Automatic Caption Generation for Video Clips Using Keyframe and Document Summarization Techniques

Masaki Hoshino Takashi Yukawa

Nagaoka University of Technology, Niigata, Japan
s173350@stn.nagaokaut.ac.jp

Abstract

This paper presents the approach of the KsLab_NUT team in the TREC VID 2020[1] VTT Task. We propose a method that focuses on reducing the processing time. By extracting only important frames from videos and using them for processing, we were able to drastically reduce the number of frames to be processed while achieving certain levels of accuracy. Furthermore, we also applied the methods used for text summarization to examine their performance.

1 Introduction

The TRECVID VTT task requires to generate a single sentence that describes the content from a video. Generating caption from videos is a very challenging task, but with the advent of deep learning, it is also a task that is gaining more and more attention because more complex sentences can be generated. Recently, several deep learning models, including the Encoder-Decoder model, have been proposed for the task of generating sentences that describe the content of a video[2][3]. However, these methods require a lot of computational resources to construct the system and also take a long time to generate sentences because they use the whole video frame. Our system aims to generate caption with high precision and at the same time significantly reduces the number of frames used for the processing. The architecture of our proposed system consists of three steps: keyframe extraction, caption generation, and caption aggregation.

2 Approach

Our approach focuses on reducing processing time by using only the keyframes. Figure 1 shows a Video To Text framework using keyframes.

Shibata proposed the Average Hash method for keyframe extraction, the NIC (Neural Image Captioning) model for caption generation, and the LSTM (Long Short Term Memory) method for the aggregation of caption[4]. Although this method achieves a certain level of accuracy while reducing the number of processing frames, it has some problems, such as the low accuracy of keyframe detection and the use of a simple LSTM, which causes the sentences obtained by the NIC model to break down grammatically in the caption aggregation step. In this study, we attempt to solve these problems by employing a more accurate keyframe extraction method and an aggregation method that maintains a correct grammar. At the keyframe extraction step, a technique used for video summarization is employed to extract the keyframes from the video. Next, in the caption generation step, an explanation sentence is generated by using a model combining CNN (Convolutional Neural Network) and LSTM for each of the extracted keyframes. And finally, in the caption aggregation step, a single sentence is output from multiple captions by applying a technique that is used in a text summarization task.

2.1 Keyframe Extraction

To generate sentences with high accuracy while reducing the number of processing frames, it is necessary to select

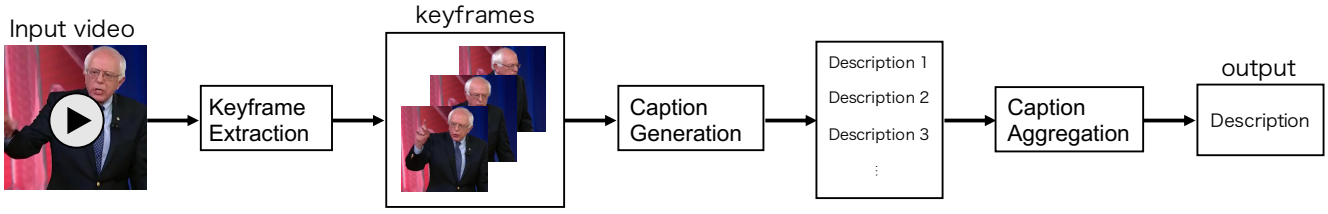

Figure 1: Video To Text framework using keyframes

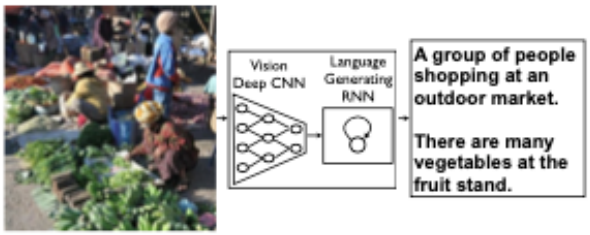

Figure 2: NIC model outline[8]

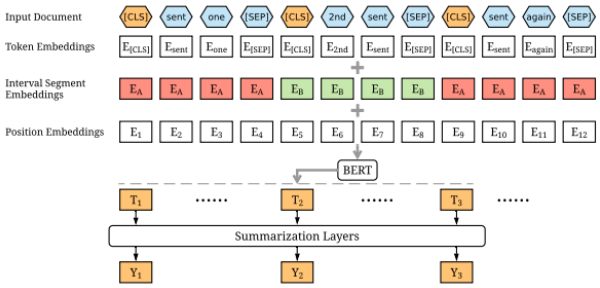

Figure 3: BERTSUM overview[10]

important frames in the video. In our method, keyframes are defined as the frames that characterize the video, and only the keyframes are used to generate captions to reduce the computational cost. KTS (Kernel Temporal Segmentation)[5] is used to extract the keyframes. KTS is one of the video segmentation methods, in which the frames whose image feature values change significantly are used as scene boundaries. In the proposed system, the first and last frames of the video are used as keyframes in addition to the three frames with the largest change in features. The image features were extracted by using a GoogLeNet[6]. The ImageNet[7] dataset was used for the pre-training of GoogLeNet.

2.2 Caption Generation

We use the NIC model[8] to generate explanatory text from the extracted keyframes. Figure 2 shows th NIC model outline. NIC model is a deep learning model, which consists of an encoder and a decoder. The encoder uses a CNN to extract feature vectors from the input images and feeds them to the decoder. Afterward, the Decoder generates the feature vectors one-by-one using the LSTM and combines them into a sentence. MS COCO[9] dataset was used for pretraining the NIC model.

2.3 Caption Aggregation

Since a requirement of the VTT is to output a single sentence, it is necessary to aggregate the sentence output for each keyframe from the NIC model. We explored whether the technique of document summarization could be applied to sentence aggregation. The methods used in the document summarization task can be classified into two main types: extractive and abstractive. In the proposed method, we adopt the extractive summarization method, which is considered to be relatively computationally inexpensive. We compared the performance of two extractive methods, BERTSUM[10] and LexRank[11].

BERTSUM is a sentence summarization model using BERT[12], which is a general-purpose language model, as a pre-training model. Figure 3 shows an overview of BERTSUM. While normal BERT places a [CLS] token only at the beginning of the input, BERTSUM places a [CLS] token at the beginning of each sentence in the input and uses the [CLS] token to delimit the sentence. In the BERTSUM model, the output vector of the top layer of the encoder for each [CLS] token is input to the sigmoid classifier to determine whether each sentence is a summary sentence.

LexRank is an extractive summarization method inspired by PageRank[13], which is used in Google's search engine. LexRank generates a ranking by constructing an undirected graph based on the similarity between sentences in the input document and calculating the importance of each sentence. Based on that ranking, a summary statement will be determined.

3 Experimental Results

3.1 Validation

To confirm that the proposed method works without any problems, we validated it using a portion of the VTT2019 dataset (300 videos). Table 1 shows the results of the validation scores using VTT2019 data. Table 1 lists the METEOR scores when using BERTSUM and LexRank as the aggregation method.

Table 1: Validation scores (VTT2019 data)

Aggregation Method	METEOR
BERTSUM	0.235
LexRank	0.218

Table 2 shows a comparison of the average number of frames per video when all the frames are used in the processing and our proposed method in validation.

Table 2: Average frames per video (VTT2019 data)

	use whole frame	our run
Number of frames	267	5

As a result of the verification, the system worked fine. From Table 1, there was no significant difference in scores between BERTSUM and LexRank. Table 2 also indicates

that a certain amount of METEOR scores could be obtained even when the frames used for processing were greatly reduced.

3.2 Evaluation at VTT2020

We tested our models using VTT2020 dataset and calculated scores. Table 3 shows the METEOR, BLEU and CIDEr scores for each run of our model.

Table 3: Test scores (VTT2020 data)

Run	METEOR	BLEU	CIDEr
run1.bsum.primary	0.195	0.009	0.137
run2.lex065	0.210	0.008	0.137

Table 4 shows the results of comparing the scores of the proposed method with those of the other participating teams in VTT2020. Our team name is KsLab_NUT. It shows the results for the primary run among the results submitted by each team. Our team submitted the method with BERTSUM as the primary run.

Table 4: Scores by VTT2020 participating teams

Team name	METEOR	BLEU	CIDEr
RUC_AIM3	0.310	0.056	0.538
PicSOM	0.262	0.053	0.319
MMCUniAugsburg	0.202	0.011	0.140
KsLab_NUT	0.195	0.009	0.137
IMFD_IMPRESEE	0.194	0.007	0.087
KU_ISPL	0.191	0.018	0.074

Table 5 shows a comparison of the average number of frames per video when all the frames are used in the processing and our submitted run in VTT2020 data.

Table 5: Average frames per video (VTT2020 data)

	use whole frame	our run
Number of frames	147	5

4 Discussion

Although the proposed method was able to reduce the number of processing frames, the score was slightly lower than that of the other teams. Also, there was no significant difference in the scores between BERTSUM and LexRank in VTT2020 datasets. A possible reason for the low scores is that the sentences generated by the NIC model are extracted unaltered, which may result in the omission of important words. BERTSUM and LexRank are both extractive summarization methods, which extract one sentence from the original text without altering it. Since the text output by the NIC model is used as it is, grammatical errors are prevented, but the information contained in the text other than the extracted text is lost. Therefore, if the sentences that the extraction method deems unimportant contain expressions that characterize the video, the loss of this information could result in lower scores on METEOR and BLEU. In addition, since our method only uses feature change as an indicator to discriminate key frames, we may not get good scores for videos with little change or extremely short videos.

In the future, we plan to revisit the methods used in the caption aggregation step. One suggestion is to apply the abstract summarization method. Abstractive summarization is a method to generate new sentences by using the elements included in the target instead of extracting one sentence from the target. By applying this method, we can expect to generate new sentences using the words in each caption generated by the NIC model to avoid missing the information that characterizes the video.

5 Conclusion

By using only the keyframes of the video for processing, we were able to significantly reduce the number of processing frames while maintaining a certain degree of accuracy. For further improvements in accuracy, possible approaches include revising KTS parameters to allow for more flexibility in scene change detection, using abstract methods (e.g., Document generation using LSTM) to aggregate captions into a single caption rather than extracting a single sentence,and modifying the dataset used to train the NIC model. In addition, it is necessary to examine and improve what factors are contributing to the decrease in scores in the three steps to generation.

References

[1] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth Jones, Wessel Kraaij, and Georges Quénot. Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *Proceedings of TRECVID 2020*. NIST, USA, 2020.

[2] A. Liu, Y. Qiu, N. Xu, Y. Su, Y. Wong, and M. S. Kankanhalli. Tianjin university and national university of singapore at trecvid 2017: Video to text description. In *TRECVID 2017 VTT Task paper*, 2017.

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[4] Akira Shibata and Takashi Yukawa. An automatic text generation system for video clips using machine learning technique. In *TRECVID 2018 VTT Task paper*. Nagaoka University of Technology, 2018.

[5] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV - European Conference on Computer Vision*, volume 8694 of *Lecture Notes in Computer Science*, pages 540–555, Zurich, Switzerland, September 2014. Springer.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[8] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator, 2015.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[10] Yang Liu. Fine-tune bert for extractive summarization, 2019.

[11] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22, Dec 2004.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.