Text-Image Transformer with Cross-Attention for Visual Question Answering

Mahdi Rezapour

1mashhad

November 29, 2023
Text-Image Transformer with Cross-Attention for Visual Question Answering

Mahdi Rezapour
Independent researcher
rezapour2088@yahoo.com
Abstract
Visual question answering (VQA) is a challenging task that requires multimodal reasoning and knowledge. In this study, we explored different methods of feature fusion for VQA, using pre-trained models to encode the text and image features, and then applying different attention mechanisms to fuse them. We evaluated our methods on the DAQUAR dataset. We used three metrics to measure the performance of our methods: WUPS, Acc, and F1. We found that concatenating raw text and image features performs slightly better than self-attention for VQA. We also found that using text as query and image as key and value performs worse than other methods of cross-attention or self-attention for VQA, because it might not capture the bidirectional interactions between the text and image modalities. We discussed possible explanations and examples for these findings.

Keywords: Visual question answering, Feature fusion, Attention mechanism, Multimodal reasoning
**Introduction**

Visual question answering (VQA) is a challenging task that requires a system to provide a natural language answer to a natural language question, given an image. VQA involves not only visual recognition [1], but also logical reasoning[2], external knowledge, and common sense. Both questions and answers are open-ended, which means that the system needs to handle a variety of questions, such as object detection, binary visual questions [3], collecting visual information recurrently [4], knowledge base reasoning, activity recognition, and so on.

In order to answer visual questions, the system needs to perform visual reasoning regarding the visual elements of the image and some general information that may not be explicitly present in the image. Figure 1 shows three images with corresponding questions and answers that illustrate the challenges and benefits of using transformer models for visual question answering. Based on Figure 1, some of the challenges associated with VQA are: The questions might be ambiguous, vague, or incomplete, requiring the model to infer the missing information or handle multiple possible interpretations. The questions may require complex reasoning skills, such as counting, comparing, spatial reasoning, or commonsense knowledge. The questions may span different domains and topics, requiring the model to have a broad and diverse knowledge base.

Some of challenges associated with Q and A based on Figure 1 are, for instance, the question “what is on the left side of the white oven on the floor and on right side of the blue armchair?” is challenging because it involves spatial reasoning and multiple references. The model needs to locate the white oven and the blue armchair in the image, and then identify the objects that are on their respective sides. The model also needs to handle the ambiguity of the word “side”, which could mean either adjacent or opposite, depending on the context.

The question “what is the white object on the table on the right side of the wall below the bookshelf?” is difficult to answer because it involves a complex hierarchical structure. The model needs to parse the question and understand the nested relations among different entities, such as table, wall, bookshelf, and object. The model also needs to deal with the vagueness of the word “object”, which could refer to any item on the table.

Or the question “how many objects are between the fire extinguisher and the white oven?” is hard to answer because it involves counting and comparison. The model needs to count the number of objects that are in between the two specified items, which may not be clearly defined or visible in the image. The model also needs to compare the sizes and distances of different objects and decide whether they are close enough or far enough to be considered as between.

These are just some examples of how VQA questions can be difficult for a model to answer correctly and confidently. There may be other factors that affect the performance of a model, such as noise, occlusion, illumination, or perspective in the image, or grammar, spelling, or punctuation errors in the question. Therefore, VQA models need to be robust and adaptable to various scenarios and challenges.
Moreover, visual questions may target various areas of an image, including underlying context and background details, which require a detailed understanding of the image and its relation to the question. In the past study, multimodal fusion transformers with Bidirectional Encoder Representations from Transformers (BERT) encoding was used to process multimodal data of video and text [5]. In another study, stacking multiple attention layers [6] and self-attention layers [7] were used for task of VQA.

It should be noted that there are various types of VQA, such as Multiple Choice (MC) VQA [8] and free-form open-ended (FFOE) VQA [9]. The FFOE VQA, for instance, is more challenging as answers are only available for training dataset and there is no candidate list for choosing answers.

VQA models need to effectively encode both the visual and textual information and fuse them in a meaningful way to generate accurate and relevant answers. The benefits of using transformers are, 1. they can learn powerful representations of both images and texts, without relying on hand-crafted features or pre-trained models. 2. They can handle different types of inputs and outputs, such as images, texts, regions, objects, attributes, or embeddings. 3. They can leverage large-scale pre-training on diverse and rich datasets, such as ImageNet or Wikipedia, to improve their generalization and robustness. 4. They can be easily adapted or fine-tuned to specific domains or tasks, such as medical VQA or visual dialog.

We evaluate our models on the Full Dataset Question Answering on Real World Images (DDAQUAR) dataset [10] incorporated real-world scenes. The dataset consists of approximately 12,500 question-answer pairs based on images. We perform data cleaning over questions and names of images. For instance, we normalize the questions by removing the image IDs present in them. We split the dataset into 9,974 for training and 2,494 for test datasets.

We create a dictionary that contains questions, answers, and image_id for each data point. Before feeding the data into our models, we use a collator function that collects and processes the questions (text) and images and returns the tokenized text with attention mask along with tokenized images. These are then fed into our multimodal fusion transformer model for question answering.

Studies conducted using VQA in different domain, such as medical VQA [11, 12]. Or in another past study, CNN and RNN were combined to map images and questions to a common feature space [13].

**Methodology**

As we considered BERT and ViT, the following paragraphs summarize their processes.

BERT stands for Bidirectional Encoder Representations from Transformers [14]. It is a language model that uses the Transformer architecture [15] to encode both left and right context of a word in a sentence.
BERT can be pre-trained on large-scale text corpora using two objectives: masked language modeling (MLM) and next sentence prediction (NSP). MLM randomly masks some words in the input and predicts them based on the rest of the words. NSP predicts whether two sentences are consecutive or not. BERT can be fine-tuned on various natural language understanding tasks, such as question answering, by adding a task-specific layer on top of the pre-trained model.

The bert-base-uncased model is a pretrained transformer model for natural language processing tasks. It has the following layers, an embedding layer that converts the input tokens into vectors of size 768. This layer also adds positional and segment embeddings to the token embeddings. 12 transformer blocks, each consisting of a multi-head self-attention layer and a feed-forward layer, with residual connections and layer normalization.

A pooling layer that takes the output of the first token ([CLS]) and applies a linear transformation and a tanh activation function. This layer produces a vector of size 768 that can be used for classification tasks. A dropout layer that randomly sets some elements of the input tensor to zero with a probability of 0.1. This layer helps to prevent overfitting and improve generalization. A classifier layer that takes the output of the pooling layer and applies a linear transformation to produce a vector of size equal to the number of labels. This layer can be used for sequence classification tasks.

ViT stands for Vision Transformer [15]. vit-base-patch16-224-in21k, pretrained on a large collection of images in a supervised fashion, namely ImageNet-21k at a resolution of 224x224 pixel. ViT divides an image into patches and treats them as tokens for the Transformer. ViT can be pre-trained on large-scale image datasets, using contrastive learning or supervised learning. ViT can be fine-tuned on various computer vision tasks, such as image classification, by adding a task-specific layer on top of the pre-trained model. For VQA, BERT and ViT can be combined to form a multimodal fusion model that can leverage both language and image features in a unified framework. ViT consists of the following layers:

Input layer: This layer takes an image as input and splits it into patches of size 16x16 pixels. Each patch is then flattened and linearly projected to a vector of dimension 768. A special token called [CLS] will be added to the beginning of the sequence to represent the whole image. Positional embeddings are also added to each patch vector to encode their spatial information. The output of this layer is a sequence of patch vectors with shape (N+1, 768), where N is the number of patches and 1 is for the [CLS] token.

Transformer encoder: This layer consists of 12 identical blocks, each containing a multi-head self-attention sublayer and a feed-forward sublayer, with residual connections and layer normalization. The self-attention sublayer allows each patch vector to attend to other patch vectors and the [CLS] token, forming a joint representation of the image and the question. The feed-forward sublayer consists of two linear layers with a ReLU activation in between. The output of this layer is a sequence of transformed patch vectors with shape (N+1, 768).

Output layer: This layer consists of a linear layer that takes the [CLS] token vector as input and predicts the class label for the image. The output of this layer is a vector of dimension C, where C is the number of classes.

In summary, we used multimodal fusion transformers with BERT encoding. We use BERT to encode both the question and the answer in our VQA system. We also use ViT to encode the image into a sequence of patches and learn global features from them. We combine BERT [14] and ViT [16] using various types of multimodal fusion transformers.
As items, F1 score and total accuracy, are too restrictive, penalizing synonym or similar words, ocean vs sea, as incorrect answer, we also considered Palmer similarity score (WUPS). WUPS take into account the semantic similarity between the predicted answer and ground truth [17]. WUPS is a metric that measures the similarity between the predicted answer and the ground truth answer based on the WordNet hierarchy. It assigns a score between 0 and 1 for each answer pair, where 1 means perfect match and 0 means no match. WUPS can be computed with different thresholds, such as 0.0 or 0.9, to control the strictness of the similarity measure.

Here we briefly summarized the process. We first, append question, answers and image_id. If there were more than a single answer, the answers were split and the first one was used. Collator was used to process questions and images, while returning tokenized text and futurized images and fed them into multimodel transformer model for questioning answering tasks.

We used the pre-trained models as a starting point and fine-tune them on a smaller dataset of visual questions and answers. This way, we can leverage the knowledge and skills that the pre-trained models have learned from large-scale text and image data, such as language understanding and image recognition, and adapt them to the specific task of answering questions about images. Using pre-trained models can also improve the performance and accuracy of the visual question answering system, as the pre-trained models can capture more information and context from the text and image inputs than randomly initialized models.

**Results**

We evaluated six models, and the results are presented in Table 1. Only two models completed the five epochs, while the rest were stopped early because their accuracy did not improve. We set early_stopping=True, which is stopping training if metric_for_best_model does not improve for patience evaluations. We set the boolean to indicates whether to use early stopping or not. The training stopped when the evaluation metric specified by metric_for_best_model did not improve for patience number of evaluations. We set the value of patience as 7. We set the WUPS score as the evaluation metric for the VQA model.

<table>
<thead>
<tr>
<th>ID</th>
<th>Wups</th>
<th>Acc</th>
<th>F1</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.31</td>
<td>0.27</td>
<td>0.069</td>
<td>Just concatenate of image and text and pass it through fusion layer</td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
<td>0.28</td>
<td>0.068</td>
<td>Just concatenate raw image and text with dropout</td>
</tr>
<tr>
<td>3</td>
<td>&lt;0.20*</td>
<td>&lt;0.20*</td>
<td>&lt;0.02*</td>
<td>Concatenate of raw image and text and pass them through sequential/linear/relu/relu</td>
</tr>
<tr>
<td>4</td>
<td>&lt;0.20*</td>
<td>&lt;0.20*</td>
<td>&lt;0.02*</td>
<td>Raw text as query, encoded_image[pooler_output] as key and value</td>
</tr>
<tr>
<td>5</td>
<td>&lt;0.20*</td>
<td>&lt;0.20*</td>
<td>&lt;0.02*</td>
<td>Raw text as query, fused_output as key and value</td>
</tr>
<tr>
<td>6</td>
<td>&lt;0.20*</td>
<td>&lt;0.20*</td>
<td>&lt;0.02*</td>
<td>Raw text as query, passed image_key and image_value through linear model</td>
</tr>
</tbody>
</table>

The model was not improved and stopped after patience

We set the number of steps between each evaluation as 100, meaning that evaluation would be done every 100 steps. The evaluation strategy was also set as steps, measuring evaluation at a fixed number of steps of 100 steps.
Various models were considered, which would be explained briefly in the next few paragraphs.

ID=2: For this method, we just concatenate the raw image and text, without passing that through a fusion/attention layer. Our better-performed method might avoid some of the problems or complexities that other methods may introduce, such as noise, distortion, mismatch, redundancy, or imbalance. This method may also preserve the original information and order of the text and image features, which may be important for answering the questions. However, this explanation is only a hypothesis and may not hold for all cases and datasets. Different methods of feature fusion and attention may have different advantages and disadvantages depending on the task, the data, and the model architecture.

ID=1: we did the similar method as the second approach, ID=2, but we also passed the concatenated data through attention layer.

ID=3: For this model, we concatenated text-image, self-attention, with linear transformation method. That model may make sense to be considered as we wanted to use the concatenated text and image features as the input and the output of the self-attention layer. This may allow the model to learn how the text and image features are related to each other and produce a new representation that captures this information. However, this method may not be optimal for visual question answering, because it may lose some information or structure from the original text and image features due to the concatenation operation.

ID=4: For the 4th model, we set the raw text as query, which may allow the model to learn how the raw text features are related to the image features, and produce a new representation that captures this information. The encoded_image['pooler_output'], as key and value, would be a tensor of shape (batch_size, hidden_size) that represents the last layer hidden-state of the first token of the sequence (classification token). However, this method may not be optimal for visual question answering, because it may lose some important information or context from the text features that could help answer the question.

ID=5: To fuse the features, we set the raw text features as the query and the fused text-image features as the key and value. This may allow the model to learn how the raw text features are related to the fused text-image features and produce a new representation that captures this information. This model was not able to get optimized and the model gets stopped due to early_stopping argument.

ID=6: For the fusion model, we set raw text self-attention with linearly transformed image key-value, which reflects the input sequence and the transformation type of the self-attention mechanism. This method may make sense to be considered as we want to use the raw text features as the query and the linearly transformed image features as the key and value.

We used a self-attention layer from PyTorch's nn.MultiheadAttention module to do fusion. This layer implements a scaled dot-product attention, which is a type of self-attention that computes the weights by taking the dot product of the query and key vectors, and then scaling them by the square root of the dimensionality. The query, key, and value vectors are obtained by applying linear transformations to the input vector.

The output vector is then computed by multiplying the weights with the value vectors. As discussed, for some models, we used the same input vector for the query, key, and value vectors. The process is similar to what is explained in the past study [15]. We first concatenated the encoded text and image features along dimension 1, column, resulting in a tensor of shape (batch_size, 1536). We then applied a linear transformation to this tensor to obtain the query, key, and value vectors of shape (batch_size, 1536) each. We then computed the scaled dot-product attention by taking the dot product of the query and key vectors, dividing it by the square root of 1536, and applying a softmax function to get the attention weights of shape (batch_size, batch_size). We then multiplied the attention weights with the value vectors to get the
output vector of shape (batch_size, 1536), which is the fused features of the text and image. We then passed this output vector to a classifier layer to produce the logits for the visual question answering task.

The equations could be written as [15]:

\[
\begin{align*}
\text{Query} &= W_Q \cdot \text{input} \\
\text{Key} &= W_K \cdot \text{input} \\
\text{Value} &= W_V \cdot \text{input} \\
\text{Attention} (Q, K, V) &= \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V
\end{align*}
\]

Where \( W_Q, W_K, W_V \) are learnable weight matrices of shape (1536,1536) and \( d_k \) is the dimensionality of the key vector, which is 1536 for our case. In all the above formulas, we are using linear transformation where we use various weights to transform the input vectors of query, key and value. Those weights metrics are learnable parameters, nn.MultiheadAttention of PyTorch, which would be updated during model optimization.

**Discussion**

In this study, we explored different methods of feature fusion for visual question answering (VQA), a task that requires multimodal reasoning and knowledge. We used pre-trained models to encode the text and image features, and then applied different attention mechanisms to fuse them. We evaluated our methods on the VQA v2 dataset. We used three metrics to measure the performance of our methods: WUPS, Acc, and F1. WUPS is a word-level semantic similarity metric that compares the predicted answer with the ground truth answer based on their word embeddings. Acc is the accuracy metric that measures the percentage of exact matches between the predicted answer and the ground truth answer. F1 is the harmonic mean of precision and recall, which measures the overlap between the predicted answer and the ground truth answer.

We found that concatenating raw text and image features performs slightly better than self-attention for VQA, but there is no definitive answer to why this is the case. One possible explanation is that concatenation preserves the original information and order of the text and image features, while self-attention may introduce some noise or distortion due to the attention weights and the matrix multiplication operations. Concatenation may also allow the model to learn more fine-grained and local interactions between the text and image features, while self-attention may focus more on the global and high-level relationships. However, this explanation is only a hypothesis and may not hold for all cases and datasets. Different methods of feature fusion may have different advantages and disadvantages depending on the task, the data, and the model architecture.

We also found that using text as query and image as key and value performs worse than other methods of cross-attention or self-attention for VQA. One possible explanation is that this method does not capture the bidirectional interactions between the text and image modalities. Using text as query and image as key and value means that the model only attends to the image features that are relevant to the question, but not vice versa. This may result in missing some important information or context from the image that could help answer the question.

For example, if the question is “What color is the sky?”, using text as query and image as key and value may only focus on the sky region in the image, but ignore other regions that could provide clues about the weather, time, or location. A better method of cross-attention might be to use both text and image as query and key and value, which means that the model attends to both modalities simultaneously and learns from
their mutual influence. This may result in a more comprehensive and accurate representation of the text-image pair that can generate better answers.

Using text as query and concatenated text and image as key and value may also be a suboptimal method for VQA, because it may create a mismatch between the query and the key-value pairs. The query is only based on the text modality, while the key-value pairs are based on both text and image modalities. This may result in a loss of information or coherence when computing the attention weights and the output features.

Using raw text as query, passed image key and image value, pass through linear model, as key and value may also be a suboptimal method of cross-attention for VQA, because it may create a mismatch or redundancy between the query and the key-value pairs. The query is based on the raw text modality, which may not be well-aligned with the image modality. The key and value are based on the linear transformation of the image features, which may not preserve the original information or structure of the image. This may result in a loss of information or coherence when computing the attention weights and the output features.

For future studies, different fusion layers with different pre-trained model combinations could be considered for performance evaluation. We also plan to explore other datasets and tasks that involve multimodal reasoning and knowledge.

References


