Multi-Modal Graph Neural Network for Joint Reasoning on Vision and Scene Text

Difei Gao\textsuperscript{1,2}\textsuperscript{*}, Ke Li\textsuperscript{1,2}\textsuperscript{*}, Ruiping Wang\textsuperscript{1,2}, Shiguang Shan\textsuperscript{1,2}, Xilin Chen\textsuperscript{1,2}

\textsuperscript{1}Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
\textsuperscript{2}University of Chinese Academy of Sciences, Beijing, 100049, China

\{difei.gao, ke.li\}@vipl.ict.ac.cn, \{wangruiping, sgshan, xlchen\}@ict.ac.cn

Abstract

Answering questions that require reading texts in an image is challenging for current models. One key difficulty of this task is that rare, polysemous, and ambiguous words frequently appear in images, e.g., names of places, products, and sports teams. To overcome this difficulty, only resorting to pre-trained word embedding models is far from enough. A desired model should utilize the rich information in multiple modalities of the image to help understand the meaning of scene texts, e.g., the prominent text on a bottle is most likely to be the brand. Following this idea, we propose a novel VQA approach, Multi-Modal Graph Neural Network (MM-GNN). It first represents an image as a graph consisting of three sub-graphs, depicting visual, semantic, and numeric modalities respectively. Then, we introduce three aggregators which guide the message passing from one graph to another to utilize the contexts in various modalities, so as to refine the features of nodes. The updated nodes have better features for the downstream question answering module. Experimental evaluations show that our MM-GNN represents the scene texts better and obviously facilitates the performances on two VQA tasks that require reading scene texts.

1. Introduction

The texts in a scene convey rich information that is crucial for performing daily tasks like finding a place, acquiring information about a product, etc. An advanced Visual Question Answering (VQA) model which is able to reason over scene texts and other visual contents could have extensive applications in practice, such as assisting visually impaired users, and education of children. Our focus in this paper is to endow VQA models the ability of better representing the image containing the scene texts, to facilitate the performances of answering on VQA tasks \cite{44, 8} that requires reading in images.

What are the unique challenges of modeling scene texts compared to the pure visual entities (such as objects and scenes) and the natural language texts (sentences or phrases)? A scene text inherently contains information in multiple modalities, visual information, including color, shape, and semantic information, e.g., “New York” is the name of a city, and numeric information for numbers, e.g., “65” is larger than “50”. These types of information are
Q1: What brand is the drink?
A: EPIC

Q2: Who’s mug is it?
A: Ged’s

Q3: What is the largest measurement shown on the ruler?
A: 40

Figure 2. Three example questions requiring leveraging different types of multi-modal contexts to answer the questions. Q1: the model should use visual context (the “EPIC” is a prominent word on the bottle) to infer the semantics of the word “EPIC”. Q2: the model should infer that “Ged” indicates the owner of the mug by using the semantic context of the word. Q3: The model should be able to utilize the semantic of the numbers to depict more informative numeric relations between numbers, such as “40” is the largest measurement among “40”, “30”.

frequently used in answering daily questions. For example in Fig. 1, Q2 requires the model to find the target scene text with its visual information; Q3 needs the model to understand the semantic of “65” which indicates the amount of the money; Q4 requires the understanding of numeric relation between numbers. Therefore, to correctly answer the questions involving scene texts, it is indispensable to clearly depict each modality of the scene texts. In addition, among these three modalities, it is more difficult to determine the semantics of scene texts, because the scene texts encountered in daily environments have a large possibility to be unknown, rare or polysynonymous words, e.g., the name of a product “STP” as shown in Fig. 1. To tackle this problem, the model should be able to determine the semantics of these texts beyond only using word embedding [38, 26] pre-trained on a text corpus. In this paper, we propose to teach the model how to utilize the context in different modalities in surrounding of the words to determine their meanings like a human, that is, 1) visual context: the prominent word on the bottle is most likely to be its brand, as Q1 in Fig. 1 and Q4 in Fig. 2, 2) semantic context: the surrounding texts of a rare or ambiguous word may help to infer its meaning, e.g. Q2 shown in Fig. 2. In addition, utilizing semantics of numbers can also depict more informative numeric relations between numbers, as Q3 shown in Fig. 2.

Following the aforementioned ideas, we propose a novel approach, Multi-Modal Graph Neural Networks (MM-GNN), to obtain a better representation of the multi-modal contents in an image and facilitate question answering. Our proposed MM-GNN contains three sub-graphs for representing three modalities in an image, i.e., visual modality for visual entities (including texts and objects), semantic modality for scene texts, and numeric modality for number-related texts, as shown in Fig. 3. The initial representations of nodes in three graphs are obtained from priors, such as word embedding learned from the corpora and Faster R-CNN features. Then, MM-GNN dynamically updates the representations of nodes by three attention-based aggregators, corresponding to utilizing three typical types of contexts in Fig. 2. These aggregators calculate the relevance scores of two nodes considering their visual appearances and layout information in the image, together with questions. Besides relevance between nodes, by attending on the basis of layout information, we are actually linking texts to their physical carriers (the object a text is printed or carved on); and given language hints, attention models can pass messages more accurately, by considering the directives implied by questions. Three different aggregators guide the message passing from one modality to another modality (or to itself) to leverage different types of contexts to refine the node features in a certain order. The updated representation contains richer and more precise information, facilitating the answering model to attend to the correct answer.

Finally, we conduct experiments with our proposed MM-GNN and its variants on two recently proposed datasets TextVQA [44] and ST-VQA [8]. The results show that our MM-GNN with newly designed aggregators effectively learns the representations of the scene texts and facilitates the performance of VQA tasks that require reading texts.

2. Related Work

Visual Question Answering Tasks. In recent years, numerous works have proposed diverse VQA tasks [39, 34, 4, 16, 48, 42, 53, 24, 23] for evaluating different types of core skills for answering visual questions. One line of datasets [39, 34, 4, 16], such as COCO-QA and VQA, studies questions about querying the visual information of an image. Relevant works [33, 14, 41, 1, 6, 35, 50] propose various attention mechanisms and multi-modal fusion techniques to better locate the image region for a given question to facilitate the answering procedure. Another line of works, such as CLEVR and GQA, introduces questions demanding complex and compositional spatial reasoning skills. Relevant works on these tasks introduce modular networks [2, 3, 20, 25, 22] and neural-symbolic model [43, 51] which can robustly generate answer by performing explicit multi-step reasoning on an image.

In this paper, we focus on a new type of questions recently proposed by the TextVQA [44] and ST-VQA [8]. Compared to other VQA tasks, these two tasks are unique in introducing questions about images that contain multi-modal contents, including visual objects and diverse scene texts. To solve these tasks, this paper focuses on how to formulate the multi-modal contents and obtain better representations of scene texts and objects.

Representation Learning in VQA. Some inspiring works have studied the representation of images to improve the performance of VQA tasks. The VQA models [33, 14, 41, 35, 50] in the early stage mainly use the VGG or ResNet feature pre-trained on the ImageNet to rep-
resent images. However, this type of grid-level feature is limited to perform object-level attention. Therefore, [1] proposes to represent one image as a list of detected object features. Besides, to solve complex compositional questions, [43, 51] propose some symbolic structural representations of the synthetic images (e.g., a scene graph extracted from an image) in CLEVR which allow a VQA system to perform explicit symbolic reasoning on them. More recently, [36, 32, 21] represent the natural image as a fully connected graph (can be viewed as an implicit scene graph where the relations between objects are not explicitly represented). This type of graph allows the model to predict dynamic edge weights to focus on a sub-graph related to the question and is widely used in natural images QA.

The above-mentioned methods all focus on the representation of visual objects, while this paper extends it to represent images with multi-modal contents. We represent one image as a graph composed of three sub-graphs to separately depict the entities in each modality and build the connections between entities in different modalities.

**Graph Neural Network.** Graph Neural Network (GNN) [40, 10, 29, 46, 49] is a powerful framework for representing graph-structured data. The GNN follows an aggregation scheme that controls how the representation vector of a node calculated by its neighboring nodes to capture specific patterns of a graph. Recently, numerous variants of GNN are proposed to capture different types of patterns of the graph in many tasks. For graph classification tasks, many works on text classification [40, 46, 11], and protein interface prediction [13] utilize the GNN to iteratively combine the information of the neighboring nodes to capture the structure information of the graph.

In addition, many interesting works [45, 36, 32, 21, 47] introduce GNN for grounding related task, such as referring expression [27] and visual question answering [54, 16, 23]. These works [45, 36, 32, 21, 47] propose GNN with language conditioned aggregator to dynamically locate a sub-graph of the scene for a given query (e.g., a referring expression or a question), then GNN updates the features of the nodes in the sub-graph to encode the relations among objects. The updated nodes have better features for latter grounding related tasks.

Similar to the previous GNNs [45, 36, 32, 21, 47] for grounding related tasks, we utilize the GNN to obtain better features. But this paper extends GNN from performing reasoning on a single-modal graph to a multi-modal graph. Besides, our proposed new aggregation schemes can explicitly capture different types of multi-modal contexts to update the representation of the nodes.

### 3. Method

In this section, we elaborate on the proposed multi-modal graph neural networks (MM-GNN) for answering visual questions that require reading. Given an image, which contains visual objects and scene texts, and a question, the goal is to generate the answer. Our model answers the question in three steps: (1) extract the multi-modal contents of an image and construct a three-layer graph, (2) perform multi-step message passing among different modalities to refine the representation of the nodes, and (3) predict the answer based on the graph representation of the image.

#### 3.1. Multi-Modal Graph Construction

As shown in Fig. 3, given an image, we first construct a multi-modal graph composed of three sub-graphs, i.e., visual graph, semantic graph and numeric graph for representing the information in three modalities. The visual graph $G_v$ is a fully connected graph, where each node $v_i \in V_v = \{v_i\}_{i=1}^N$ encodes the pure visual information of entities (i.e., objects and scene texts) and $N$ is the number of candidate objects generated by the extractor. The initial representation $v_i^{(0)}$ of $v_i$ is obtained by using an image feature extractor, e.g., Faster R-CNN [15] detector.

The semantic graph $G_s$ is also a fully connected graph, and each node $s_i \in V_s = \{s_i\}_{i=1}^M$ represents the semantic meaning of a scene text, e.g., “New York” is the name of a city, “Sunday” is one day in a week, and $M$ is the number of extracted tokens. Concretely, to obtain the semantic graph, we first use an Optical Character Recognition (OCR) model to extract word tokens in images. Then, the $i$-th token is embedded by a pre-trained word embedding model as the initial representation $s_i^{(0)}$ of node $s_i$.

Besides, for number-type strings, e.g., “2000”, they not only contain semantic meanings indicating string type, e.g., year (or dollars), but also numeric meanings which indicate the numeric relations among other number-type strings, e.g., “2000” is larger than “1900”. Thus, we construct a fully connected numeric graph $G_n$ to represent such information of number-type texts $x_i \in V_n = \{x_i\}_{i=1}^K$. We categorize common numeric texts into several types, e.g., number, time, etc. Then number-type texts are embedded into -1 to 1, denoted as $x_i^{(0)}$, with sigmoid function (for monotone number, like “12”) or cosine function (for period number, like “10:00”) according to their categories, where $K$ is the number of number-type texts. More details of the number encoder are in the Supp. Besides, the entire graph composed of three sub-graphs is overall fully connected, but only a specific part of nodes and edges is used in one aggregator.

#### 3.2. Aggregation Scheme

After constructing the graph and initializing the representation of each node, we propose three aggregators which guide the information flow between one sub-graph to another or itself to utilize the different types of context to refine the representation of the nodes, as shown in Fig. 3.
Visual-Semantic Aggregator. The first aggregator is the Visual-Semantic aggregator, goal of which is two-fold: 1) leverage the visual context to refine a semantic node (for solving questions like Q1 in Fig. 2) and 2) utilize the semantic context to refine a visual node, making the visual representation of texts’ physical carriers aware of the text (for solving questions like Q3 in Fig. 5). Here, we first illustrate the implementation of the first goal. For each node \( s_i \) in semantic graph \( G_s \), the aggregator updates the representation of \( s_i \) by first attending on relevant neighbour nodes in visual graph \( v_j \in \mathcal{N}_{s_i}^v = \{v_j|j=1, \text{then}\text{aggregating}\text{the} \text{information of attended nodes to update the representation of} \text{\( s_i \).} \text{Concretely, we first calculate the relevance score} \text{\( a_{v_j,s_i}^v \text{\( \)) between the node} \text{\( s_i \) and its neighboring node} v_j based on their visual representation and their location features} b_{s_i} \text{and} b_{v_j} \text{(i.e. the coordinates of bounding boxes) and the question feature} q \text{obtained by embedding the question words and going through an LSTM [19], formulated as,}

\[
a_{v_j,s_i}^v = f_s([s_i^{(0)}; f_s(b_{s_i})])^T (f_v([v_j^{(0)}; f_v(b_{v_j})]) \odot f_q(q)), \quad a_{v_j,s_i}^v = \frac{\exp(a_{v_j,s_i}^v)}{\sum_{s_j \in \mathcal{N}_{s_i}^v} \exp(a_{v_j,s_j}^v)},
\]

where \( f_s \), \( f_v \), \( f_b \) and \( f_q \) are the MLPs for encoding the semantic nodes, visual nodes, bounding boxes features, and question feature respectively, \( [:] \) indicates concatenating two vectors, and \( \odot \) is element-wise multiplication. Here, we also consider the question information in calculating the attention score, because we hope the model can aggregate the related nodes considering the information in the question. Then, we aggregate the information of attended nodes and append the aggregated features to \( s_i^{(0)} \) depicting the additional information of this node to obtain the updated semantic representation, formulated as, 

\[
s_i^{(1)} = [s_i^{(0)}; \sum_{v_j \in \mathcal{N}_{s_i}^v} a_{v_j,s_i}^v f_v(v_j^{(0)})],
\]

where \( s_i^{(1)} \) is the updated node representation at \( t=1 \) (shown in Fig. 3), and \( f_{v} \) is an MLP to encode the features of neighboring nodes.

Similar to the scheme of refining semantic nodes, we obtain the updated representation of nodes \( v_j^{(1)} \) in \( G_v \) by

\[
a_{v_j,s_i}^v = \frac{\exp(a_{v_j,s_i}^v)}{\sum_{s_j \in \mathcal{N}_{v_j}^s} \exp(a_{v_j,s_j}^v)},
\]

\[
v_j^{(1)} = [v_j^{(0)}; \sum_{s_i \in \mathcal{N}_{v_j}^s} a_{v_j,s_i}^v f_v(s_i^{(0)})],
\]

where \( f_v \) is an MLP to encode the \( s_j \), and \( \mathcal{N}_{v_j}^s \) indicates the neighboring nodes of \( v_j \) in the visual graph. Note that in all aggregators, the additional information is appended after original features; specifically, after Visual-Semantic aggregation, the dimensions of both semantic and visual features are multiplied by two.

Semantic-Semantic Aggregator. This aggregator then refines the representation of each semantic node by considering its semantic context (for solving questions like Q2 in Fig. 2). For each node \( s_i \), the aggregator finds the proper neighboring nodes in semantic graph \( \mathcal{N}_{s_i}^s = \{s_j|j \in \{1, \ldots, M\} \text{and} j \neq i\} \) by attention mechanism, then aggregating the information of attended nodes.

More specifically, the relevance score \( a_{s_j,s_i} \) of the node \( s_i \) and its neighboring node \( s_j \) is computed by their semantic representation and their location features \( b_{s_i} \) and \( b_{s_j} \) in images, formulated as,

\[
a_{s_j,s_i} = g_{s_i}([s_i^{(1)}; g_b(b_{s_i})])^T (g_{s_j}([s_j^{(1)}; g_b(b_{s_j})]) \odot g_q(q)), \quad a_{s_j,s_i} = \frac{\exp(a_{s_j,s_i})}{\sum_{s_j \in \mathcal{N}_{s_i}^s} \exp(a_{s_j,s_i})},
\]

where \( g_{s_i} \), \( g_{s_j} \), \( g_b \), and \( g_q \) are the MLPs for encoding the node features (the first two), bounding boxes features and
question features. Then, we aggregate the information of attended nodes, and append the aggregated features to \( s_i \) as,

\[
s_i^{(2)} = [s_i^{(1)}; \sum_{s_j \in N_s^x} a_{s_j,s_i} g_{s_j}(s_j^{(1)})],
\]

where \( s_i^{(2)} \) is the updated node representation at \( t=2 \), and \( g_{s_j} \) is an MLP to encode the features of neighboring nodes.

**Semantic-Numeric Aggregator.** The goal of this aggregator is to leverage the semantic context to refine the value nodes to depict more informative numeric relations between numbers (for solving questions like Q3 in Fig. 2). The mechanism of semantic-numeric aggregator is similar to the mechanism of achieving the first goal in Visual-Semantic aggregator. We first calculate the relevance score \( a_{s_j,x_i} \) between nodes \( s_j \) and \( x_i \), then aggregate the information of semantic nodes to numeric nodes, formulated as,

\[
x_i^{(3)} = [x_i^{(0)}; \sum_{s_j \in N_{x_i}^s} a_{s_j,x_i} h(s_j^{(2)})],
\]

where \( h \) is for encoding the semantic nodes and \( N_{x_i}^s = \{s_j\}_{j=1}^M \). Finally, we append the numeric nodes to their corresponding semantic nodes as the representation of OCR tokens, denoted as \( c = [c_1, ..., c_M] \). For OCR tokens which are not number-type, we concatenate a vector where the elements are all 0.

### 3.3. Answer Prediction

The answer prediction module takes the updated visual features \( v = [v_1, ..., v_N] \) and OCR features \( c = [c_1, ..., c_M] \) as inputs, and outputs the answer with copy mechanism [17]. Concretely, first, the size of output space is extended to the vocabulary size + OCR number, where some indexes in the output space indicate copying the corresponding OCR as the answer, as shown in Fig. 3. Then, we calculate the attention score on features of two modalities, and use attended features to generate the score of each answer, formulated as,

\[
y = f_a((f^{v}_{an}(v,q)^Tv; f^{c}_{an}(c,q)^Tc)),
\]

where \( f^{v}_{an} \) and \( f^{c}_{an} \) are Top-down attention networks in [1] and \( f_a \) is an MLP to output the scores on all candidate answers. Finally, we optimize the binary cross entropy loss to train the whole network. This allows us to handle cases that the answer is in both the pre-defined answer space and the OCR tokens without penalizing for predicting either one.

### 4. Experiments

#### 4.1. Experiments Setup

**Datasets.** We evaluate our model using the TextVQA dataset and Scene-Text VQA (ST-VQA) dataset.

For **TextVQA** dataset, it contains a total of 45,336 human-asked questions on 28,408 images from Open Image dataset [30]. Each question-answer pair comes along with a list of tokens extracted by Object Character Recognition (OCR) models, Rosetta [9]. These questions are evaluated by VQA accuracy metric [16].

For **ST-VQA** dataset, it is composed of 23,038 images, paired with 31,791 human-annotated questions. In the Weakly Contextualized task of ST-VQA, a dictionary of 30,000 words are provided for all questions in this task; and the Open Dictionary task is open-lexicon. These questions are evaluated by two metrics, Average Normalized Levenshtein Similarity (ANLS) [31] and accuracy.

**Implementation Details.** For experiments on **TextVQA** dataset, we use answers that appear at least twice in the training set as our vocabulary. Thus, the size of our output space is the sum of the vocabulary size and the OCR number, that is, \( 3997 + 50 \). For question features, we use GloVe [38], which is widely used in VQA models, to embed the words, then feed word embeddings to an LSTM [19] with self-attention [52] to generate the question embedding. For encoding OCR tokens, GloVe only can represent out-of-vocabulary (OOV) words as 0-vectors which are not suitable for initialization them, so we use fastText [26], which can represent OOV words as different vectors, to initialize OCR tokens. For image features, we use two kinds of pre-extracted visual features for each image provided by the TextVQA dataset, 1) 196 grid-based features obtained from pre-trained ResNet-152, and 2) 100 region-based features extracted from pre-trained Faster R-CNN model like [1].

Both two visual features are 2048-dimensional. Note that, the Faster R-CNN provides the visual features of both objects and the scene texts because the detector produces an excessive amount of bounding boxes, where some bounding boxes will bound the scene texts.

Bounding box coordinates of objects and OCR tokens are first normalized into the interval of [0, 1]. Then we concatenate its center point, lower-left corner and upper-right corner’s coordinates, width, height, area, and aspect ratio, into a 10-dimensional feature. We used AdaMax optimizer [28] for optimization. A learning rate of 1e-2 is applied on all parameters except the fc7 layer for finetuning, which are trained with 5e-3.

For experiments on **ST-VQA** dataset, due to no available OCR results are provided, we use TextSpotter [18] to extract scene text in images. For question and OCR token embedding, we use the same models as in TextVQA; and for image features, we only use Faster R-CNN features. Besides, we swap the prediction vocabulary to suit the change of the dataset. For Open Dictionary task, we collect answers which appear at least twice together with single-word answers which appear once in the training set as our vocabulary. For Weakly Contextualized task, given vocabulary of
size 30,000 is directly utilized. Besides, the source codes are implemented with PyTorch [37].

4.2. Results

Comparison with state-of-the-arts. Table 1 shows the comparison between our method and state-of-the-art approaches on validation and test set of TextVQA dataset. In the table, LoRRA (Pythia) is the baseline provided by TextVQA dataset [44]. BERT + MFH is the winner of CVPR 2019 TextVQA challenge, which is considered as the state-of-the-art, and its results are quoted from its challenge winner talk. LA+OCR UB refers to maximum accuracy achievable by models using Large Vocabulary of LoRRA and OCR results provided by TextVQA dataset [44]. LoRRA and BERT+MFH utilize advanced fusion techniques to attend to OCR tokens which are encoded by pre-trained FastText [26]. BERT+MFH additionally introduces the powerful question encoder BERT [12] into the answering model. Our approach outperforms the above methods which mainly rely on pre-trained word embedding, and achieves state-of-the-art results. Table 2 compares our method and state-of-the-art approaches on Weakly Contextualized and Open Dictionary tasks of Scene-Text VQA datasets, where VTA is the winner model of ICDAR 2019 Competition on ST-VQA, which extends the Bottom-Up VQA model [1] with BERT to encode the question and text. From the results, we can see that MM-GNN obtains an obvious improvement over baseline methods, e.g. SAN(CNN)+STR, and achieves comparable accuracies with VTA.

Effectiveness of Multi-Modal GNN. Our model’s advantage lies in the introduction of a multi-modal graph and a well-designed message passing strategy between different sub-graphs to capture different types of contexts. Thus, we propose several variants of our model, where each variant ablates some aggregators to show their indispensability.

- No-GNN: This variant directly uses the object and OCR token features extracted from pre-trained models to answer the questions without going through the multi-modal GNNs. Other modules (output, embeddings) are kept the same to MM-GNN.
- Vanilla GNN: This variant puts object and OCR token features in a single graph. It then performs an aggregation similar to Semantic-Semantic aggregator to update the representation of the nodes. Other modules are kept the same to MM-GNN.
- Combinations of VS, SS, and SN: These variants construct the multi-modal graph like MM-GNN, but only use one or two of the aggregators to update representations. VS, SS, and SN represent Visual-Semantic, Semantic-Semantic, and Semantic-Numeric aggregators respectively.

In addition, to better compare the results in detail, we categorize the questions in TextVQA into three types. The first type of question is Unanswerable, including questions that are unanswerable for given currently provided OCR tokens in TextVQA dataset. We obtain this type of question by checking whether the ground-truth answer absent from predefined answer vocabulary and provided OCR tokens.

Table 1. VQA accuracy (%) on the TextVQA dataset, comparison with baselines and state-of-the-art models. LA+OCR UB refers to maximum accuracy achievable by models using Large Vocabulary of LoRRA and OCR results provided by TextVQA dataset [44].

<table>
<thead>
<tr>
<th>Method</th>
<th>Val</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pythia</td>
<td>13.04</td>
<td>14.01</td>
</tr>
<tr>
<td>LoRRA (BAN)</td>
<td>18.41</td>
<td>-</td>
</tr>
<tr>
<td>LoRRA (Pythia)</td>
<td>26.56</td>
<td>27.63</td>
</tr>
<tr>
<td>BERT + MFH</td>
<td>28.96</td>
<td>-</td>
</tr>
<tr>
<td>MM-GNN (ours)</td>
<td>31.44</td>
<td>31.10</td>
</tr>
<tr>
<td>BERT + MFH (ensemble)</td>
<td>31.50</td>
<td>31.44</td>
</tr>
<tr>
<td>MM-GNN (ensemble) (ours)</td>
<td>32.92</td>
<td>32.46</td>
</tr>
<tr>
<td>LA+OCR UB</td>
<td>67.56</td>
<td>68.24</td>
</tr>
</tbody>
</table>

Table 2. Average Normalized Levenshtein Similarity (ANLS) and accuracies (%) of different methods on Weakly Contextualized and Open Dictionary tasks on the test set of ST-VQA dataset.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SAAA</td>
<td>0.085</td>
<td>6.36</td>
<td>0.085</td>
<td>6.36</td>
</tr>
<tr>
<td>SAAA+STR</td>
<td>0.096</td>
<td>7.41</td>
<td>0.096</td>
<td>7.41</td>
</tr>
<tr>
<td>SAN(LSTM)+STR</td>
<td>0.136</td>
<td>10.34</td>
<td>0.136</td>
<td>10.34</td>
</tr>
<tr>
<td>SAN(CNN)+STR</td>
<td>0.135</td>
<td>10.46</td>
<td>0.135</td>
<td>10.46</td>
</tr>
<tr>
<td>VTA [7]</td>
<td>0.279</td>
<td>17.77</td>
<td>0.282</td>
<td>18.13</td>
</tr>
<tr>
<td>MM-GNN (ours)</td>
<td>0.203</td>
<td>15.69</td>
<td>0.207</td>
<td>16.00</td>
</tr>
</tbody>
</table>

1Our source codes are available at http://vipl.ict.ac.cn/resources/codes.
TextVQA model can effectively help the answering procedure. Comparing the results of Vanilla GNN with MM-GNN series, we find that if message passing in GNN is not well-designed, directly applying GNN to TextVQA task is of little help. By comparing the results of SS, SN, and VS, we find that Visual-Semantic aggregator contributes most performance gain to OCR-related questions and overall accuracies. This demonstrates our idea that multi-modal contexts are effective in improving the quality of scene text representation.

However, we find that Numeric-Semantic aggregator contributes smaller than the other two aggregators, probably because the portion of questions querying the relations between the numbers, such as “what is the largest number in the image?”, is relatively small. Thus, it limits the space to show the effectiveness of this aggregator.

**Impact of different combining methods.** Choosing combination schemes controlling the fusion of a source node and the aggregated features of its neighboring nodes is one crucial part of Graph Neural Network design. Original MM-GNN is designed to gradually append additional information to each node to serve as hints to distinguish OCR tokens from each other and facilitate the answering model to locate the proper OCR token. Here, we replace our concatenation updater by several variants which are broadly used in other GNNs:

- **Sum:** this variant combines the features of source nodes and its neighboring nodes by sum operation, which is widely used in existing GNN works, such as [5].
- **Product:** this variant updates each node by computing the element-wise multiplication of the node feature and aggregated features of its neighboring nodes.
- **Concat + MLP:** this variant updates each node by concatenating the node feature and aggregated features of its neighboring nodes, then uses an MLP to encode the concatenated features, which is used in previous visual language-related methods [21].

<table>
<thead>
<tr>
<th>Method</th>
<th>Answerable</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vocab</td>
<td>OCR</td>
</tr>
<tr>
<td>No-GNN</td>
<td>28.88</td>
<td>35.38</td>
</tr>
<tr>
<td>Vanilla GNN</td>
<td>28.29</td>
<td>37.70</td>
</tr>
<tr>
<td>VS</td>
<td>27.54</td>
<td>41.38</td>
</tr>
<tr>
<td>SS</td>
<td><strong>29.75</strong></td>
<td>38.89</td>
</tr>
<tr>
<td>SN</td>
<td>25.67</td>
<td>40.30</td>
</tr>
<tr>
<td>VS + SS</td>
<td>28.81</td>
<td>42.16</td>
</tr>
<tr>
<td>VS + SN</td>
<td>28.61</td>
<td>41.30</td>
</tr>
<tr>
<td>SS + SN</td>
<td>25.69</td>
<td>41.99</td>
</tr>
<tr>
<td><strong>VS + SS + SN</strong> (ours)</td>
<td><strong>27.85</strong></td>
<td><strong>43.36</strong></td>
</tr>
</tbody>
</table>

Table 3. VQA accuracy (%) of VQA models with different kinds of Graph Neural Networks on validation set of the TextVQA dataset.

We evaluate their performances on the validation set of TextVQA dataset, and the performances are shown in Table 4. We can see that all three schemes harm the performances more or less. Empirically, this is because the information between nodes and their neighborhoods are compressed, gradually averaging out the differences between node features, thereby bewildering the answering module when it tries to locate the question-related OCR token. Note that all three above combination schemes have the superiority of not changing node feature size through iterations; while our concatenation scheme loosens this restriction to preserve more information in combination stage.

<table>
<thead>
<tr>
<th>Method</th>
<th>Answerable</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vocab</td>
<td>OCR</td>
</tr>
<tr>
<td>Sum</td>
<td>27.40</td>
<td>40.40</td>
</tr>
<tr>
<td>Product</td>
<td>27.89</td>
<td>32.18</td>
</tr>
<tr>
<td><strong>Concat + MLP</strong></td>
<td><strong>28.11</strong></td>
<td><strong>38.44</strong></td>
</tr>
<tr>
<td><strong>Concat</strong> (ours)</td>
<td><strong>27.85</strong></td>
<td><strong>43.36</strong></td>
</tr>
</tbody>
</table>

Table 4. VQA accuracy (%) of variants of MM-GNN with different combination schemes on validation set of TextVQA dataset.

We evaluate their performances on the validation set of TextVQA dataset, and the performances are shown in Table 4. We can see that all three schemes harm the performances more or less. Empirically, this is because the information between nodes and their neighborhoods are compressed, gradually averaging out the differences between node features, thereby bewildering the answering module when it tries to locate the question-related OCR token. Note that all three above combination schemes have the superiority of not changing node feature size through iterations; while our concatenation scheme loosens this restriction to preserve more information in combination stage.

**4.3. Qualitative Analysis**

To gain an intuition of the attention distribution in aggregators, we visualized them in Fig. 4. It shows that our model can produce very sharp attentions to do reasoning on graphs, and the attentions have good interpretability. In Q1, with the question querying about the player with a ball, OCR tokens are guided by attention module to incorporate more information related to the basketball; besides question hints, “WDOVER” naturally attends to the area of the player. In Q2, the OCR token “Panera” incorporates the position and semantic information from “BREAD” accord-
Q1: What is the bank called?
MM-GNN: transilvania
No-GNN: bt
Intuition: physical carrier linking (prominent signboard might be the name) and inter-OCR reasoning

Q2: What city does the white book 4th from the top say?
MM-GNN: Boston
No-GNN: New York
Intuition: question hints and visual context direction

Q3: What color is the text pie?
MM-GNN: black
No-GNN: answering does not require reading text in the image
Intuition: strengthened visual feature by texts in the image

Q4: What national park is mentioned on the license plate?
MM-GNN: yosemite
No-GNN: CANCALE
Intuition: inter-OCR reasoning

Q5: How long does the tape measure to?
MM-GNN: 90
No-GNN: 20
Intuition: reasoning among numeric-type texts

Figure 5. Visualization of the reasoning procedure of MM-GNN model. We only display the attention from the OCR token selected as the answer in the answering module. The predicted OCR tokens are in white boxes. In the Visual-Semantic Attention column, we show the attention from OCR tokens to the most attended two visual objects, which are in red bounding boxes. The Semantic-Semantic Attention column displays attention between the predicted OCR token to the most attended OCR tokens, which are in yellow bounding boxes. In the Semantic-Numeric Attention column, the attentions from the predicted OCR token to other OCR tokens are shown (if any) in cyan. Images most important for answering the question are framed in orange, and the thickness of bounding boxes is proportional to their attention weights. These satisfying visualizations demonstrate that our model learns to do step-by-step reasoning in an explainable way.

5. Conclusion

In this paper, we introduce a novel framework Multi-Modal Graph Neural Network (MM-GNN) for VQA with scene texts. The MM-GNN represents the image with multi-modal contents as a composition of three graphs, where each graph represents one modality. In addition, the designed multi-modal aggregators in MM-GNN utilize multi-modal contexts to obtain a finer representation of elements in the image, especially for unknown, rare or polysemous words. Experimentally, we show that our new image representation and message passing schemes greatly improve the performance of the VQA with scene texts and provide interpretable intermediate results.

Acknowledgements. This work is partially supported by Natural Science Foundation of China under contracts Nos. 61922080, U19B2036, 61772500, and CAS Frontier Science Key Research Project No. QYZDJ-SSWJSC009.
References


