INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Integrating Video Retrieval and Moment Detection in a Unified Corpus for Video Question Answering

Hongyin Luo[1], *Mitra Mohtarami*[1], *James Glass*[1], *Karthik Krishnamurthy*[2], *Brigitte Richardson*[2]

[1]MIT Computer Science and Artificial Intelligence Laboratory
[2]Ford Motor Company

{hyluo,mitram,glass}@mit.edu,{kkrish65,bricha46}@ford.com

Abstract

Traditional video question answering models have been designed to retrieve videos to answer input questions. A drawback of this scenario is that users have to watch the entire video to find their desired answer. Recent work presented unsupervised neural models with attention mechanisms to find moments or segments from retrieved videos to provide accurate answers to input questions. Although these two tasks look similar, the latter is more challenging because the former task only needs to judge whether the question is answered in a video and returns the entire video, while the latter is expected to judge which moment within a video matches the question and accurately returns a segment of the video. Moreover, there is a lack of labeled data for training moment detection models. In this paper, we focus on integrating video retrieval and moment detection in a unified corpus. We further develop two models–a self-attention convolutional network and a memory network–for the tasks. Experimental results on our corpus show that the neural models can accurately detect and retrieve moments in supervised settings.

Index Terms: video question answering, video retrieval, moment detection

1. Introduction

With the tremendous increase in new devices and machines, people are not always aware of the various features and functions of their devices or how to use new functions they have never tried before. Fortunately, there are a growing multitude of video instructions on the web to help people understand how to use their devices and corresponding functions. Most people usually enter their questions as a query to a video search engine, e.g. Youtube, to search for an instruction. However, search engines have limitations. Firstly, they are not designed for answering questions. Secondly, they retrieve entire videos - which can be very long. Therefore, users need to manually search inside a video to find their desired answers.

To address these limitations, several models have been proposed to find the users' interest points in videos. Unfortunately, these approaches are confounded by another limitation: lack of labeled data, including manual annotation of video segments and moments. While deep neural networks with some attention mechanisms can infer and extract such moments automatically in an unsupervised way, potentially better results can be achieved when having the target moments provided in advance, which enables supervised or semi-supervised training of the attention. This would allow not only more reliable video retrieval, but also better moment detection.

Figure 1: *An example of moment detection in a video for an input question "what does the auto function for air conditioner do?"*

Following this idea, we have developed a corpus that identifies the related videos and its segments to provide an accurate answer for an input question. In our corpus, each video introduces a set of devices and describes their aspects and functions. The videos also include instructions about how users can operate, or interact with the devices. We annotated the videos to manually split them into smaller segments, where each segment focuses on a single aspect or a single function for answering users' questions more directly. For example, the question in Figure 1, "What does the auto function for air conditioner do?" can be answered by a 30s segment, instead of the entire video.

We develop two models–self-attention convolutional neural networks and memory neural networks [1]–with our corpus for the video retrieval and moment detection tasks. The models encode input questions, videos and their segments into their embedding representations, and use attentions over the encoded representations to retrieve the best video and to detect the desired moment for answering the input question. In general, the experiments show that (*i*) the moment detection task is more challenging than the video retrieval task, and (*ii*) the models can significantly perform better if they use the labels for video segments/moments during training, and (*iii*) our models outperform the YouTube baseline.

http://dx.doi.org/10.21437/Interspeech.2019-1736

2. Related Work

Our work and collected corpus have the following features:

- **Modality:** In our corpus, there are textual questions with answers in two modalities–videos and transcripts.
- **Integration:** In our corpus, there are videos that discuss more general topics, e.g., introducing the air conditioner (AC) of a vehicle. These videos have different segment parts about more detailed sub-topics, e.g., how to turn on the AC or how to adjust it. The former is related to video retrieval and the latter is related to the moment detection task. This enables us to integrate both video retrieval and moment detection tasks to find the exact answer of a given question.
- **Chain:** In our corpus, a video is highly focused on a certain topic with segments that are usually contain similar sub-topics and highly related to each other. This makes the moment detection task more challenging.

With respect to the above features, we compare our work with following previous work categories.

2.1. Video Retrieval.

[2] presented a model with rich hand-crafted features to retrieve the related videos for a given query, while [3] proposed a model that jointly learns video and language embeddings for a better retrieval task. [4] showed that videos can be retrieved with image queries, while, instead of merely using visual inputs, [5] used transcripts to improve the performance of video retrieval. The studies in this category aim to retrieve the videos related to a specific query without considering which segment of the video is the exact answer to the given query (i.e., moment detection). Thus, our work is different from this category in terms of the *Integration* and *Chain* features of our work.

2.2. Visual/Video Question Answering.

[6] and [7] worked on the Visual Questions Answering (VQA) task and respectively presented `MSCOCO` and `VQA` datasets focused on answering questions about scene understanding, and [8] presented a multi-turn visual question corpus. While VQA is developed for images, our work focuses on videos. In video QA, [9] presented the `MPII-MD` dataset that contains movies and their descriptions. [10] presented the `MovieQA` dataset which contains collected movies, subtitles, stories, questions, and candidate textual answers for multiple choice questions. The answers could be generated or selected from the textual candidates. In contrast to their work, we aim to retrieve videos that include answers for a given question and then detect the moments of the retrieved videos that provide the best answers. Thus, our work is different in terms of the *Integration* and *Chain* features.

2.3. Community Question Answering.

Given a Community Question Answering (cQA) thread containing a question and a list of answers, the work in this category aims to automatically rank the answers according to their relevance to the question [11, 12, 13, 14, 15]. The answers may have some relation to each other [16, 17], but in general they are written by different users and are mostly independent. Thus, this is different from the *Chain* feature of our work. Furthermore, our work is different in terms of the *Integration* and *Chain* features.

Table 1: *Statistics of collected videos, transcripts, and questions.*

	Videos	**Segments**
Videos		
1. Num. of Videos	107	464
2. Avg. Num. of seg.	4.34	-
3. Total Length (sec)	9,605.35	-
4. Avg. of Length (sec)	89.77	20.70
5. Min. of Length (sec)	11.45	4.13
6. Max. of Length (sec)	292.87	104.3
Transcripts		
7. Avg. Num. of Words	264.50	60.99
8. Total Num. of Words	28,301	28,301
9. Vocab. Size	2,489	2,489
Questions		
10. Num. of Questions	-	9,482
11. Num. of Ques./seg.	-	20.44
12. Avg. Num. of Words	-	9.32
13. Vocab. Size	-	3,329

3. The Corpus

Our corpus contains videos and their transcripts, where each video is divided into several segments (i.e. video clips) and each segment is annotated with a set of questions, and each question has one related answer. Overall, the process of corpus creation has several stages: (*i*) video extraction, (*ii*) video segmentation, and (*iii*) question annotation, which we describe below.

We consider a YouTube channel–Ford Motor Company[1]–as the source of our videos. This channel contains *How-To* videos that introduce a set of functions on vehicles, e.g., "*How to Check Your Tires with the Penny Test?*" We collected all 107 *How-To* videos on this channel, and transcribed them as a part of this corpus. The statistics of the videos and transcripts, e.g., the lengths of videos and transcripts, the vocabulary size, are presented in rows 1–9 of Table 1.

Following our aim of detecting the moment of a video with respect to a given question, we split each video into segments based on its transcript. Each segment includes one or more complete sentences and can be used to answer the How-to questions about a specific topic. For example, if a video is about the air conditioner (AC) system of a vehicle, a segment might introduce how to turn on the AC or the function of the "AUTO" button on the panel. The annotators also provide questions based on a single video segment instead of watching the entire video. The statistics of the segments are shown in Table 1.

We have used Amazon Mechanical Turk[2] (AMT) to collect questions for the video segments. To collect enough questions, each video segment was assigned to 10 to 12 turkers, who were asked to submit two different questions that were answered by the content of the given video segment. Note that only the videos were shown (i.e., no transcripts) to try to minimize the bias effect of having turkers use the exact same words for their submitted questions. In total, we collected around 10K questions. Rows 10–13 in Table 1 show the statistics of the collected questions, and an example of the questions is shown in Figure 1. The corpus and the implementation of our models are publicly available[3].

[1] `https://www.youtube.com/user/ford`
[2] `https://www.mturk.com/`
[3] `https://github.com/luohongyin/VehicleVQA`

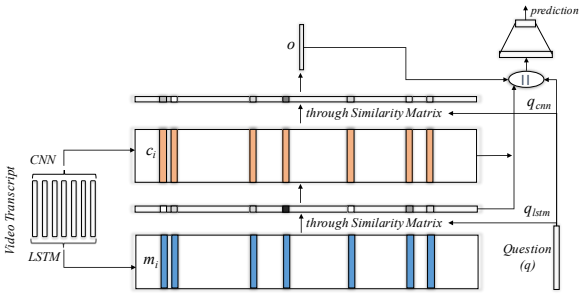

Figure 2: *The simplified architecture of the applied MemNN model [1] for video question answering. Each video segment is stored in a memory cell and the final output predicts the entire video related to given question. We conduct moment detection with the attention weights assigned to video segments.*

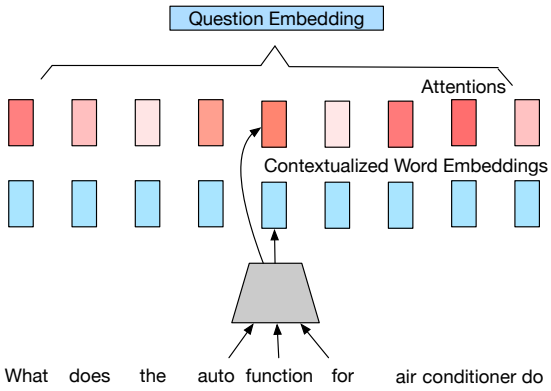

Figure 3: *The architecture of the SACNN model. The blue blocks stand for word-level and sentence-level distributed embeddings. The red blocks stand for the attention weights assigned to each word. The sentence/question embedding is calculated by averaging all word embeddings with the attention distribution.*

4. Models

In this work, we evaluate three models for video retrieval and moment detection tasks using our corpus; YouTube video retrieval search engine[4], self-attention convolutional neural networks and memory neural networks.

We use the YouTube API[5] as a baseline model. Given a question as a query to the API, it attempts to find the related videos to the question from the specific channel used for our corpus as explained in Section 3. Then, it retrieves a ranked set of videos, and this set is used to evaluate the performance of YouTube.

4.1. MemNNs

A memory network model with a similarity matrix is proposed in [1] for a stance detection task with the capability of extracting rationales from documents with respect to given claims. In this work, we investigate the model for the video retrieval and moment detection tasks. We give a question and a video including all its segments to the model and it outputs a score for the video and a set of scores corresponding to the video segments–as rationales–that indicate the relatedness of the video and its segments to the input question. In this work, we employ the same MemNN architecture proposed in [1]. The architecture of the model is simplified in Figure 2 [1].

We train a single MemNN for both video retrieval and moment detection. The only supervision signal we used for the training is the video-level labels. Thus, the moment detection task is a semi-supervised task for MemNN. In this work, we use the final output of the MemNN to find the best video and use the attention weights over segments–computed by similarity matrix [1]–to find the best segment.

4.2. SACNNs

We also present the self-attention convolutional neural network (SACNN) model for our tasks. We apply a convolutional neural network (CNN) as the first step to encode the words and their contexts to their embedding representations. Upon the CNN layers, we apply a self-attention mechanism [18, 19, 20] as a pooling layer to generate fixed-length embeddings for input questions, videos, and video segments. These embeddings are used to calculate the cosine similarity between the input questions with the videos or their segments. Then, these scores are used to select the top N videos as possible answers. The details of the model are shown in Figure 3.

In practice, we use a two-layer CNN with ReLU activation function for the hidden layer. The size of the convolution window is 5, for both layers. The output of the CNN consists of two parts–an attention score and a embedding vector. The feed forward process of the SACNN is shown as follows,

$$H = ReLU(W_h * X + b_h) \tag{1}$$

$$\hat{E}, Y = W_y * H + b_y \tag{2}$$

$$E = ReLU(\hat{E}) \tag{3}$$

where $E = [e_1, e_2, \ldots, e_n]$ stands for the context embeddings output by the CNN module. $Y = [y_1, y_2, \ldots, y_n]$ is a vector of scores. We then calculate an attention distribution with the scores Y. W_h, W_y, b_h, and b_y are learnable parameters tuned during training. The attention of the i-th word, α_i, is

$$\alpha_i = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{4}$$

With the attention distribution over the input question or transcript, the final sentence embedding is calculated as:

$$s^t = \sum_i \alpha_i \cdot e_i \tag{5}$$

where $t \in \{Q, T\}$, standing for either question and transcript. In practice, the question and transcript encoders do not share parameters.

4.3. Training

In this work, we train our models with negative sampling. For each question-transcript pair, we randomly select 15 negative transcripts for the input question. With the question embedding q and a series of transcript embedding $[t^p, t_1^n, t_2^n, \ldots, t_m^n]$, where t^p stands for the positive sample and t_i^n are negative samples,

[4] https://www.youtube.com/

[5] https://www.youtube.com/yt/dev/api-resources/

we calculate the probability that a transcript is the answer to a question by

$$p(t_i|q) = \frac{e^{t_i} \cdot q}{\sum_j e^{t_j} \cdot q} \quad (6)$$

Then we update the parameters in the model with stochastic gradient descent (SGD) based on cross-entropy losses,

$$l = -\sum_i (y_i \cdot log(p(t_i|q)) + (1 - y_i) \cdot log(1 - p(t_i|q))) \quad (7)$$

where y_i is 1 for the positive samples and 0 for negative samples.

5. Experiments and Evaluation

We evaluate the models explained in Section 4 with our corpus on the video retrieval and moment detection tasks. We apply 10-fold cross-validation to evaluate the performance of our models, where all questions for the same video are assigned to the same fold. We report mean averaged precision (MAP) at $N = \{1, 5, 10\}$, which is the standard evaluation metric of ranking and retrieval tasks. We do not report the MAP@10 score of the local moment detection task, since many videos contain less than 10 segments. In our published corpus, we also split the folds the same way that was used in our experiments.

5.1. Video Retrieval

The experimental results for the video retrieval task are shown in Table 2 (rows 1–3). The results indicate that the YouTube API does not perform well (row 1), since traditional video search engines are not specially designed for retrieving information based on questions. This suggests that designing a special video question answering system for video instruction retrieval is necessary.

Both neural models significantly outperform the YouTube baseline. The MemNN and SACNN can achieve high performance, in particular for MAP@5 and MAP@10. The reason is that they encode the video transcriptions using embedding representations through an attention mechanism. With this mechanism, the models can highlight vehicle-related words and terms, which are more important and informative for the video retrieval model to make the decision. The MemNN performs relatively better than SACNN, because it applies higher-level attentions over video segments, helping the model to learn better representations of the entire videos.

5.2. Local Moment Detection

In this task, we assume the model is given a *related* video to an input question, and the model aims to retrieve the moment in the given video that makes the best answer to the question. The experimental results are shown in Table 2 (rows 4–5). We do not report the performance of Youtube search engine because it cannot perform moment detection.

Experimental results show that the SACNN leads to significantly better performance than MemNN, in particular for MAP@1. The reason is that SACNN uses the labels for video segments during training, while MemNN uses only the video labels–not segment labels–during training. Although not using explicit segment-level labels, the MemNN model still achieved high MAP@5 accuracy. This suggests that the segmental attention of the MemNN successfully captured some of the moments that directly answer the questions.

Table 2: *Experimental results of SACNN and MemNN models, and YouTube baseline for video retrieval and moment detection tasks. The experimental results show that our models can significantly outperform the YouTube baseline in the video retrieval task. The proposed models also perform well on moment detection task.*

	MAP@1	**MAP@5**	**MAP@10**
Video Retrieval			
1. YouTube	36.54	56.24	-
2. MemNN	65.02	**90.36**	**93.91**
3. SACNN	**66.69**	87.42	91.09
Local Moment Detection			
4. MemNN	37.38	80.17	-
5. SACNN	**77.94**	**97.65**	-
Global Moment Detection			
6. MemNN	24.53	54.66	75.14
7. SACNN	**57.13**	**80.75**	**85.20**

However, in real-life situations, the video question answering system is not provided with the groundtruth related video as the settings in the local moment detection task. Thus, the global moment detection task is more important.

5.3. Global Moment Detection

To align our experiments better with the real-life application, we propose the global moment detection task. In this task, we relax our assumption for the local moment detection (described above), where the model is given all *related* and *unrelated* videos for an input question, and is asked to retrieve the best moment from the entire given set of videos.

The experimental results are shown in Table 2 (rows 6–7). The retrieving performances of both models are lower than the local moment detection task, indicating that the problem becomes more difficult when considering all videos, which significantly enlarged the search space of the question-answering model.

With segment-level supervision, the SACNN model achieved higher performance than MemNN. The model is able to successfully find the moment with the best answer in its top-5 choices in around 4 out of 5 test cases. In addition, although the MemNN model for global moment detection is semi-supervised, it still attained a good MAP@10 performance. This indicates that the MemNN learns to retrieve the best video by paying attention to the most related video segments.

6. Conclusion and Future Work

In this paper, we have presented a novel corpus that unifies video retrieval and moment detection tasks. This is the first corpus to offer such a combination. We further developed a self-attention convolutional neural network, a memory network model and the YouTube video search engine, and evaluated them on our corpus. The results showed that the neural models can achieve better performance compared to the YouTube baseline. In future work, we plan to extend the annotations to cover other domains, other modalities such as spoken language, and other important aspects of video and moment retrieval such as personalized retrieval using the personal interests of users, which have been shown useful in previous research [21, 22, 23].

7. References

[1] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Màrquez, and A. Moschitti, "Automatic stance detection using end-to-end memory networks," in *Proceedings of the NAACL*, ser. NAACL-HLT '18, New Orleans, LA, USA, 2018.

[2] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 494–501.

[3] R. Xu, C. Xiong, W. Chen, and J. J. Corso, "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework." in *AAAI*, vol. 5, 2015, p. 6.

[4] A. Araujo and B. Girod, "Large-scale video retrieval using image queries," *IEEE transactions on circuits and systems for video technology*, vol. 28, no. 6, pp. 1406–1420, 2018.

[5] H. Yang and C. Meinel, "Content based lecture video retrieval using speech and video text information," *IEEE Transactions on Learning Technologies*, no. 2, pp. 142–154, 2014.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[7] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[8] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual Dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3202–3212.

[10] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "MovieQA: Understanding Stories in Movies through Question-Answering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] M. Mohtarami, Y. Belinkov, W.-N. Hsu, Y. Zhang, T. Lei, K. Bar, S. Cyphers, and J. Glass, "Sls at semeval-2016 task 3: Neural-based approaches for ranking in community question answering," in *Proceedings of NAACL-HLT Workshop on Semantic Evaluation*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 753–760.

[12] L. Màrquez, J. Glass, W. Magdy, A. Moschitti, P. Nakov, and B. Randeree, "SemEval-2015 Task 3: Answer Selection in Community Question Answering," in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015.

[13] Y. Belinkov, M. Mohtarami, S. Cyphers, and J. Glass, "Vectorslu: A continuous word vector approach to answer selection in community question answering systems," *SemEval-2015*, p. 282, 2015.

[14] P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. Glass, and B. Randeree, "SemEval-2016 task 3: Community question answering," in *Proceedings of the 10th International Workshop on Semantic Evaluation*, ser. SemEval '16. San Diego, California: Association for Computational Linguistics, June 2016.

[15] H. Nassif, M. Mohtarami, and J. Glass, "Learning semantic relatedness in community question answering using neural models," in *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 137–147. [Online]. Available: https://www.aclweb.org/anthology/W16-1616

[16] A. Barrón-Cedeno, S. Filice, G. Da San Martino, S. Joty, L. Marquez, P. Nakov, and A. Moschitti, "Threadlevel information for comment classification in community question answering," in *Proceedings of the ACL-IJCNLP*, vol. 15, 2015, pp. 687–693.

[17] S. Joty, A. Barrón-Cedeno, G. Da San Martino, S. Filice, L. Marquez, A. Moschitti, and P. Nakov, "Global thread-level inference for comment classification in community question answering," in *Proceedings of the EMNLP*, vol. 15, 2015.

[18] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," *arXiv preprint arXiv:1601.06733*, 2016.

[19] K. Tran, A. Bisazza, and C. Monz, "Recurrent memory networks for language modeling," *arXiv preprint arXiv:1601.01272*, 2016.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[21] X. Wang, L. Nie, X. Song, D. Zhang, and T.-S. Chua, "Unifying virtual and physical worlds: Learning toward local and global consistency," *ACM Trans. Inf. Syst.*, vol. 36, no. 1, pp. 4:1–4:26, Apr. 2017.

[22] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17, 2017, pp. 173–182.

[23] D. Cao, L. Nie, X. He, X. Wei, S. Zhu, and T.-S. Chua, "Embedding factorization models for jointly recommending items and user generated lists," in *Proceedings of the SIGIR*, New York, NY, USA, 2017, pp. 585–594.