HomeworkP - ClipFusions: Cross-Modal Transformers for Video Question Answering

Giuseppina Iannotti, 1938436
iannotti.1938436@studenti.uniroma1.it
MSc in Artificial Intelligence and Robotics

Maria Emilia Russo, 1966203
russo.1966203@studenti.uniroma1.it
MSc in Artificial Intelligence and Robotics

Abstract

Video Question Answering is a complex task requiring models to analyze dynamic visual content and align it with natural language queries, incorporating temporal reasoning and multimodal understanding. In this work, we develop cross-modal transformer models based on pre-trained BERT embeddings for text and video encoders for visual features. Our models include masked self-attention and multimodal fusion to align and reason effectively between the modalities. The design is informed by several state-of-the-art models, including ClipBERT[9] and VIOLET[2]. We evaluate the models on real-world datasets to demonstrate their ability to tackle the intricacies involved with VideoQA.

1 Introduction

Video Question Answering (VideoQA) is a challenging and rapidly evolving task at the intersection of natural language processing (NLP) and computer vision (CV). Unlike traditional question-answering tasks that focus solely on textual data, VideoQA requires models to analyze dynamic visual content alongside natural language questions. This introduces a temporal dimension, as models must process sequences of video frames to understand events, interactions, and object relationships over time. The task demands not only an understanding of static visual scenes but also the ability to interpret changes and relationships as they unfold in a video.

At its core, VideoQA involves answering natural language questions based on video content. For instance, a question like ”*Who scored the goal?*” requires the model to identify relevant frames, recognize objects (e.g., players and the ball), track movements, and infer the action of scoring. This interplay of linguistic comprehension and visual reasoning makes VideoQA a uniquely multidisciplinary challenge.

The complexity of this task arises from its reliance on interconnected tasks such as visual understanding, temporal reasoning, and question parsing. To succeed, models must excel in areas like object detection, motion tracking, semantic understanding, and multimodal reasoning. These challenges highlight the need for robust architectures capable of integrating and reasoning across both visual and textual domains.

In this project, we address these challenges by developing and analyzing cross-modal transformer models tailored for VideoQA. Our focus is on answering multiple-choice questions based on video clips. Specifically, our approach involves:

- Leveraging pre-trained models to extract rich initial representations for both text and video. BERT embeddings capture semantic nuances in text, while pre-trained video encoders provide robust visual features.
- Implementing advanced transformer-based architectures to fuse multimodal information and enable effective cross-modal reasoning.
- Exploring alignment strategies between linguistic and visual features to ensure the model captures contextually relevant information.

Our goal is to improve the effectiveness of VideoQA systems and deepen understanding of multimodal fusion techniques for handling complex reasoning tasks.

2 Related Works

Understanding the state of the art is essential for shaping this project, as it helps us building on existing methods and addressing gaps in previous works. We explore several key techniques in the field, including:

- **Welcome Self-Attention :** *Beyond rnns: Positional self-attention with co-attention for video question answering*[10] is the first work of replacing RNNs with self-attention for the task of VideoQA. It proposes Positional Self-Attention with Coattention (PSAC), in which Positional Self-Attention is used to calculate the response at each position by attending to all positions within the same sequence, and then add representations of absolute positions. Furthermore, in addition to attending to the video features relevant to the given questions (i.e., video attention), they utilize the co-attention mechanism by simultaneously modeling "what words to listen to" (question attention).

- **Uni & Multimodal Streams :** Several approaches have been proposed for Video Question Answering (VideoQA) that leverage multi-modal information, integrating both visual and language features. All these methods focus on fusing visual and language information for VideoQA, often using BERT-based models for text processing and combining information from separate streams (video, subtitles, questions) to enhance multi-modal understanding.
  The key difference lies in how they process and fuse this information, with *ClipBERT*[9] focusing on end-to-end learning and introducing sparse sampling of video clips per training step, while others (*Bert Representations for Video Question Answering*[11] and *MMFT-BERT*[6]) use pre-extracted features and multiple BERT streams.

- **Latest Works:** Recent advancements in Video-QA have been driven by state-of-the-art models like **MERLOT**[14] and **VIOLET**[2], which aim to capture temporal dynamics and cross-modal interactions between visual and textual information through innovative pre-training strategies. In particular,

  - **MERLOT**[14] : It is a self-supervised model that learns common-sense reasoning from videos by pairing video frames with transcripts and focusing on temporal dynamics. It uses a ResNet-50 image encoder and a joint vision-language Transformer initialized with RoBERTa weights. Its pre-training objectives include contrastive frame-transcript matching, masked language modeling (MLM), and temporal reordering, helping it develop strong multimodal reasoning.
  - **VIOLET**[2] : This end-to-end model employs a video transformer to capture temporal dynamics in video inputs. It introduces three pre-training tasks: Masked Language Modeling (MLM), Visual-Text Matching (VT), and the novel Masked Visual-token Modeling (MVM), which involves recovering original video frame patches from masked tokens. Moreover, it also introduces two masking strategies: Blockwise Masking (BM), which forces the model to use visual reasoning by masking blocks of video patches, and Attended Masking (AM), which prioritizes important elements in video and text based on attention weights from a Cross-modal Transformer.

3 Datasets, Benchmarks and Evaluation

There are mainly two types of tasks in VideoQA[15]:

- **Multiple-Choice QA** : For multi-choice QA, the models are presented with several candidate answers for each question and are required to pick the correct one

- **Open-Ended QA** : For open-ended QA, the problem can be classification (the most popular), generation (word-by-word) and regression (for counting) depending on the specific datasets.

According to the data modality invoked in the questions and answers, VideoQA can be classified into

- **Normal VideoQA** : Only visual resources are invoked to understand the question and to derive the correct answer

- **Multi-Modal VideoQA** : Differently from normal VideoQA, MM VideoQA often involves other resources aside from visual contents, such as subtitles/transcripts and text plots of movies

- **Knowledge VideoQA** : It demands external knowledge distillation from explicit knowledge bases or commonsense reasoning

Moreover, according to the type of question (or the challenges posted in the questions), VideoQA can be further classified into

- **Factoid VideoQA** : Factoid questions directly ask about the visual fact, such as the location (where is), objects/attributes (who/what (color) is), and invokes little relations to understand the questions and infer the correct answers.
- **Inference VideoQA** : In contrast, inference VideoQA aims to explore the logic and knowledge reasoning ability in dynamic scenarios. It emphasizes temporal (before/after) and causal (why/How/what if) relationships that feature temporal dynamics

Our work focuses on the multiple-choice question answering (MCQA) task. To this end, we have selected two widely used dataset, **TGIF-Action** and **MSRVTT-MC**, to train and evaluate our models. These datasets are recognized for their relevance in video-based question answering tasks, providing diverse challenges in terms of understanding actions, frames, and multimodal content.

Evaluation Accuracy is the predominant metric used in state-of-the-art (SOTA) research for evaluating model performance in the MCQA domain. This metric provides an interpretable measure of how well a model selects the correct answers among the given options.

Existing Tools: For our work, we mainly looked at the code provided by Huggin Face[1] for the creation of the Multiple Choice Dataset, handling the textual aspects of our data. Additionaly, the design of our text encoder was inspired by [2][2] and [9][3], two highly influential works in this context.

4 Data Pre-Processing

Data preprocessing is a fundamental step in preparing raw data for analysis, ensuring that it is clean, structured, and ready to be used. This process addresses issues such as inconsistent formatting, missing values, irrelevant information, and noisy data, all of which can affect the quality and reliability of analytical results. It involves standardizing datasets by renaming and reordering columns to create a consistent structure, modifying identifiers for uniformity, and cleaning textual data to improve its usability. Text preprocessing includes operations such as removing special characters, punctuation, and stopwords, normalizing text through lemmatization, and tokenizing it. By addressing these issues, preprocessing ensures that both structured and unstructured data are in the best possible shape for analysis, setting the stage for meaningful and accurate insights.

Paired Dataset Building on these principles, we create a *Paired Dataset* function that prepares both text and video data for our multimodal task. Its objective is to pair text and video embeddings with corresponding labels.
We provide two implementations of it :

- **MSRVTT-MC Matchy Dataset**: It processes video files directly and explicitly separating question and answer tokens and managing multiple-choice formatting using attention masks.
- **TGIF Paired Dataset** : It focuses on pairing text embeddings and video embeddings for GIFs, handling external data sources by downloading and verifying GIF alignment before embedding extraction

While **TGIF Paired Dataset** emphasizes dataset preparation with alignment checks, **Matchy-Dataset** is designed to handle question-answer pairs, with a process of token separation and formatting in a multiple-choice context.

5 Method

In this work, we present two cross-modal transformers, each developed with different architectural designs and inspired by [9] and [2].

5.1 CLIP for Video Embeddings

Due to limited computational power and memory constraints, we opted to use the pre-trained CLIP

[1] https://huggingface.co/docs/transformers/tasks/multiple_choice
[2] https://github.com/jayleicn/ClipBERT
[3] https://github.com/tsujuifu/pytorch_violet

model to extract video embeddings. CLIP, pre-trained on a wide range of image-text pairs, is particularly effective at capturing rich visual features that can be used for our task.

Moreover, in order to handle the MSRVTT-MC dataset, we selected frames at regular intervals to ensure a representative sampling of the video while managing computational efficiency. This approach reduces the number of frames processed without sacrificing important visual information. The same issue did not arise for the TGIF dataset due to the short duration of its videos, which is inherent to the nature of GIFs. These videos were instead padded to the maximum number of frames and subsequently passed to the Cross-Modal Transformer.

5.2 Text Encoder

To effectively process textual information, two specialized text encoders were created, one for each dataset considered.

5.2.1 MSRVTT Text Encoder

This encoder is built using a BERT-based architecture, developed to process and encode textual inputs such as questions and answers into meaningful embeddings.

It leverages a **pre-trained BERT model**, specifically utilizing its embedding and encoder layers, to generate these representations. The encoder initializes its weights using best practices like Xavier initialization for linear layers and normal distribution initialization for embeddings, ensuring effective parameter tuning.

In its forward pass, the encoder takes as input batches of tokenized questions and answers, along with optional attention masks, token type IDs, and position IDs. It flattens and prepares these inputs for processing, ensuring compatibility with BERT's architecture. For each input, the encoder first generates initial embeddings using the BERT embedding layer, capturing the positional, token, and segment information of the input.

Next, the Bert Encoder layer refines these embeddings by applying a **multi-head self-attention mechanism** and feed-forward layers, conditioned on the **attention masks**. This process outputs contextualized embeddings, where each token's representation is enriched by its relationships with other tokens in the input. The same process is applied independently to both questions and answers.

Finally, the encoder outputs the processed embeddings for the questions and answers. These embeddings serve as inputs to the Cross-Modal Transformer, which leverages them alongside the visual features to better capture and understand the complex relationships between the textual data and the video content.

5.2.2 TGIF Text Encoder

This encoder integrates pre-trained BERT embeddings with attention mechanisms that can represent the rich semantic meaning in textual inputs. Therefore, **BERT** acts as the backbone in this model, providing contextually robust embeddings that capture semantic meaning and syntactic flow within the text input. These therefore provide a really strong foundation whereby further refinements can effectively be built on top of it. On top of the BERT embeddings, masked multi-head attentions were applied. These layers allow the model to capture complicated patterns of inter-relationships between tokens through which the model can identify contextually valuable words or terms. Each multi-head self-attention mechanism enables the models to attend to multiple disparate parts of a given input in parallel. This helps further emphasize the right tokens, or most relevant, according to the attention mechanism, therefore enabling better feature extraction from text concerning downstream tasks about nuanced textual information.

The **Masked Attention Mechanism** is designed to focus on specific parts of the input, depending on the task, by ignoring irrelevant or inaccessible tokens. This it does by incorporating an attention mask that defines which tokens the model can attend to during computation. Meanwhile, the mask is instrumental in multi-head attention for the restriction of the scope of attention: tokens that are masked are practically ”hidden” for the attention mechanism, and the model considers only the relevant parts of the input.

The TGIF Text Encoder is adapted for multiple-choice question-answering tasks. Given a question, it will output embeddings for the text with each of the possible answers. These embeddings capture the subtle interactions between the question and its possible answers. The resultant representations are semantically rich, contextually aligned, and prepared for integration with visual features in cross-modal tasks. This design allows for a guarantee of consistency and effectiveness in handling complex textual inputs with associated visual data.

5.3 Cross-Modal Transformer

The Cross-Modal Transformer (CMT) is a model designed to integrate and align textual and visual modalities for a range of downstream tasks, such as video-based question answering. This model builds on the foundation of two key components: text embeddings derived from the previously specified text encoders and video embeddings extracted using a pre-trained video model. By combining these modalities, the CMT captures both semantic and contextual relationships across the textual and visual domains, enabling effective multi-modal reasoning.

5.3.1 MSRVTT-MC CMT

The architecture begins by processing three types of inputs: questions, answers, and video embeddings. The questions and answers are textual embeddings, which represent the semantic content of the text, while the video embeddings encode visual features extracted from the video frames. All these inputs are projected into a common feature space using linear layers, ensuring they share the same dimensionality.

After projecting the inputs into a unified feature space, the architecture follows a structured workflow composed of three key stages, each designed to handle a specific aspect of the integration and reasoning process:

- **Positional Encoding** : Once projected, positional encodings are added to these embeddings to encode the sequential nature of the data, such as the order of tokens in the text or the temporal sequence of video frames. This step is crucial since transformers lack an inherent understanding of order, and positional encodings enable the model to capture these relationships effectively.

- **1st Decoding Stage** : The processing pipeline involves two decoding stages. In the first stage, the model integrates the question embeddings with the video embeddings using a series of transformer decoder layers. Each layer applies self-attention to the question embeddings to capture intra-question relationships and cross-attention to link the question tokens to relevant video features. This step produces refined question embeddings that are contextually informed by the video content.

- **2nd Decoding Stage** : In the second decoding stage, the model processes the answer embeddings with the refined question-video embeddings generated in the first stage. Here, the self-attention mechanism captures intra-answer relationships, while cross-attention links the answers to the question-video embeddings. The output of this stage is a set of contextualized answer embeddings, where each answer representation reflects its relevance to the question and the associated video.

To make a final prediction, the model extracts a special [CLS] token from the output of the second decoder for each answer. This token serves as a summary of the answer's contextual representation. These [CLS] tokens are passed through a classification layer, which generates logits for each answer, indicating the model's confidence in each being correct. The logits are reshaped to align with the batch size and the number of choices, and the model predicts the answer with the highest score.

Overall, by processing questions, answers, and video embeddings through two stages of decoding, the model tries to capture the dependencies necessary for a multimodal task like Video Question Answering.

5.3.2 TGIF CMT

The model pipeline begins with the Paired Dataset with Embeddings function, which preprocesses the input data and generates paired embeddings for text and videos. For textual inputs, the TGIF Text Encoder is used to create embeddings for questions and their associated answer choices. This encoder incorporates pre-trained BERT embeddings alongside additional masked multi-head attention layers to refine the textual representations.

For videos, embeddings are extracted using a pre-trained visual model, which encodes temporal and spatial features from GIFs or video clips into a fixed-dimensional representation. The text embeddings and video embeddings are paired with their corresponding labels and stored as a dataset. This step ensures that both modalities are aligned and prepared for input into the cross-modal transformer.

The model architecture is characterized by 3 main elements:

1. **Cross-Modal Embedding Layers**, responsible for integrating text and video embeddings. Text embeddings are processed through the pre-trained BERT backbone, while video embeddings are projected into the same dimensional space using a fully connected layer. The embeddings are concatenated along the sequence dimension, creating a unified representation of both modalities. A layer normalization and dropout operation are applied to stabilize training and improve generalization.

2. The **CMT Encoder** which is a stack of transformer layers, each consisting of multi-head self-attention and feed-forward networks. This encoder refines the joint embeddings by modeling complex interactions between the textual and visual tokens. Attention masks are used to differentiate between text and video tokens, allowing the model to attend selectively to relevant parts of each modality.

3. The **CMT Decoder** extends the encoder's capabilities by incorporating cross-attention mechanisms. Each decoder layer includes self-attention, cross-attention (to attend to the encoder's outputs), and a feed-forward network. The decoder further processes the joint embeddings, emphasizing task-specific representations.

The final output of the decoder is passed through a pooling layer, which extracts a fixed-size vector representation. This can be done using the CLS token embedding or mean pooling. The pooled representation is then fed into a classification head, consisting of fully connected layers with ReLU activation. The classifier outputs logits corresponding to the target classes, enabling the model to perform tasks such as multiple-choice question answering.
The model calculates cross-entropy loss during training to optimize its predictions. This loss function measures the difference between the predicted logits and the true labels, guiding the model to improve its accuracy over time.

6 Experiments

To evaluate the flexibility and effectiveness of our Cross-Modal Transformer models, we conducted several experiments with different configurations. These experiments were designed to test various hypotheses about how the architectures and components contribute to performance in VideoQA tasks. Below are the key experiments we explored:

- **Baseline Configuration** to establish reference performancse for the models using a standard setup with a BERT-based text encoder, pre-trained video encoder, and default transformer architectures

- **DistilBERT** instead of BERT to understand if a smaller and faster text encoder like DistilBERT can reduce computational costs without compromising accuracy

- Combining BERT with **T5-Small** to leverage their complementary strenghts and evaluate whether a hybrid architecture improves performance

- For **TGIF only** : **Increasing the number of Hidden Layers** to test if adding more depth to the transformer architectures improves their capacity to learn complex relationships between video and text

- For **MSRVTT only** : Using **cross-attention** between **questions** and **answers** within the text encoder did not allow the model to sufficiently capture the relationships between the input modalities. As a result, it was decided to remove cross-attention from the text encoder and instead introduce the current Cross-Modal Transformer, enabling a more meaningful integration of information across all input modalities.

7 Results

In this section the results achieved by our experiments are presented. Both architectures were trained and tested on Kaggle using GPU 100.

7.1 MSRVTT-MC CMT Results

All experiments on MSRVTT-MC CMT were conducted on the full dataset, consisting of approximately 2000 videos. By performing a 60-40 train-test split, we obtained 1200 videos for training and 800 for testing. The transformer was trained using mixed precision for around 50 epochs, achieving an accuracy of 95% and a loss of 0.1835.

To explore potential improvements to our network, Instead of using BERT encoder layers and embeddings for the text encoder, we explored how T5 and DistilBERT would perform in the Cross-Modal Transformer. While DistilBERT slightly outperformed the baseline, achieving an accuracy of 96.92% and loss of approximately 0.1665, T5 performed slightly worse, with an accuracy of 93.33 % and loss of 0.2290.

BERT likely outperformed T5 because its architecture and pretraining objectives align more closely with the requirements of the MC-VQA task. Its ability to generate compact, discriminative embeddings optimized for classification and alignment with video features gave it an advantage. In contrast, T5's generative focus and sequence-to-sequence structure may have introduced challenges in fine-tuning and effective multimodal alignment.

Despite the strong performance during training, the model did not surpass the state-of-the-art on the test set, achieving an accuracy that ranges between 38% and 41%.

7.2 TGIF CMT Results

All the experiments for TGIF CMT were executed on 1200 video-text pairs and trained over 100 epochs.

The baseline setup performed exceptionally well for TGIF CMT, providing a solid foundation for further experimentation. It effectively aligned visual and textual features for VideoQA tasks, achieving an impressive accuracy of 97.57% and a loss of 0.0906. Interestingly, swapping out BERT for DistilBERT slightly boosted performance while reducing computational demands. Accuracy improved to 97.85%, and the loss dropped to 0.0798. This highlights that DistilBERT's streamlined design is more than capable of handling the complexity of VideoQA tasks in this scenario.

On the other hand, the hybrid architecture showed promise but didn't outperform the baseline in test. It appears this approach needs more fine-tuning to balance the interaction between the two encoder types and fully leverage their complementary strengths.

Increasing the number of hidden layers turned out to be counterproductive for TGIF CMT. Accuracy took a significant hit, dropping to 61.01% during training. This emphasizes the need to carefully balance depth and complexity in the model architecture to avoid diminishing returns in performance.

However, despite strong results on the training set, the model struggled to generalize well on the test set with performance ranging between 20% and 25% depending on the text-video pairs considered. This is likely due to the limited number of samples available for testing, which restricts the model's ability to effectively learn patterns that generalize across unseen data.

7.3 Comparison with SOTA

Method	Action
ST-VQA [4]	60.8
Co-Memory [3]	68.2
PSAC [10]	70.4
Heterogeneous Memory [1]	73.9
HCRN [8]	75.0
QueST [5]	75.9
CLIPBERT [9]	82.8
MERLOT [14]	94.0
VIOLET [2]	92.5
Ours	25.1

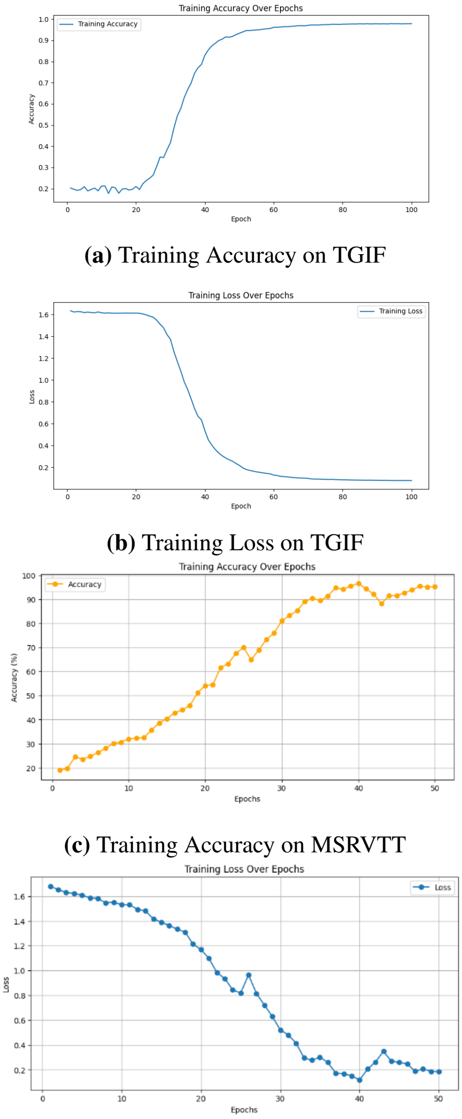
Table 1: TGIF-QA test set.

Method	Accuracy
SNUVL [13] (by [12])	65.4
ST-VQA [4] (by [12])	66.1
CT-SAN [13] (by [12])	66.4
MLB [7] (by [12])	76.1
JSFusion [12]	83.4
ActBERT [16] PT	85.7
CLIPBERT [9]	88.2
MERLOT [14]	90.9
VIOLET [2]	91.4
Ours	38.9

Table 2: MSRVTT multiple-choice test set.

8 Conclusions

This work explored the development and evaluation of two Cross-Modal Transformers (CMT) tailored for VideoQA tasks using the TGIF and MSRVTT datasets. By leveraging state-of-the-art architectures and pre-trained models, we achieved strong results, particularly with the baseline setups and their adaptation using DistilBERT, which demonstrated the potential of our architectures for



(a) Training Accuracy on TGIF

(b) Training Loss on TGIF

(c) Training Accuracy on MSRVTT

(d) Training Loss on MSRVTT

Figure 1: Visualization of training and evaluation metrics across TGIF and MSRVTT datasets. The top row shows training accuracy and loss on the TGIF dataset, while the bottom row depicts training accuracy and loss on the MSRVTT dataset.

aligning visual and textual modalities effectively. However, experiments also showed that there were limitations, especially in generalization to unseen data, as reflected in the overall lower performance of the models on the test set. Where this work might fall short, future work will involve extending the dataset with more video-text pairs. This will result in a more varied and richer training environment; thus, the models will be better at generalizing and will perform consistently across both training and testing scenarios.

We also want to explore some architectural modifications, such as reducing the number of attention heads for better computational efficiency and using other pre-trained language models like GPT-2 to check their compatibility with the VideoQA task. These adjustments will help to strike the balance between complexity and performance so that the models could further improve their adaptability and effectiveness.

Contributions. This work was a collaborative effort. All authors contributed equally to the project. In particular, the MSRVTT-MC dataset preprocessing and corresponding Cross-Modal Transformer (CMT) was developed by Giuseppina Iannotti, while the work involving the TGIF dataset and its corresponding CMT was carried out by Maria Emilia Russo.

References

[1] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1999–2007, 2019. 7

[2] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet : End-to-end video-language transformers with masked visual-token modeling, 2022. 1, 2, 3, 7

[3] Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. Motion-appearance co-memory networks for video question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6576–6585, 2018. 7

[4] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 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, 2017. 7

[5] Jianwen Jiang, Ziqiang Chen, Haojie Lin, Xibin Zhao, and Yue Gao. Divide and conquer: Question-guided spatio-temporal contextual attention for video question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11101–11108, 2020. 7

[6] Aisha Urooj Khan, Amir Mazaheri, Niels Da Vitoria Lobo, and Mubarak Shah. Mmft-bert: Multimodal fusion transformer with bert encodings for visual question answering. *arXiv preprint arXiv:2010.14095*, 2020. 2

[7] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325*, 2016. 7

[8] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9972–9981, 2020. 7

[9] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling, 2021. 1, 2, 3, 7

[10] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8658–8665, 2019. 2, 7

[11] Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. Bert representations for video question answering. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1556–1565, 2020. 2

[12] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, pages 471–487, 2018. 7

[13] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3165–3173, 2017. 7

[14] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in neural information processing systems*, 34:23634–23651, 2021. 2, 7

[15] Yaoyao Zhong, Junbin Xiao, Wei Ji, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. *arXiv preprint arXiv:2203.01225*, 2022. 2

[16] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755, 2020. 7