Improvements on the Hierarchical Conditional Relation Networks for VideoQA tasks

Nicolas DUFOUR
ENS Paris Saclay
4 Avenue des Sciences, 91190 Gif-sur-Yvette
nicolas.dufourn@gmail.com

Abstract

The objective of this project is to reproduce and try to improve the results of the paper "Conditional Relation Networks for Video Question Answering." [7]. The goal is to be able to answer questions about some videos. The proposed architecture leverages a novel hierarchical architecture.

1. Introduction

Question Answering is a task that consist in answering questions depending on some data. This has been done for text based question answering or image based question answering. This task is more challenging since it relies on multiple things. We need to have a good semantic understanding of the question and a good representation of the data we query on. Here since, we query on video, we would require both a good spatial and temporal understanding of the data. Video can also bring multiple modalities such as sound or subtitles. This means that questions answering demands a multi-modal understanding of the data.

In this work we are going upon the work of [7]. The authors propose a building block the Conditional Relational Network (CRN) unit to capture the relationships between the different modalities. This blocks then build an architecture called the Hierarchical Conditional Relational Network (HCRN). In this project, we are going to reproduce the results of the authors and build upon them to improve. We are also going to adapt the model to include new modalities such as subtitles.

The task we are trying to solve consist in training a model that is going to be able to answer different questions about a video. The questions can span to a frame-wise question but also to the whole video. For example, we can try to answer questions about an action. This needs a good spatio-temporal understanding from the model. Also, this task needs some semantic modeling to understand the question and formulate the correct answer.

2. Presentation of the model

We are going to present the model introduced by [7]. This model aims to capture the dependencies between the different modalities. It is built in a hierarchical fashion: We first handle smaller chunks of videos (clips) that we later combine at a video level. That way the model hope to capture local features as well as more global ones.

2.1. Preprocessing

2.1.1 Visual Inputs

For the visual inputs, we process videos as a succession of frames $V_1, ..., V_s$ For each frame we take the features from pre-trained ConvNets models. We take a classical ResNet[5] feature extractor to extract appearance features. We divide the videos into smaller clips. Then for each clip, we use a ResNext [4] network to extract motion features. Because of material constraint, we actually use the extracted features provided by [7].

2.1.2 Textual Inputs

For the textual input we pull features using a GloVe [12] model, that will provide some textual features, as the authors of [7] do to reproduce their results. They use this text model to encode questions and answer proposal.

However, in this work, we decided to explore an other text modeling techniques, a Transformer based model, the BERT model [2]. We will use the Hugging Face [13] Library to use a pre-trained model. We will detail how we use such a model and conduct a study on the best way to use a BERT[2] model in the VideoQA task.

2.2. The HCRN Unit

For the HCRN model we are going to stack multiples CRN (see Figure 1). We first divide the videos into multiple clips. For each clip, we run it through 2 CRN. The first one is conditioned according to the motion of the clip and the second according to the question.

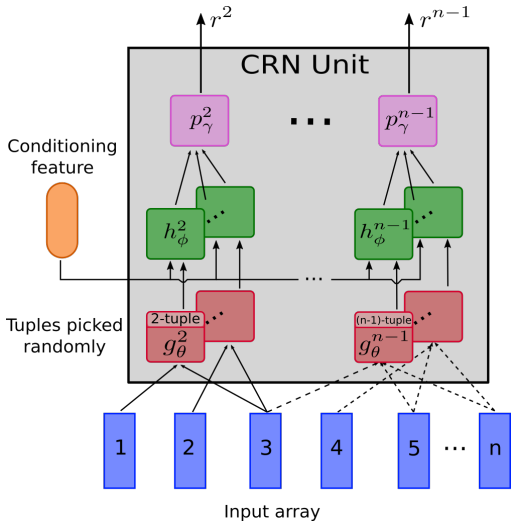

Figure 1. The CRN unit. Figure taken from [7]

Then, we run the resulting output to 2 video level CRNs. The first one is conditioned according to a video-level motion features and the last one is conditioned again according to the question. Finally, we condition the output with the features using an attention mechanism and run it through an output layer (classification layer or regression layer according with the task). For more details on the model, see [7]

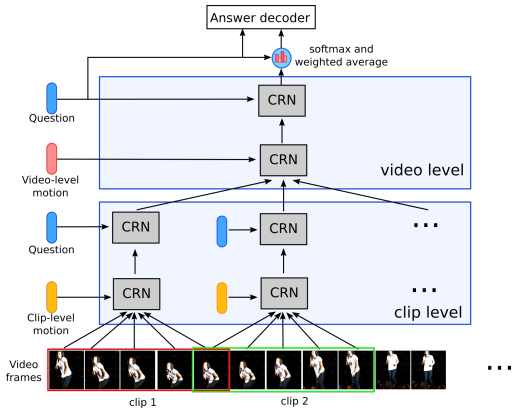

Figure 2. The HCRN model. Figure taken from [7]

3. Experiments

3.1. Datasets

We reproduce the results over 3 datasets: MSRVTT-QA [14], MSVD-QA [14] and TGIF-QA FrameQA [6].

MSVD-QA [14] dataset has a total number of 1,970 video clips and 50,505 question answer pairs.

MSRVTT-QA [14] contains 10K video clips and 243k question answer pairs

TGIF-QA FrameQA [6] is based on a GIF database and has 39,479 GIFs with 53,083 question answer pairs

We also are going to try this model over TVQA [8] a dataset based on TV Shows, where we are going to exploit a new modality: Subtitles. TVQA has 152.5K questions pairs over 21.8K clips

3.2. Improving the training

First, we first recreate the results. The original paper uses an SGD optimizer with a learning rate of 0.0001. We recreate the results for 3 datasets: MSVD-QA , MSRVTT-QA and TGIF-QA FrameQA. We achieve to recreate the results using the same hyper parameters.

However, we see that the model usually begin to over-fit after 2 or 3 epochs. This makes the rest of the 25 epochs a bit useless. We then try to reduce the over-fitting. After trying multiple variations, changing the optimizer to AdamW[10] allows us to increase the performance of our model. Seeing the performances increase we can see at Table 1 that MSVD-QA has the most improve. Indeed, MSVD-QA was the dataset on which the model over-fitted the quickest. This could support our hypothesis that reducing over-fitting could lead to better performances. Indeed, AdamW his known to better handle weight decay, and weight decay has the property to help with over-fitting. For the rest of the experiments, we will use AdamW as an optimizer.

Optimizer	MSVD-QA	MSRVTT-QA	TGIF-QA FrameQA
SGD	0.3598	0.3534	0.5547
AdamW	**0.368**	**0.3549**	**0.5574**

Table 1. Accuracy of HCRN depending of the optimizer used

3.3. Replacing Glove by Bert

In this section, we are going to discuss our attempt to replace the Glove text encoding by a BERT text encoder. We will use the BERT pre-trained model from the Hugging Face library. To make BERT work, we need to explore different ways to use it.

We first are going to train it as a feature extractor: While training the HCRN, we aren't going to propagate the gradients through the BERT architecture. We are going to extract the last hidden layer (1 hidden frozen) as features. According to the authors of the original BERT paper [2], the most optimal way to use BERT as a feature extractor is to concatenate the 4 last hidden layers (4 hidden cat frozen) and use this as an embedding.

We are also going to fine-tune beforehand the pre-trained BERT model with the question dataset hopping to adapt the model to our dataset and getting better results. To do this, we do mask modeling on the question dataset. We then use 1 hidden layer (1 hidden adapted) or 4 hidden layer (4 hidden cat adapted).

Finally, we are also going to fine-tune the model while

training the HCRN. Since BERT is a big model we try to only train the 4 last layers but also train it fully. We fine-tune BERT with a lower learning rate than the rest of the HCRN (1e-5). We also try to fine-tune the Roberta model in the same fashion as the best model

Training Technique	Results
Glove	0.5574
1 hidden frozen	0.5228
4 hidden cat frozen	0.5308
1 hidden adapted	0.5249

Training Technique	Results
4 hidden cat adapted	0.5279
finetune last 4	0.5485
Finetune BERT	**0.5579**
Finetune Roberta	0.552

Table 2. Accuracy of HCRN depending of the text embedding technique on TGIF-QA FrameQA

As we see in Table 2 and 3, we improve only slightly using the BERT model. We also see that training the results is necessary for optimal performances. If we don't want to train the model, the best techniques is to concatenate the 4 last hidden BERT layers according to our results on TGIF-QA FrameQA.

One can explain this results by 2 factors: The first one is that the question corpus is actually composed of small sentences and therefore, BERT doesn't shine. The other reason is by the construction of the dataset. Indeed, this QA questions are constructed automatically and some of them aren't grammatically correct. The worlds are often ordered in a random way. Therefore, BERT could not shine since it's actually reading a set of words and not a correct grammatical sentence.

Text embeder	MSVD-QA	MSRVTT-QA	TGIF-QA FrameQA
Glove	**0.368**	**0.3549**	0.5574
BERT	0.3522	0.3432	**0.5579**

Table 3. Accuracy of HCRN depending of the text embedding on 3 datasets

4. Ablation study

To better understand our model, we are going to run an ablation study. We are going to remove different features to the model. We fill the vector with zeros to keep the same number of parameters in the model. We remove all motion conditioning, only clip level motion conditioning, only video level motion conditioning, all question conditioning, only clip level question conditioning, only video level question conditioning.

We also remove the video features. First we remove all the visual features (appearance and motion) and also only appearance.

Ablation	MSVD	TGIF-QA FrameQA
No motion	0.3467	0.5555
No clip Motion	0.353	0.5456
No Video Motion	**0.3587**	0.5536
No Question Cond	0.357	0.5503
No Question clip level	0.348	0.5452
No Question Video Level	0.3511	0.5514
No visual Features	0.282	0.4578
No Appearance Features	0.3385	0.4962
HCRN BERT	0.3522	**0.5579**

Table 4. Ablation Study on MSVD and TGIF-QA FrameQA (Accuracy)

We can see in Table 4 that we have surprising results. First removing motion or question conditioning has almost no impact on the results. This means that this conditioning might be unnecessary.

The most surprising result is what happen when we remove the visual features: We actually keep pretty good performances. This means that the model can rely only the question itself to answer the questions. Our model isn't actually taking a lot of information from the video, it relies on the knowledge of BERT to answer questions. It also points out to a flaw in the datasets: Some questions are too easy to answer without any other information than the question.

4.1. Adding a new modality: Subtitles

For this part we are going to try to include a new modality: subtitles. Subtitles can contain useful information of what's happening in the video. To add the subtitles, we segment the subtitles following the clips segmentation. Doing that we hope to match the subtitles with the relevant frames. We then handle subtitles as a conditional feature in the HCRN. We replace the motion features by subtitles features.

We also try to do subtitles pre-conditioning. The idea behind this is that all the subtitles aren't relevant to the question. So we take the subtitles, concatenate them with the question and run it thought an MLP with ELU activation before giving it as a conditional feature to the HCRN. We use an LSTM to compute the video level subtitles features.

The problem is that we didn't manage to run our models for a full training run. The results we have are after 5 epochs since for this dataset 1 epoch takes a few hours. This results could be improved with a bit more time.

Model	Accuracy
HCRN No Subtitles	0.4001
HCRN Subtitles (w/o precondtioning)	0.4169
HCRN Subtitles (with precondtioning)	0.4198

Table 5. Accuracy on TVQA

We can see that the subtitles improve the score of the

HCRN but not by much. This might be due of the small number of epochs we had to use. However, one can also conjecture that the data in the conditioning features isn't very effective as we've seen with TGIF-QA. A future work could explore what would we get if we consider the subtitles as an entry (instead of visual features) of a CRN. We could then combine the 2 HCRN to answer question having independent understanding of both the visual features and the subtitles features. We also see that the preconditioning improves the score, if we consider the subtitles as an entry of a CRN we should first precondition them.

5. Implementation details

We reuse the model code from [7]. However, we refactor the code using Pytorch Lightning[3] to make it easier to train. This allows us to use half precision training to diminish the training time. We also rewrite the data-loading to load the data in the RAM, allowing us to speed up the training dividing by 2 the time needed to train the model. Finally, we use Weights and Biases [1] to cleanly log our experiments.

The codes can be finded at `https://github.com/nicolas-dufour/video-qa-recvis`

6. Conclusion

In this work, we manage to reproduce the results from [7] and deepen our understanding of the model. We explore the use of a BERT model to handle the language modeling and find out that the improvements weren't that substantial. We also found some flaws in the model: The conditioning features have low impact on the performances of the model. We also find out that the datasets used to evaluate VideoQA can a be solved by only looking at the question for a portion of such dataset. Finally we attempted a new modality to the model: Subtitles. We didn't achieve a huge improvement due to time limits but have layed ground for future progress.

As we saw, it could be interesting to consider an independent model to handle subtitles instead of using them as conditioning features. That could allow to retrieve more information from the subtitles. Finally, future work should focus on avoiding the over-fitting of the model. Indeed, it has quickly appear that over-fitting is a big bottleneck to training. To do that, we should explore the impact of self-supervised training like the HERO model [9] to be able to exploit huge datasets such as [11] We should also see if a transformer could bring a better video representation to allow to extract more information from the visual features.

References

[1] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: `https://www.wandb.com/`.

[2] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: `1810.04805`. URL: `http://arxiv.org/abs/1810.04805`.

[3] WA Falcon. "PyTorch Lightning". In: *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning* 3 (2019).

[4] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6546–6555.

[5] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[6] Yunseok Jang et al. "Video Question Answering with Spatio-Temporal Reasoning". In: *IJCV* (2019).

[7] Thao Minh Le et al. "Hierarchical Conditional Relation Networks for Video Question Answering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[8] Jie Lei et al. "TVQA: Localized, Compositional Video Question Answering". In: *EMNLP*. 2018.

[9] Linjie Li et al. "HERO: Hierarchical Encoder for Video+ Language Omni-representation Pre-training". In: *EMNLP*. 2020.

[10] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[11] Antoine Miech et al. "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips". In: *ICCV*. 2019.

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: `http://www.aclweb.org/anthology/D14-1162`.

[13] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

[14] Dejing Xu et al. “Video Question Answering via Gradually Refined Attention over Appearance and Motion”. In: *ACM Multimedia*. 2017.