Video Question Answering to Find a Desired Video Segment

Mayu Otani
Nara Institute of Science and Technology
otani.mayu.ob9@is.naist.jp

Esa Rahtu
University of Oulu
esa.rahtu@ee.oulu.fi

Yuta Nakashima
Osaka University
n-yuta@ids.osaka-u.ac.jp

Janne Heikkilä
University of Oulu
janne.heikkila@ee.oulu.fi

We refer to the task of finding one or more video segments in a video clip to fine-grained video retrieval, or FGVR in short (Figure 1). Various approaches can address this task. For example, existing video retrieval approaches [4, 5] that deal with short video clips can be applied by segmenting a long video into shorter ones, which may require sophisticated video segmentation or lose temporal dependencies among different segments. Another interesting approach can be judging if a frame matches the query or not with retaining temporal dependencies by using recurrent neural networks, which we call the frame-level approach.

In this demo, we extend the idea of FGVR to a question answering system that firstly answers to the question in a specific form (i.e., "Does this video contain a clip, in which ...?"
and show a corresponding clip for grounding. We implement a DNN-based system in the frame-level approach. One practical problem to realize this system is the lack of a dataset to train the DNN. We address this problem by concatenating randomly selected short video clips, which allows us to generate an arbitrary number of long videos with corresponding natural language queries.

2 DEMO SYSTEM OVERVIEW

Figure 2 shows the screenshot of our demo system. The top pane shows the video to be retrieved. "Open video" and "Play" buttons are to load the video to be retrieved and to play it back. Below these buttons is the text box to specify the question. The answer to the question (either "Yes" or "No") is shown below. The graph shows the frame-level relevance between the question (or the text in the text box) and the video. If the video has frames with relevance scores higher than a predetermined threshold, the system sets the answer to "Yes." Using the slider at the bottom, the user can freely browse the video. In the demo, users can try some multi-clip videos synthesized based on YouTube videos in the Microsoft Video to Text dataset [3] as well as movies from MPII Video Description datasets [2].

3 DNN-BASED FGVR

The key component of our demo system is DNN-based FGVR in the frame-level approach, that computes the frame-level relevance scores given a video and a natural language query. Figure 3 shows the network architecture. After the user specifies the video to be retrieved and inputs the question (or the query) in the text box, video X is decomposed into a sequence \((x_1, \ldots, x_T)\) of frames \(x_t\), where each frame is transformed into a feature vector \(V = (v_1, \ldots, v_T)\) using ResNet [1], and the query \(Y\) is decomposed into a sequence \((y_1, \ldots, y_M)\) of words \(y_m\).
The feature vector $v_t$ in $V$ computed from video frame $x_t$ is fed into bidirectional LSTM layers, which produce two hidden states for time step $t$. These hidden states are concatenated into a single vector and passed to a two-layer perceptron with the hyperbolic tangent nonlinearity to obtain the video encoding for this time step. Due to the bidirectional LSTM layers, the video embedding for each time step can contain temporal dependencies to describe the concept included in the nearby frames. For the word sequence $(y_1, \ldots, y_M)$ obtained from the query, each word $y_m$ is transformed into word vector and then a single LSTM layer is used to generate a query embedding. The video embedding and the query embedding have the same dimensionality (i.e., 256-D) so that the relevance score between them can be computed using the cosine similarity function. Since this is a very new task, there is no dataset that can be used for training this DNN. Therefore, we automatically synthesize multi-clip video and natural language query pairs based on existing datasets for video captioning (i.e., YouTube videos [3] and movies [2]). Firstly we pick out a single video clip in a dataset together with its corresponding caption, and then randomly pick out other two video clips in the dataset. These three videos are randomly shuffled and concatenated into a longer video clip. We use these data to train the DNN.

4 CONCLUSION

In this demo, we show how our question answering system over DNN-based FGVR works. This task can be the basis for various types of video retrieval applications, such as movie scene identification and event extraction in a surveillance video. The DNN of our current implementation is relatively simple but shows promising performance. Our next step is to evaluate our system in a more realistic scenario (e.g., movie scene identification), which requires making a dataset by human annotators. This work is partly supported by JSPS KAKENHI No. 16K16086.

REFERENCES