wu, Y. Wang, and Y. You, “Multitask learning for visual question answering,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 3, pp. 1380–1394, Mar. 2023.

[30] P. Wang, Q. Wu, C. Shen, A. Dick, and A. van den Hengel, “FFVA: Fact-based visual question answering,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 10, pp. 2413–2427, Oct. 2018.

[31] D. H. Park et al., “Multimodal explanations: Justifying decisions and pointing to the evidence,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8779–8788.
[32] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, “Neural-symbolic VQA: Disentangling reasoning from vision and language understanding,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 1031–1042.

[33] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, “The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision,” in Proc. Int. Conf. Learn. Represent., 2019, pp. 1–28.

[34] J. Ma et al., “Jointly optimized neural coreference resolution with mutual attention,” in Proc. 13th Int. Conf. Web Search Data Mining, Jan. 2020, pp. 402–410.

[35] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” 2020, arXiv:2002.05709.

[36] J.-H. Kim, J. Jun, and B.-T. Zhang, “Bilinear attention networks,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 1564–1574.

[37] M. Haurilet, Z. Al-Halah, and R. Stiefelhagen, “MoQA—a multi-modal question answering architecture,” in Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops, in Lecture Notes in Computer Science, vol. 11132. Munich, Germany: Springer, 2018, pp. 106–113.

[38] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[39] Y. Zhu et al., “Improving semantic segmentation via video propagation and label relaxation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 8848–8857.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., vol. 1, Jun. 2019, pp. 4171–4186.

[41] J. B. Tenenbaum and W. T. Freeman, “Separating style and content,” in Proc. Adv. Neural Inf. Process. Syst., 1997, pp. 662–668.

[42] H. Ben-younes, R. Cadene, M. Cord, and N. Thome, “MUTAN: Multi-modal tucker fusion for visual question answering,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2631–2639.

[43] C. Zheng, Q. Guo, and P. Kordjamshidi, “Cross-modality relevancy for reasoning on language and vision,” in Proc. 5th Int. Conf. Meeting Assoc. Comput. Linguistics, Stroudsburg, PA, USA: Association for Computational Linguistics, 2020, pp. 7642–7651.

Jun Liu (Senior Member, IEEE) received the B.S. degree in computer science and technology and the Ph.D. degree in systems engineering from Xi’an Jiaotong University, Xi’an, China, in 1995 and 2004, respectively.

He is currently a Professor with the Department of Computer Science, Xi’an Jiaotong University. He has authored more than 90 research papers in various journals and conference proceedings. His research interests include natural language processing (NLP) and e-learning.

Dr. Liu received the Best Paper Awards at IEEE International Symposium on Software Reliability Engineering (ISSRE) 2016 and IEEE International Conference on Big Knowledge (ICBK) 2016. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS since 2020. He has served as a Guest Editor for many technical journals, such as INFORMATION FUSION and IEEE SYSTEMS JOURNAL. He also acted as a conference/workshop/track chair for numerous conferences.

Qi Chai received the B.E. degree from Xi’an Jiaotong University, Xi’an, China, in 2021, where he is currently pursuing the M.E. degree with the School of Computer Science and Technology.

His research interests include multimodal question answering, image processing, and natural language processing.

Jun Liu

Qinghua Zheng (Member, IEEE) received the B.S. degree in computer software, the M.S. degree in computer organization and architecture, and the Ph.D. degree in system engineering from Xi’an Jiaotong University, Xi’an, China, in 1990, 1993, and 1997, respectively.

He was a Post-Doctoral Researcher with Harvard University, Cambridge, MA, USA, in 2002. He is currently a Professor with the Department of Computer Science and Technology, Xi’an Jiaotong University. His research interests include knowledge engineering of big data and software trustworthiness evaluation.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.