MMFT-BERT: Multimodal Fusion Transformer with BERT Encodings for Visual Question Answering

Aisha Urooj Khan **Amir Mazaheri** **Niels Da Vitoria Lobo** **Mubarak Shah**
Center for Research in Computer Vision, University of Central Florida

aishaurooj@gmail.com, amirmazaheri@knights.ucf.edu
shah@crcv.ucf.edu, niels@cs.ucf.edu

Abstract

We present MMFT-BERT (**M**ulti**M**odal **F**usion **T**ransformer with **BERT** encodings), to solve Visual Question Answering (VQA) ensuring individual and combined processing of multiple input modalities. Our approach benefits from processing multimodal data (video and text) adopting the BERT encodings individually and using a novel transformer-based fusion method to fuse them together. Our method decomposes the different sources of modalities, into different BERT instances with similar architectures, but variable weights. This achieves SOTA results on the TVQA dataset. Additionally, we provide TVQA-Visual, an isolated diagnostic subset of TVQA, which strictly requires the knowledge of visual (V) modality based on a human annotator's judgment. This set of questions helps us to study the model's behavior and the challenges TVQA poses to prevent the achievement of super human performance. Extensive experiments show the effectiveness and superiority of our method[1].

1 Introduction

In the real world, acquiring knowledge requires processing multiple information sources such as visual, sound, and natural language individually and collectively. As humans, we can capture experience from each of these sources (like an isolated sound); however, we acquire the maximum knowledge when exposed to all sources concurrently. Thus, it is crucial for an ideal Artificial Intelligence (AI) system to process modalities individually and jointly. One of the ways to understand and communicate with the world around us is by observing the environment and using language (dialogue) to interact with it (Lei et al., 2018). A smart system, therefore, should be able to process visual information to extract meaningful knowledge as well as be able to use that knowledge to tell us what is happening. The story is incomplete if we isolate the visual domain from language. Now that advancements in both computer vision and natural language processing are substantial, solving problems demanding multimodal understanding (their fusion) is the next step. Answering questions about what can be seen and heard lies somewhere along this direction of investigation. In research towards the pursuit of combining language and vision, visual features are extracted using pre-trained neural networks for visual perception (He et al., 2016; Ren et al., 2015), and word embeddings are obtained from pre-trained language models (Mikolov et al., 2013b,a; Pennington et al., 2014; Devlin et al., 2018) and these are merged to process multiple modalities for various tasks: visual question answering (VQA), visual reasoning, visual grounding.

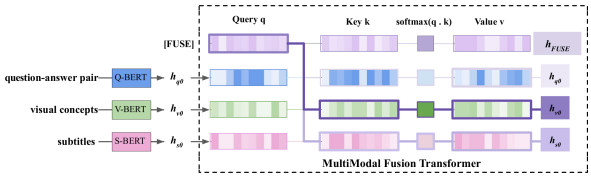

Figure 1: MultiModal Fusion Transformer (MMFT): We treat input modalities as a sequence. [FUSE] is a trainable vector; $h_{q_0 j}, h_{v_0 j}$, and $h_{s_0 j}$ are fixed-length features aggregated over question-answer (QA) pairs, visual concepts, and subtitles. Using a transformer encoder block, [FUSE] attends all source vectors and assigns weights based on the importance of each input source. Training end to end for VQA enables the MMFT module to learn to aggregate input sources w.r.t. the nature of the question. For illustration purposes, we show that for a single head, MMFT collects more knowledge from the visual source $h_{v_0 j}$ (green colored) than from the QA and subtitles. Best viewed in color.

TVQA (Lei et al., 2018), a video-based ques-

[1]Code will be available at https://github.com/aurooj/MMFT-BERT

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4648–4660
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

tion answering dataset, is challenging as it provides more realistic multimodal question answers (QA) compared to other existing datasets . To answer TVQA questions, the system needs an understanding of both visual cues and language. In contrast, some datasets are focused either visually: MovieFIB (Maharaj et al., 2017), Video Context QA (Zhu et al., 2017), TGIF-QA (Jang et al., 2017); or by language: MovieQA (Tapaswi et al., 2016); or based on synthetic environments: MarioQA (Mun et al., 2017) and PororoQA (Kim et al., 2017). We choose TVQA because of its challenges.

The introduction of transformers (Vaswani et al., 2017) has advanced research in visual question answering and shows promise in the field of language and vision in general. Here, we adopt the pretrained language-based transformer model, BERT (Devlin et al., 2018) to solve the VQA task. The human brain has vast capabilities and probably conducts processing concurrently. Like humans, an intelligent agent should also be able to process each input modality individually and collectively as needed. Our method starts with independent processing of modalities and the joint understanding happens at a later stage. Therefore, our method is one step forward toward better joint understanding of multiple modalities. We use separate BERT encoders to process each of the input modalities namely Q-BERT, V-BERT and S-BERT to process question (Q), video (V), and subtitles (S) respectively. Each BERT encoder takes an input source with question and candidate answer paired together. This is important because we want each encoder to answer the questions targeted at its individual source input. Thus, pairing up the question and candidate answers enables each stream to attend to the relevant knowledge pertinent to the question by using a multi-head attention mechanism between question words and a source modality. We then use a novel transformer based fusion mechanism to jointly attend to aggregated knowledge from each input source, learning to obtain a joint encoding. In a sense, our approach is using two levels of question-to-input attention: first, inside each BERT encoder to select only relevant input; and second, at the fusion level, in order to fuse all sources to answer the common question. We show in our experiments that using Q-BERT, a separate BERT encoder for question and answer is helpful.

Our contribution is three-fold:

First, we propose a novel multi-stream end-to-end trainable architecture which processes each input source separately followed by feature fusion over aggregated source features. Instead of combining input sources before input to BERT, we propose to process them individually and define an objective function to optimize multiple BERTs jointly. Our approach achieves state-of-the-art results on the video-based question answering task.

Second, we propose a novel MultiModal Fusion Transformer (MMFT) module, repurposing transformers for fusion among multiple modalities. To the best of our knowledge, we are the first to use transformers for fusion.

Third, we isolate a subset of visual questions, called TVQA-Visual (questions which require only visual information to answer them). Studying our method's behavior on this small subset illustrates the role each input stream is playing in improving the overall performance. We also present detailed analysis on this subset.

2 Related Work

Image-based Question Answering. Image-based VQA (Yu et al., 2015; Antol et al., 2015; Zhu et al., 2016; Jabri et al., 2016; Chao et al., 2018) has shown great progress recently. A key ingredient is attention (Ilievski et al., 2016; Chen et al., 2015; Yu et al., 2017a,b; Xu and Saenko, 2016; Anderson et al., 2018). Image based VQA can be divided based on the objectives such as generic VQA on real world images (Antol et al., 2015; Goyal et al., 2017), asking binary visual questions (Zhang et al., 2016) and reasoning based VQA collecting visual information recurrently (Kumar et al., 2016; Xiong et al., 2016; Weston et al., 2014; Sukhbaatar et al., 2015; Hudson and Manning, 2018) to answer the question both in synthetic (Johnson et al., 2016; Yang et al., 2018; Suhr et al., 2017) as well as real image datasets (Hudson and Manning, 2019).

Video-based Question Answering. Video-based QA is more challenging as it requires spatiotemporal reasoning to answer the question. (Lei et al., 2018) introduced a video-based QA dataset along with a two-stream model processing both video and subtitles to pick the correct answer among candidate answers. Some studies are: grounding of spatiotemporal features to answer questions (Lei et al., 2019); a video fill in the blank version of VQA (Mazaheri et al., 2017); other examples include (Kim et al., 2019b,a; Zadeh et al., 2019; Yi et al., 2019; Mazaheri and Shah, 2018).

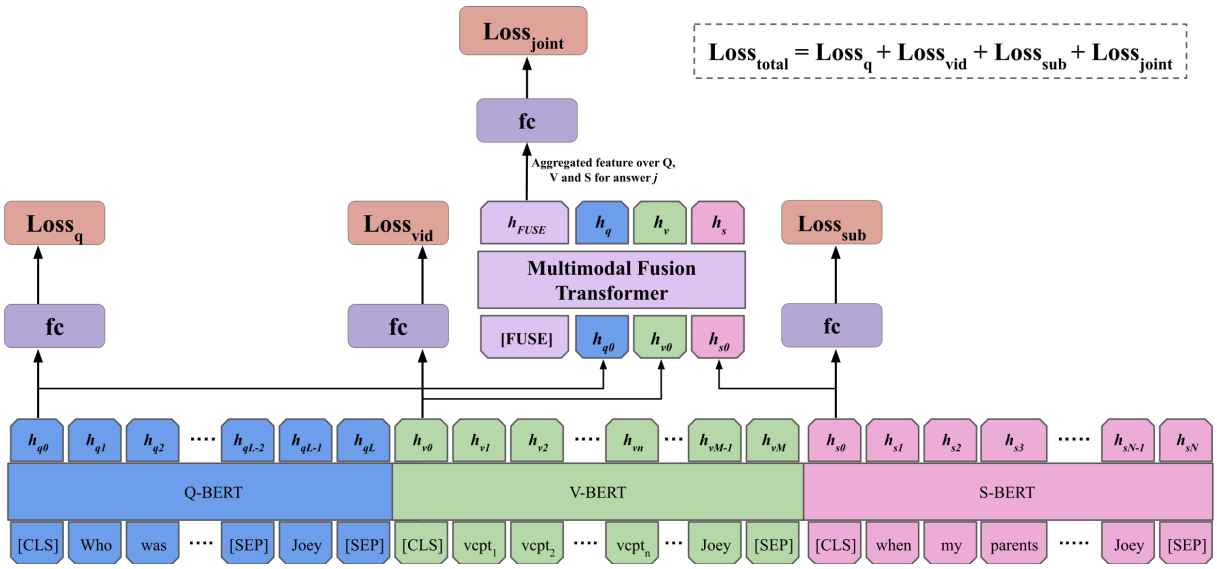

Figure 2: Overview of the proposed approach. Q-BERT, V-BERT and S-BERT represent text encoder, visual encoder and subtitles encoder respectively. If h_j=Q+A_j is j_{th} hypothesis, then Q-BERT takes h_j, V-BERT takes visual concepts V+h_j, and S-BERT takes subtitles S+h_j as inputs respectively. The aggregated features from each BERT are concatenated with [FUSE], a special trainable vector, to form a sequence and input into the MMFT module (see section 3.2.4 for details). Outputs from the MMFT module for each answer choice are concatenated together and are input into a linear classifier to obtain answer probabilities. We optimize individual BERTs along with optimizing the full model together. $Loss_{total}$ denotes our objective function used to train the proposed architecture. At inference time, we take features only from the MMFT module.

Representation Learning. BERT has demonstrated effective representation learning using self-supervised tasks such as masked language modeling and next sentence prediction tasks. The pre-trained model can then be finetuned for a variety of supervised tasks. QA is one such task. A single-stream approach takes visual input and text into a BERT-like transformer-based encoder; examples are: VisualBERT (Li et al., 2019b), VL-BERT (Su et al., 2019), Unicoder-VL (Li et al., 2019a) and B2T2 (Alberti et al., 2019). Two-stream approaches need an additional fusion step; ViLBERT (Lu et al., 2019) and LXMERT (Tan and Bansal, 2019) employ two modality-specific streams for images. We take this a step further by employing three streams. We use a separate BERT encoder for the question-answer pair. We are specifically targeting video QA and do not need any additional pre-training except using pre-trained BERT.

3 Approach

Our approach permits each stream to take care of the questions requiring only that input modality. As an embodiment of this idea, we introduce the MultiModal Fusion Transformer with BERT encodings (MMFT-BERT) to solve VQA in videos. See fig. 1 for the proposed MMFT module and fig. 2 for illustration of our full architecture.

3.1 Problem Formulation

In this work, we assume that each data sample is a tuple (V,T,S,Q,A,l) comprised of the following:V: input video; T: $T=[t_{start},t_{end}]$, i.e., start and end timestamps for answer localization in the video; S: subtitles for the input video; Q: question about the video and/or subtitles; A: set of C answer choices; l: label for the correct answer choice.

Given a question with both subtitles and video input, our goal is to pick the correct answer from C candidate answers. TVQA has 5 candidate answers for each question. Thus, it becomes a 5-way classification problem.

3.2 MultiModal Fusion Transformer with BERT encodings (MMFT-BERT)

3.2.1 Q-BERT:

Our text encoder named Q-BERT takes only QA pairs. The question is paired with each candidate answer A_j, where, $j=0,1,2,3,4; |A|=C$. BERT uses a special token $[CLS]$ to obtain an aggregated feature for the input sequence, and uses $[SEP]$ to deal with separate sentences. We, therefore, use the output corresponding to the $[CLS]$ token as the aggregated feature from Q-BERT and $[SEP]$ is used to treat the question and the answer choice as sep-

Method	Text Feat	Vis. Feat	Input							
			Q		Q+V		Q+S		Q+V+S	
			w/o ts	w/ ts	w/o ts	w/ ts	w/o ts	w/ ts	w/o ts	w/ ts
LSTM(Q)	Glove	-	42.74	42.74	-	-	-	-	-	-
MTL (Kim et al., 2019a)	Glove	cpt	-	-	-	43.45	-	64.36	-	66.22
Two-stream(Lei et al., 2018)	Glove	cpt	43.50	43.50	43.03	45.03	62.99	65.15	65.46	67.70
PAMN (Kim et al., 2019b)	Word2vec	cpt	-	-	-	-	-	-	66.77	-
Single BERT	BERT	cpt	-	-	-	48.95	-	-	-	72.20
STAGE (Lei et al., 2019)	BERT	reg	-	-	-	-	-	-	**68.56**	70.50
WACV20(Yang et al., 2020)	BERT	cpt	46.88	46.88	-	48.95	-	70.65	63.07	72.45
Ours-SF	BERT	cpt	**47.64**	**47.64**	**49.52**	50.65	69.92*	70.33	65.55	73.10
Ours-MMFT	BERT	cpt	**47.64**	**47.64**	49.32	**51.36**	**69.98***	**70.79**	66.10	**73.55**
Ours-MMFT(ensemble)	BERT	cpt	-	-	-	**53.08**	-	-	-	**74.97**

Table 1: Comparison of our method with baseline methods on TVQA validation set. STAGE uses regional features for detected objects in the video, all other models use visual concepts, ts= timestamp annotation, cpt=visual concepts, reg=regional features. Ours-SF represents proposed method with simple fusion, MMFT represents proposed multimodal fusion transformer, * indicates model trained with max_seq_len=512. The MMFT ensemble is 7x systems which use different training seeds.

arate sentences. The input I to the text encoder is formulated as:

$$I_{q_j} = [CLS] + Q + [SEP] + A_j, \quad (1)$$

where, + is the concatenation operator, $[CLS]$ and $[SEP]$ are special tokens, Q denotes the question, and A_j denotes the answer choice j, I_{q_j} is the input sequence which goes into Q-BERT and represents the combination of question and the j^{th} answer. We initiate an instance of the pre-trained BERT to encode each of the I_{q_j} sequences:

$$h_{q_0 j} = Q\text{-}BERT\,(I_{q_j})\,[0], \quad (2)$$

where [0] denotes the index position of the aggregated sequence representation for only textual input. Note that, the [0] position of the input sequence is $[CLS]$.

3.2.2 V-BERT:

We concatenate each QA pair with the video to input to our visual encoder V-BERT. V-BERT is responsible for taking care of the visual questions. Pairing question and candidate answer with visual concepts allows V-BERT to extract visual knowledge relevant to the question and paired answer choice. Input to our visual encoder is thus formulated as follows:

$$I_{v_j} = [CLS] + V + "." + Q + [SEP] + A_j, \quad (3)$$

where, V is the sequence of visual concepts[2], "." is used as a special input character, I_{v_j} is the input sequence which goes into our visual encoder.

$$h_{v_0 j} = V\text{-}BERT\,(I_{v_j})\,[0], \quad (4)$$

where, [0] denotes the index position of the aggregated sequence representation for visual input.

[2]Visual concepts is a list of detected object labels using FasterRCNN (Ren et al., 2015) pre-trained on Visual Genome dataset. We use visual concepts provided by (Lei et al., 2018).

3.2.3 S-BERT:

The S-BERT encoder applies attention between each QA pair and subtitles and results in an aggregated representation of subtitles and question for each answer choice. Similar to the visual encoder, we concatenate the QA pair with subtitles as well; and the input is:

$$I_{s_j} = [CLS] + S + "." + Q + [SEP] + A_j, \quad (5)$$

where, S is the subtitles input, I_{s_j} is the resulting input sequence which goes into the S-BERT encoder.

$$h_{s_0 j} = S\text{-}BERT\,(I_{s_j})\,[0]. \quad (6)$$

where, [0] denotes the index position of the aggregated sequence representation for subtitles input.

3.2.4 Fusion Methods

Let $I_i \in R^d$ denote the feature vector for i_{th} input modality with total n input modalities $I_1, I_2, ..., I_n$, d represents the input dimensionality. We discuss two possible fusion methods:

Simple Fusion: A simple fusion method is a Hadamard product between all input modalities and given as follows:

$$h_{FUSE} = I_1 \odot I_2 \odot ... \odot I_n, \quad (7)$$

where, h_{FUSE} is the resulting multimodal representation which goes into the classifier. Despite being extremely simple, this method is very effective in fusing multiple input modalities.

MultiModal Fusion Tranformer (MMFT): The MMFT module is illustrated in fig. 1. We treat I_i as a fixed d-dimensional feature aggregated over input for modality i. Inspired by BERT(Devlin et al., 2018), we treat aggregated input features from multiple modalities as a sequence of features

by concatenating them together. We concatenate a special trainable vector $[FUSE]$[3] as the first feature vector of this sequence. The final hidden state output corresponding to this feature vector is used as the aggregated sequence representation over input from multiple modalities denoted as h_{FUSE}.

$$h_{FUSE} = MMFT(I_1 + I_2 + ... + I_n)[0], \quad (8)$$

where, $+$ is the concatenation operator, [0] indicates the index position of the aggregated sequence representation over all input modalities.

In our case, we have three input types: QA pair, visual concepts and subtitles. For inputs $i = \{1,2,3\}$ and answer index $j = \{0,1,2,3,4\}$, the input to our MMFT module is $I_1 = h_{q_0 j}$, $I_2 = h_{v_0 j}$, and $I_3 = h_{s_0 j}$ and the output is h_{FUSE} denoting hidden output corresponding to the [FUSE] vector. Here, $h_{q_0 j}$, $h_{v_0 j}$, and $h_{s_0 j}$ are the aggregated outputs we obtain from Q-BERT, V-BERT and S-BERT respectively.

3.2.5 Joint Classifier

Assuming a hypothesis for each tuple (V,T,S,Q,A_j), where $A_j \in A; j = 0,..,4$ denotes five answer choices, our proposed Transformer Fusion module outputs $h_{FUSE\,j} \in R^d$. We concatenate the aggregated feature representation for all the answers together and send this to a joint classifier to produce 5 answer scores, as follows:

$$h_{final} = h_{FUSE_0} + h_{FUSE_1} + ... + h_{FUSE_4}, \quad (9)$$

$$scores_{joint} = classifier_{joint}(h_{final}), \quad (10)$$

where, $h_{final} \in R^{C \cdot d}$ and $scores_{joint} \in R^C$, C denotes number of classes.

3.3 Objective Function

Along with joint optimization, each of the Q-BERT, V-BERT and S-BERT are optimized with a single layer classifier using a dedicated loss function for each of them. Our objective function is thus composed of four loss terms: one each to optimize each of the input encoders Q-BERT, V-BERT and S-BERT, and a joint loss term over classification using the combined feature vector. The formulation of the final objective function is as follows:

$$L_{total} = L_q + L_{vid} + L_{sub} + L_{joint}, \quad (11)$$

where, L_q, L_{vid}, L_{sub}, and L_{joint} denote loss functions for question-only, video, subtitles, and joint loss respectively; all loss terms are computed using softmax cross-entropy loss function using label l. The model is trained end-to-end using L_{total}.

[3] $[FUSE]$ is initialized as a d-dimensional zero vector.

Input	Model	Acc (%)
Q+V	MTL (Kim et al., 2019a)	44.42
	Two-stream (Lei et al., 2018)	45.44
	Ours - MMFT	**51.83**
Q+V+S	MTL (Kim et al., 2019a)	67.05
	Two-stream (Lei et al., 2018)	68.48
	STAGE (Lei et al., 2019)	70.23
	WACV20 (Yang et al., 2020)	72.71
	Ours - MMFT model	**72.89**

Table 2: Performance comparison of different models on TVQA testset-public with timestamp annotations. All models use visual concepts except STAGE. We do not report numbers for other comparisons (Q+S and w/o ts) because only limited attempts are allowed to the test server for evaluation.

Inp.	Method	Question family (Accuracy%)						
		what	who	where	why	how	others	all
Q+V	Two-stream	47.70	34.60	47.86	45.92	42.44	39.10	45.03
	WACV20	51.31	41.14	52.86	48.45	46.24	36.86	48.95
	Ours-SF	52.76	42.52	52.36	51.42	46.75	41.61	50.65
	Ours-MMFT	**52.97**	**43.58**	**54.00**	**53.00**	**46.97**	**44.16**	**51.36**
Q+V+S	Two-stream	66.05	67.99	61.46	71.53	78.77	74.09	67.70
	Ours-SF	71.52	72.10	68.93	**76.99**	**82.25**	**83.2**	73.10
	Ours-MMFT	**72.22**	**72.39**	**69.89**	76.92	81.74	82.48	**73.55**

Table 3: Performance comparison for each question family. All models are trained with localized input (w/ ts).

4 Dataset

In TVQA, each question (Q) has 5 answer choices. It consists of 152K QA pairs with 21.8K video clips. Each question-answer pair has been provided with the localized video V to answer the question Q, i.e., start and end timestamps are annotated. Subtitles S have also been provided for each video clip. See supplementary work for a few examples.

4.1 TVQA-Visual

To study the behavior of state-of-the-art models on questions where only visual information is required to answer the question correctly, we selected 236 such visual questions. Due to imperfections in the object detection labels, only approximately 41% of these questions have the adequate visual input available. We, therefore, refer to TVQA-Visual in two settings: **TVQA-Visual (full):**– full set of 236 questions. A human annotator looked into the video carefully to ensure that the raw video is sufficient to answer the question without using subtitles. **TVQA-Visual (clean):** This is the subset of 96 questions where the relevant input was available, yet the models perform poorly. For this subset, we rely on a human annotator's judgement who verified that either the directly related visual concept or the concepts hinting toward the correct answer are present in the list of detected visual concepts. For instance, if the correct answer is "kitchen", ei-

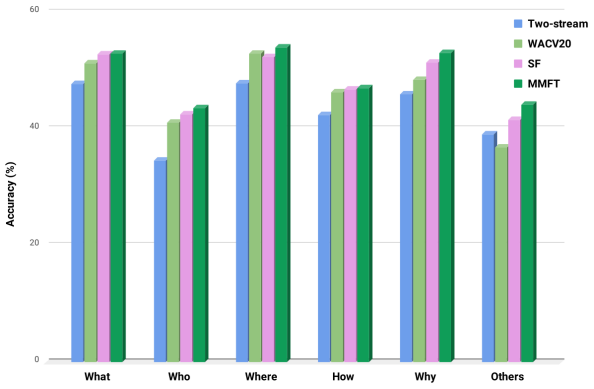

Figure 3: Accuracy comparison with respect to the question family between Two-stream (Lei et al., 2018), WACV20 (Yang et al., 2020) and our method on the validation set of TVQA. Models were trained on Q+V. MMFT outperforms on all question types for Q+V.

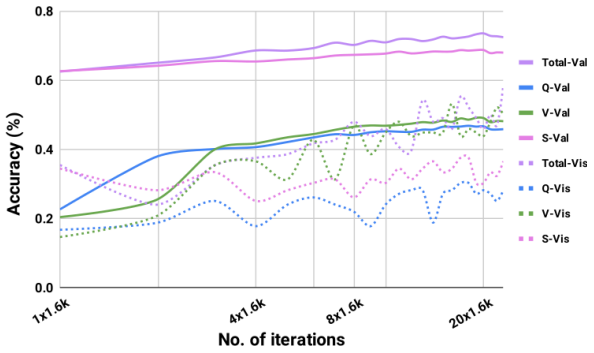

Figure 4: Testing accuracy curves on TVQA-Vis. (clean) and TVQA val set for different BERT streams during training. Solid lines: validation accuracy, dotted lines: visual set accuracy. Although S-BERT is significantly above V-BERT for full validation set, for visual set, we can see that V-BERT is well above Q-BERT and S-BERT. This shows that each BERT contributes to the questions it is responsible for. Numbers are log-scaled.

ther "kitchen" or related concepts (e.g. "stove", "plate", "glass", etcetera) should be present in the list of visual concepts. Thus, this easier subset is termed as TVQA-Visual (clean). TVQA-visual, although small, is a diagnostic video dataset for systematic evaluation of computational models on spatio-temporal question answering tasks and will help in looking for ways to make the V-stream contribution more effective. See supplementary material for the distribution of visual questions based on reasons for failure. If a model is correctly answering TVQA-visual questions which are not "clean" (the relevant concepts are missing from the visual input), that is because of statistical bias in the data.

5 Experiments and Results

5.1 Baselines

LSTM(Q): LSTM(Q) is a BiLSTM model to encode question and answer choices. The output from LSTM for question and each answer choice is concatenated and is input to a 5-way classifier to output 5 answer probability scores.

Method	TVQA-Vis. (clean)	TVQA Vis. (full)
Two-stream	35.42	29.49
WACV20	42.71	**40.00**
SF	42.71	34.37
MMFT	**46.88**	39.57

Table 4: Performance comparison on TVQA-Visual questions for clean set and full set. Numbers reported for only Q+V (w/ts) model. Numbers are reported as percentage.

MTL: MTL (Kim et al., 2019a) uses two auxiliary tasks with the VQA task: temporal alignment and modality alignment.

Two-stream: (Lei et al., 2018) uses two separate streams for attention-based context matching each input modality with question and candidate answers. A BiLSTM with max pooling over time is used to aggregate the resulting sequence.

BERT: A single pre-trained BERT instance is fine-tuned on QA pair along with visual concepts and subtitles all together (Q+V+S).

STAGE: (Lei et al., 2019) uses moment localization and object grounding along with QA pair and subtitles. STAGE uses BERT features to encode text and spatio-temporal features for video.

WACV20: (Yang et al., 2020) concatenates subtitles and visual concepts with QA pairs and input to BERT along with late fusion for Q+V and Q+S.

5.2 MMFT-BERT

For video representation, we use detected attribute object pairs as visual features provided by (Lei et al., 2018). We follow (Lei et al., 2018) and only unique attribute-object pairs are kept. Q-BERT, V-BERT and S-BERT are initialized with $BERT_{base}$ pre-trained on lower-cased English text with masked language modeling task. The MMFT module uses single transformer encoder layer (L=1) with multi-head attention. We use 12 heads (H=12) for multi-head attention in the MMFT module for our best model. We initialize the MMFT module with random weights. A d-dimensional hidden feature output corresponding to [CLS] token is used as an aggregated source feature from each BERT. We concatenate these aggregated features for each candidate answer together to acquire a feature of size $5 \cdot d$. A 5-way classifier is then used to optimize each of Q-BERT, V-BERT and S-BERT independently. For joint optimization of the full model, we treat the encoders' output as a sequence of features with the order $[[FUSE], h_{q_0 j}, h_{v_0 j}, h_{s_0 j}]$ and input this into the MMFT module (*[FUSE]* is a

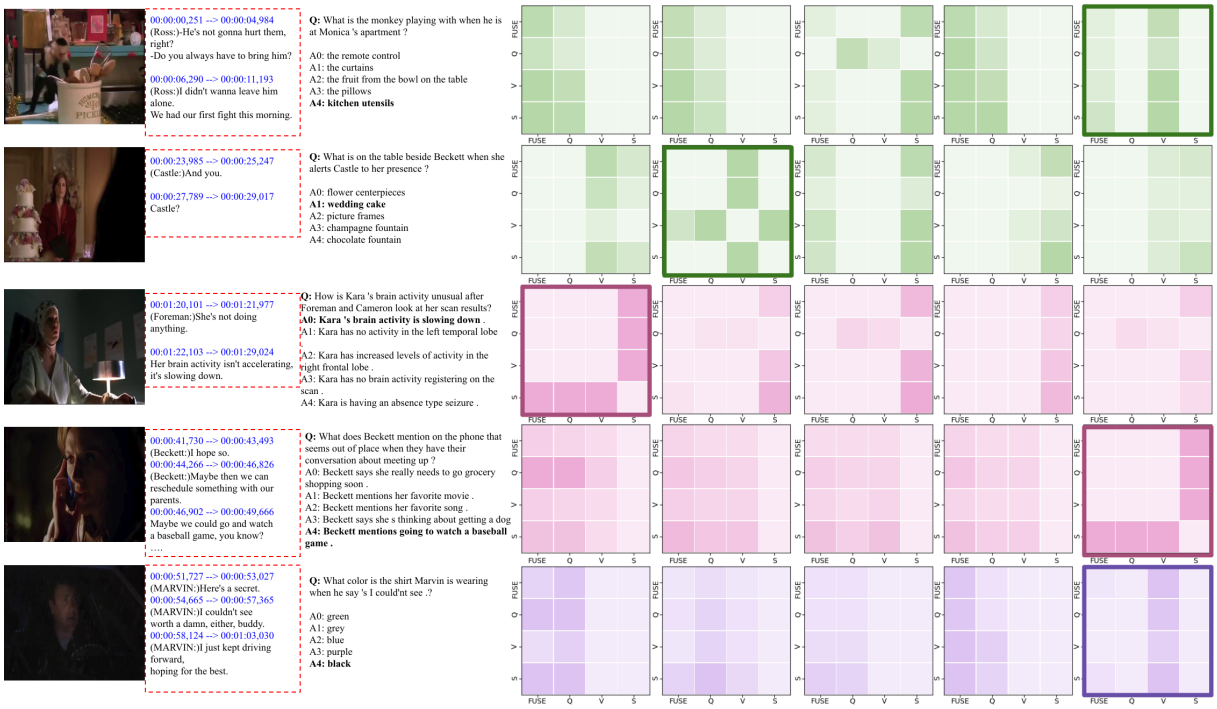

Figure 5: Visualization of multi-head attention (averaged over all heads) between different source features: Q, V and S for our best model. MMFT takes a sequence: [FUSE, Q, V, S] and uses multi-head attention for multimodal fusion. [FUSE] is the aggregated feature over Q, V, and S; column 1: representative image frame, column 2: localized subtitles, column 3: question with candidate answers (correct answer with corresponding attention map is in bold text and box respectively), columns 4-8 show attention for A0, A1, A2, A3, and A4 respectively. Top 2 rows show attention weights for visual questions, next 2 rows are subtitles-based questions. Last row, depends on both subtitles and visual information. See sec. 5.3.1 and supplementary work for details and insights.

trainable d-dimensional vector parameter). Output corresponding to *[FUSE]* token is treated as an accumulated representation $h_{FUSE\,j}$ over all input modalities for answer j. We concatenate $h_{FUSE\,j}$ for each answer choice to obtain h_{final} for the joint classification. We learn four linear layers, one on top of each of the three input encoders and the MMFT encoder respectively. Thus, each linear layer takes a $(5 \cdot d)$-dimensional input and produces 5 prediction scores.

Training Details. The entire architecture was implemented using Pytorch (Paszke et al., 2019) framework. All the reported results were obtained using the Adam optimizer (Kingma and Ba, 2014) with a minibatch size of 8 and a learning rate of 2e-5. Weight decay is set to 1e-5. All the experiments were performed under CUDA acceleration with two NVIDIA Turing (24GB of memory) GPUs. In all experiments, the recommended train / validation / test split was strictly observed. We use the 4th last layer from each BERT encoder for aggregated source feature extraction. The training time varies based on the input configuration. It takes ∼4 hrs to train our model with Q+V and ∼8-9 hrs to train on the full model for a single epoch. All models were trained for 10 epochs. Our method achieves its best accuracy often within 5 epochs.

5.3 Results

All results here use the following hyperparameters: input sequence length max_seq_len=256, # heads H=12, # encoder layers L=1 for the MMFT module, and pre-trained $BERT_{base}$ weights for Q-BERT, V-BERT and S-BERT unless specified explicitly.

With timestamp annotations (w/ ts). Columns with "w/ ts" in table 1 show results for input with timestamp localization. We get consistently better results when using localized visual concepts and subtitles. We get 1.7% and 0.65% improvement over WACV20 (Yang et al., 2020) with simple fusion for Q+V and Q+V+S inputs respectively. When using the MMFT for fusion, our method achieves SOTA performance with all three input settings: Q+V (↑ 2.41%), Q+S (↑ 0.14) and Q+V+S (↑ 1.1%) (see table 1). Our fusion approach contributes to improved performance and gives best results for localized input. See table 3 and fig. 3 for results w.r.t. question family.

Without timestamp annotations (w/o ts). We

	Model	Acc.(%)
1	Single BERT	72.20
2	Ours Simple Fusion, Single Loss	71.82
3	Ours Simple Fusion, FO	73.10
4	MMFT w/ BERT Encoder freezed	57.94
5	Ours Simple Fusion, FO, +Img	71.82
6	MMFT-BERT L=2, H=12	72.61
7	MMFT-BERT L=2, H=12 w/ skip	72.62
8	MMFT-BERT L=1, H=1	72.66
9	MMFT-BERT L=1, H=12	**73.55**

Table 5: Ablations over the design choices for the proposed architecture. L = no. of encoder layers in MMFT module, H = no. of heads in MMFT module, +Img = Resnet101 features pooled over video frames. Rows 3-9 are trained with our full objective (FO). All models are trained for Q+V+S with timestamp annotations.

also train our model on full length visual features and subtitles. Our method with simple fusion and MMFT on Q+V input outperforms Two-stream (Lei et al., 2018) by absolute 6.49% and 5.59% with simple fusion and MMFT respectively. We truncate the input sequence if it exceeds max_seq_len. Subtitles without timestamps are very long sequences (49% of subtitles are longer than length 256), hence QA pair might be truncated. Thus, we rearrange our input without timestamps as follows: "$Q\ [SEP]\ A_j\ .\ V$" and "$Q\ [SEP]\ A_j\ .\ S$" for V-BERT and S-BERT respectively. Models with Q+S input are trained with max_seq_len=512 and Q+V+S models are trained with max_seq_len=256 due to GPU memory constraints. For Q+S and Q+V+S, we observe 69.92% and 65.55% with simple fusion, using MMFT produces 69.98% and 66.10% val. accuracy respectively.

Results on test set. TVQA test-public set does not provide answer labels and requires submission of the model's predictions to the evaluation server. Only limited attempts are permitted. The server's evaluation results are shown in table 2. MMFT improves results by (↑ 6.39%) on Q+V. For Q+V+S, WACV20 reported 73.57% accuracy with a different input arrangement than MMFT. When compared with the model with the same input, MMFT performs slightly better (↑ 0.17%). Due to limited chances for submission to the test server for evaluation, the reported accuracy for Q+V+S is from one of our earlier models, not from our best model.

5.3.1 Model Analysis

Performance analysis on TVQA-Visual. To study the models, we evaluate Two-stream (Lei et al., 2018), WACV20 (Yang et al., 2020) and our method on both TVQA-Visual (full) and TVQA-Visual (clean). See table 4 for full results.

TVQA-Visual (full): Our method outperforms Two-stream (Lei et al., 2018) by 10.08% but drops by 0.43% compared to WACV20 (Yang et al., 2020). TVQA-Visual (full) has approximately 59% of the questions with missing visual concept or require extra visual knowledge. All three models including ours were trained on visual concepts. Inadequate input, therefore, makes it difficult for the models to attend the missing information.

TVQA-Visual (clean): We observe (↑ 11.46%) and (↑ 4.17%) improvement for clean set compared to Two-stream and WACV20. TVQA-Visual (clean) has relevant visual concepts or related concepts to the answer present in the input. Yet, it is challenging for existing methods (including ours) to perform well. Although our model observes significant improvement (↑4-11%) over baselines for this experiment, the take away message is that it is not enough. This subset of TVQA, therefore, serves as a good diagnostic benchmark to study the progress of exploiting visual features for multimodal QA tasks. Fig. 4 visualizes the test performance of each stream on TVQA-Visual clean during training our Q+V model on TVQA.

Performance analysis w.r.t multimodal attention. We study the behavior of the MMFT module for aggregating multimodal source inputs (Q, V, and S), we take our best model trained on all three sources, and evaluate it on questions which need knowledge about either the visual world, dialogue or both. We then visualize the average attention score map over all heads inside MMFT module (H=12) for each candidate answer, see fig. 5. Top 2 rows show attention scores computed among all 3 input sources and the [FUSE] vector for visual questions. Since, [FUSE] is the aggregated output over all input modalities. For instance, visual part should contribute more if the question is about the visual world. We can see the attention map for the correct answer has high attention scores between V and [FUSE] vector. The incorrect answers attend to the wrong sources (either Q or S). Similar is the behavior for rows 3-5, where the question is about subtitles, and the correct answer gives most weight to the subtitles compared to the incorrect answers. Heatmaps for incorrect answers are either focused more on a wrong single input source or the combination of them.

Positional Encodings for V-BERT. Positional encoding is done internally in BERT. When finetuned, for V-BERT, the positional encoding has no effect.

This has been verified by training our Q+V model with simple fusion (Ours-SF), where the input to V-BERT is a shuffled sequence of objects; no drastic difference was observed (shuffled: 50.32% vs. not shuffled: 50.65%).

5.3.2 Ablations

All ablations were done with Q+V+S input. See table 5 for complete results.

Simple fusion vs. MMFT Though using simple fusion for combining multimodal inputs is very effective and already outperforms all of the baselines, it lacks the basic functionality of explainability. Using MMFT instead, not only gives us an improvement (↑0.71% for Q+V and ↑0.39% for Q+V+S) over simple fusion, but is also more explainable.

Single loss vs. multiple losses. A simple design choice could be to use just joint loss instead of multiple loss terms. However, through our experiments, we find that using single joint loss term hurts the performance (71.82%). Optimizing each BERT along with optimizing the full model jointly gives us best results (73.10%) even without using MMFT.

Single head vs. multi-head MMFT. In an attempt to know if simplicity (single head) has an advantage over using multi-head attention, we trained MMFT-BERT with H=1. Using single head attention for fusion consistently performed lower than using multiple heads (72.66% vs. 73.55%) (we set H=12). Our hypothesis is that since pretrained BERTs have 12 heads, attention within each source BERT was local (d_{model}/H). Using single head attention over features which were attended in a multi-head fashion may be hurting the features coming out of each modality encoder. Thus, it makes more sense to keep the attention local inside MMFT if the input encoders use local attention to attend to input sequences.

Single layer fusion vs. stacked fusion. Another design parameter is encoder layers (L) in MMFT. We trained our full model with three settings: a) single encoder layer L=1, b) stacked encoder layer L=2, and c) stacked encoder with skip connection. a) gives best results (73.55%), whereas both b) and c) fusion hurts (72.61% and 72.62%). Note that all variants of our models are slightly better in performance than our baseline methods.

Resnet features vs. visual concepts. To study if incorporating additional visual context is advantageous, we experimented with Resnet101 features for visual information. We used Resnet101 features pooled over time along with visual concepts. We used question-words-to-region attention for aggregating visual features; adding this aggregated visual feature to Ours-SF hurts the performance (71.82%); using object labels was consistently more useful than visual features in various other experimental settings.

6 Conclusion

Our method for VQA uses multiple BERT encodings to process each input type separately with a novel fusion mechanism to merge them together. We repurpose transformers for using attention between different input sources and aggregating the information relevant to the question being asked. Our method outperforms state-of-the-art methods by an absolute ∼2.41% on Q+V and ∼1.1% on Q+V+S on TVQA validation set. Our proposed fusion lays the groundwork to rethink transformers for fusion of multimodal data in the feature dimension.

Acknowledgments

We thank the reviewers for their helpful feedback. This research is supported by the Army Research Office under Grant Number W911NF-19-1-0356. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. 2019. Fusion of detected objects in text for visual question answering. *arXiv preprint arXiv:1908.05054*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Wei-Lun Chao, Hexiang Hu, and F. Sha. 2018. Being negative but constructively: Lessons learnt from creating better visual question answering datasets. In *NAACL-HLT*.

Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. 2015. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. *International Conference on Learning Representations (ICLR)*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ilija Ilievski, Shuicheng Yan, and Jiashi Feng. 2016. A focused dynamic attention model for visual question answering. *arXiv preprint arXiv:1604.01485*.

A. Jabri, Armand Joulin, and L. V. D. Maaten. 2016. Revisiting visual question answering baselines. In *ECCV*.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2758–2766.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2016. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890.

Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D Yoo. 2019a. Gaining extra supervision via multi-task learning for multi-modal video question answering. *arXiv preprint arXiv:1905.13540*.

Junyeong Kim, Minuk Ma, Kyungsu Kim, Sungjin Kim, and Chang D Yoo. 2019b. Progressive attention memory network for movie story question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8337–8346.

Kyung-Min Kim, Min-Oh Heo, Seong-Ho Choi, and Byoung-Tak Zhang. 2017. Deepstory: Video story qa by deep embedded memory networks. *arXiv preprint arXiv:1707.00836*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. 2018. Tvqa: Localized, compositional video question answering. In *EMNLP*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2019. Tvqa+: Spatio-temporal grounding for video question answering. In *Tech Report, arXiv*.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2019a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019b. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. 2017. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6884–6893.

Amir Mazaheri and Mubarak Shah. 2018. Visual text correction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 155–171.

Amir Mazaheri, Dong Zhang, and Mubarak Shah. 2017. Video fill in the blank using lr/rl lstms with spatial-temporal attentions. In *The IEEE International Conference on Computer Vision (ICCV)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. 2017. Marioqa: Answering questions by watching gameplay videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2867–2875.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406.

Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer.

Guangyu Robert Yang, Igor Ganichev, Xiao-Jing Wang, Jonathon Shlens, and David Sussillo. 2018. A dataset and architecture for visual reasoning with a working memory. In *European Conference on Computer Vision*, pages 729–745. Springer.

Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. 2020. Bert representations for video question answering. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. 2019. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*.

Dongfei Yu, Jianlong Fu, Tao Mei, and Yong Rui. 2017a. Multi-level attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4709–4717.

Licheng Yu, Eunbyung Park, Alexander C. Berg, and Tamara L. Berg. 2015. Visual madlibs: Fill in the blank description generation and question answering. In *The IEEE International Conference on Computer Vision (ICCV)*.

Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017b. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 1821–1830.

Amir Zadeh, Michael Chan, Paul Pu Liang, Edmund Tong, and Louis-Philippe Morency. 2019. Social-iq: A question answering benchmark for artificial social intelligence. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Linchao Zhu, Zhongwen Xu, Yi Yang, and Alexander G Hauptmann. 2017. Uncovering the temporal context for video question answering. *International Journal of Computer Vision*, 124(3):409–421.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004.

A Supplementary Material

A.1 Improved comprehension of submitted paper

Certain parts of the submitted paper will be more clear to the reader if s/he is familiar with the concepts explained in (Vaswani et al., 2017) and (Devlin et al., 2018). For instance, the attention mechanism illustrated in the submitted paper's figure 1 needs understanding of transformers (Vaswani et al., 2017).

A.2 TVQA-Visual

See figure 7 for some statistics about TVQA-Visual set. Almost 59% of the questions have not enough input available (plot shows results for 200 questions, rest of the 35 questions are "who" questions and need character recognition). We will make this list of questions available to the community for further research.

A.3 Experiments and Results

Evaluation Metric. Multiple choice question answering accuracy is used to evaluate each model in this work.

Further discussion about results without timestamp annotations. For experiments without timestamp annotations, for Q+V and Q+S, the only competitor whose results are available is Two-stream (Lei et al., 2018); in these categories, MMFT is more than 6-7% better than the Two-stream, where, for Q+S, we train MMFT with a sequence length of 512. For Q+V+S, MMFT achieves 66.10% with max_seq_len=256. STAGE reports 2.46% higher accuracy. We had a GPU memory limitation and could only train our model with input size of 256. Had we had access to at least 4 GPUs (24GB of memory), we would have been able to train our full model with input size of 512, which would have presumably given us a similar boost we witnessed for Q+S without timestamps (Q+S is ∼3% better than Q+V+S). Therefore, we believe our model would perform better when provided with increased input length.

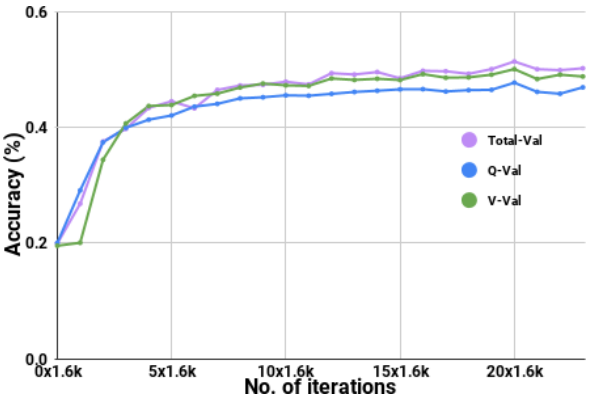

Figure 6: Validation accuracy curves for Q-BERT, V-BERT and full model when trained on Q+V input. Although Q-BERT performs lower than V-BERT as expected, it helps when Q-BERT is kept as a separate stream. During initial training, Q-BERT trains quickly than V-BERT. After first epoch, V-BERT starts outperforming Q-BERT as the model learns to leverage visual stream to answer the questions.

Performance analysis on multimodal questions. For true multimodal questions which cannot be answered without looking at both video and subtitles, the aggregated feature should rely on both modalities. The last row in figure 5 of the submitted paper is an attempt to study such questions. However, we observed, that many of these type of such questions which apparently require both modalities, can in practice be answered by just one of them. Although the question in last row is intended towards both video and dialogue (subtitles), the actual nature of the question is visual. We don't need to know what someone is saying to observe how they are dressed. To the best of our knowledge, no such constraints were imposed while collecting the original TVQA dataset. For instance, a true multimodal question about a specific appearance would be asked if the person appears multiple times with varying appearance in a video. Referring to dialogue in that case to localize the visual input is a true multimodal question. For example, in row 5, the question is " What color is the shirt Marvin is wearing when he say's I could'nt see.?", with the corresponding subtitles, MMFT chooses to ignore subtitles yet giving the correct answer. For the last row in figure 5, examine the attention maps to see how MMFT gives more attention to V source than subtitles S for the correct answer (which is in the last column).

Fusion Techniques: We also tried several other fusion methods including: a) gated fusion where each

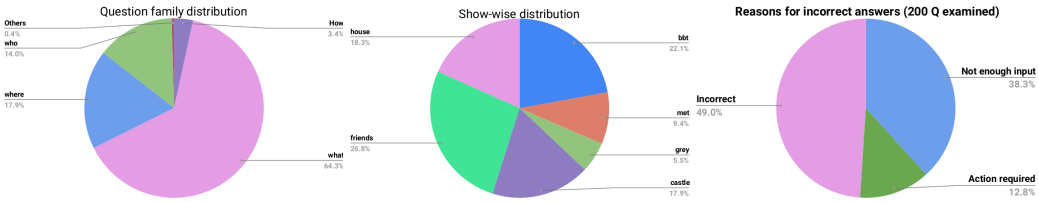

Figure 7: Few statistics for TVQA-Visual (full) set. Left) distribution of questions w.r.t question family, Center) distribution of questions w.r.t TV show, Right) Distribution of questions w.r.t reason of failure.

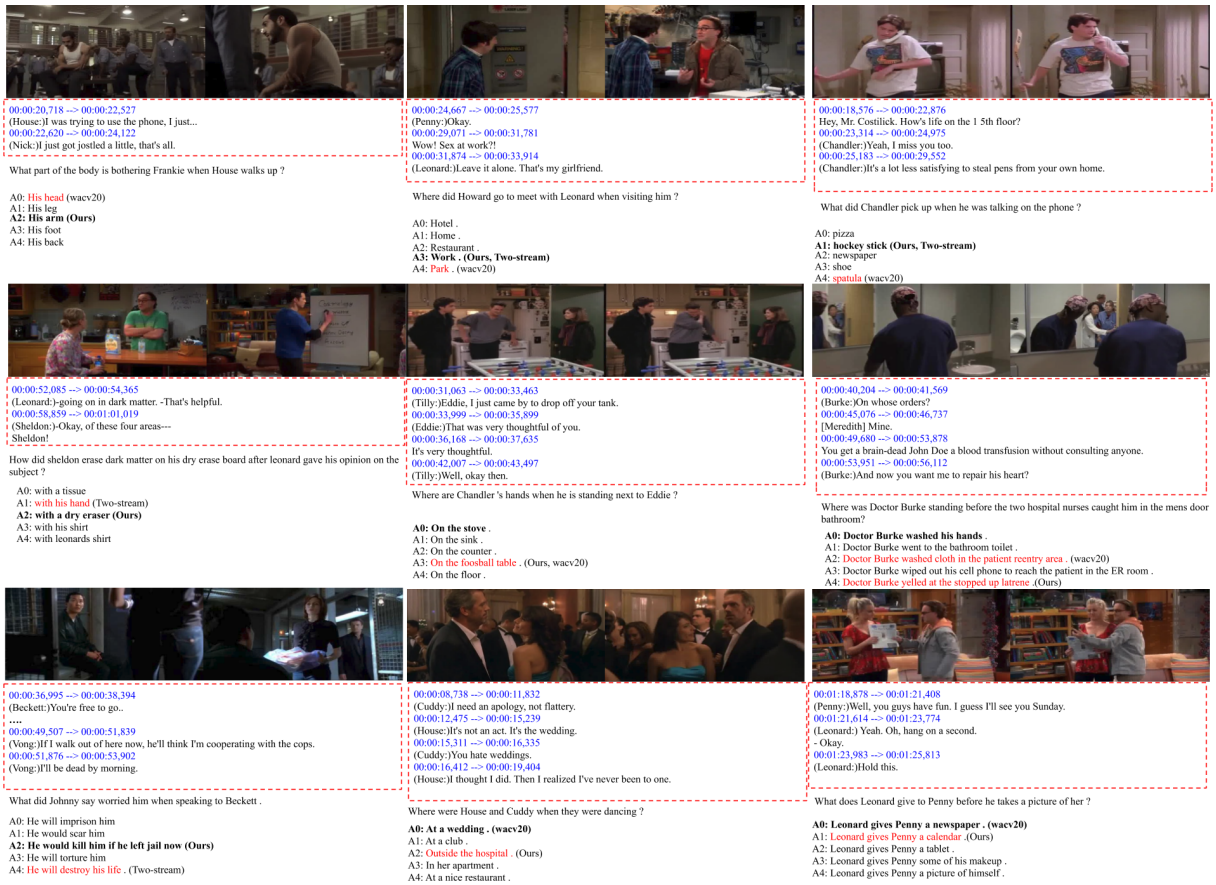

Figure 8: Qualitative results from validation set. Success and failure cases on visual and multimodal questions. Bold text shows correct answer, prediction of each model is in parenthesis. Incorrect prediction is in red font.

source vector is gated w.r.t. every other source vectors before fusing them together. We merge the resulting gated source features with i) concatenation followed by a linear layer, ii) taking the product of the gated source vectors, iii) concatenation of the gated fusion feature and the simple fusion feature. All of them result in suboptimal performance than our simple fusion method with a performance drop of 1-2%.

A.3.1 Qualitative Results

Some of the qualitative results are shown in figure 8 including both success and failure cases of our method and the baselines for Q+V+S input.