Journal of Advanced College of Engineering and Management, Vol. 9, 2024

VIDEO CAPTIONING IN NEPALI USING ENCODER DECODER

Kabita Parajuli,[1], Shashidhar R Joshi,[2]

Department of Electronics and Computer Engineering
Pulchowk Campus, Tribhuvan University Lalitpur, Nepal
[1]parajulikabita10@gmail.com, [2]srjoshi@ioe.edu.np

Abstract

Video captioning is a challenging task as it requires accurately transforming visual understanding into natural language descriptions. This challenge is further compounded when dealing with Nepali, due to the lack of existing academic work in this domain. This study develops an encoder-decoder paradigm for Nepali video captioning to address this difficulty. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) sequence-to-sequence models are utilized to produce relevant textual descriptions based on features extracted from video frames using Convolutional Neural Networks (CNNs). Additionally, a Nepali video captioning dataset is created by adapting the Microsoft Research Video Description Corpus (MSVD) datasets through Google Translate, followed by manual post-editing. The efficiency of the model for video captioning in Nepali is demonstrated using BLEU, METEOR, and ROUGE metrics to assess its performance.

Keywords: *MSVD, Encoder, Decoder LSTM, GRU*

1 Introduction

The increasing availability of multimedia data, particularly videos, has brought numerous advantages but also posed challenges in organizing and accessing the vast amount of visual information. The abundance of online videos has made video captioning a significant area of research. Effectively organizing, indexing, and retrieving videos is crucial for managing and understanding this massive volume of visual data. The growing popularity of video-sharing websites has intensified the need for accurate and efficient methods of video comprehension.

Video captioning, the process of automatically generating a natural language description of a video, is inherently challenging due to the dynamic and complex nature of videos. Unlike static images, videos contain a temporal component, varying over time, which complicates the extraction of the necessary temporal and spatial information to produce meaningful and accurate captions. Deep learning-based techniques have recently set the state-of-the-art in video captioning. These methods typically involve extracting visual features from videos using convolutional neural networks (CNNs) and generating captions based on these features using either transformer models or recurrent neural networks (RNNs). CNNs, with their ability to learn intricate spatial patterns, are well-suited for extracting visual data from videos. RNNs, capable of capturing temporal connections between words, are ideal for generating captions. Transformer models have shown promising performance in video captioning due to their attention mechanism, which efficiently understands long-range dependencies.

Recurrent neural network architectures, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are frequently employed for video captioning, each with its advantages and disadvantages. LSTM excels at capturing long-term dependencies in data, making it a strong choice for tasks requiring long-term memory, including video captioning. However, this increased complexity comes with higher computational overhead. In contrast, GRU has a simpler structure, which speeds up training and inference, thereby enhancing processing efficiency. However, its simplicity may compromise its effectiveness in modeling long-term dependencies.

2. LITERATURE REVIEW

L. Yan et.al.[1] introduced a technique for creating descriptive and contextually relevant video subtitles by combining global and local representations. Their approach captures high-level knowledge by extracting features from the entire video sequence for global representation, while the second stream focuses on local representations by identifying regions within video frames. These global and local features are then fed into a caption generator using transformer-based architectures or recurrent neural networks. Their experiments show that this method outperforms current approaches in terms of the relevance and quality of captions.

Zhiwen Yan et.al.[2] introduced an innovative video captioning framework called Object Relation and Multimodal Feature Fusion (ORMF). ORMF employs a Graph Convolution Network (GCN) to encode object relationships, creating a graph of object features based on their spatiotemporal correlations and similarities within the video. Additionally, ORMF constructs a multimodal feature fusion network to integrate features from various modalities, enhancing the richness of the captions produced. The effectiveness of this approach is validated by experimental results on two publicly available datasets, Microsoft Research Video to Text (MSR-VTT).

Sandeep Samleti et.al.[4] developed a video captioning system that utilizes a CNN for extracting frame-level features and an LSTM for sequence synthesis. They represent the entire video using a mean-pooled vector of all extracted features. However, this method falls short in capturing temporal correlations between frames due to the mean pooling approach. Building on this, Kevin Lin et.al.[3] introduced a two-layer LSTM encoder-decoder model for video captioning, where each frame is used to construct a fixed-size feature vector comprising visual features at each time step.

The study "SBAT: Video Captioning with Sparse Boundary-Aware Transformer" by Tao Jin et.al.[7] introduces a novel approach for video captioning using the Sparse Boundary-Aware Transformer (SBAT). Unlike the vanilla transformer, which is suited for unimodal tasks like machine translation, SBAT addresses the multimodal challenges of video data by eliminating redundant information. It selects diverse features from various contexts and applies a boundary-aware pooling technique to multi-head attention scores. To mitigate local information loss from sparse operations, SBAT employs a local correlation strategy. Tested on the MSVD dataset, SBAT surpasses models like TVT, MARN, GRUEVE, SCN, and POS-CG. This approach significantly enhances the accuracy and relevance of video captions, effectively handling the complexity of video data.

Encoder-decoder frameworks, which incorporate convolutional neural networks (CNNs) and various adaptations of recurrent neural networks, are currently the predominant approaches for video captioning. A notable method employs CNNs to extract frame-level features, which are then aggregated into a mean-pooled vector representing the entire video. This vector is subsequently input into a Long Short-Term Memory (LSTM) network to generate sequences. However, the limitation of this approach lies in the inability of mean pooling to capture temporal correlations between frames. To address this, an enhanced encoder-decoder system utilizing two layers of LSTMs has been proposed. This improved framework encodes the visual features of a video into a fixed-size feature vector by processing each frame as input at each time step, thereby better capturing the temporal dynamics inherent in video data. LSTM and GRU models are highly effective for video captioning tasks due to their ability to handle sequential data and long-term dependencies. Prior work has been completed by concentrating on important languages like Chinese, English, German, and Hindi. This inspired me to use publicly available MSVD dataset, which is translated into Nepali language, followed by post editing of each reference caption, this research attempt to address the problem of video captioning in NEPALI language.

2.1 Long Short-Term Memory Architecture

An artificial recurrent neural network architecture called long short-term memory combines feedforward and feedback neural networks. Long-term reliance is resolved by LSTMs because of their unique architecture, which allows them to manage the relationship between recent past knowledge and current tasks even as the gap widens. Information moves across cells in the LSTM structure, a type of memory system that can selectively distinguish between information that should be remembered and information that should be spread. Information about sequential data processing, including speech, video, text, etc., can be carried by the cell state**.**

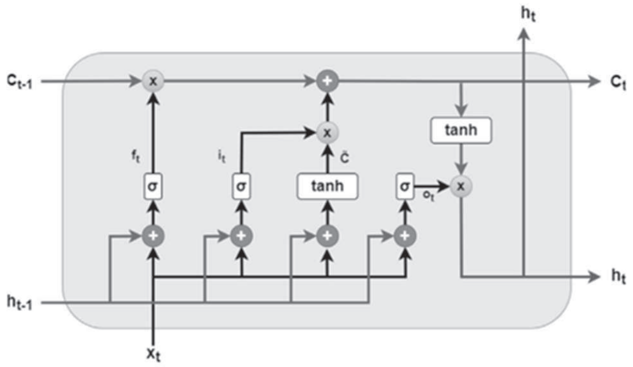

Figure 2.1: Basic Architecture of LSTM [22]

The input gate (xt), output gate (ht), and forget gate (ft) compose a cell state. An input gate is used in an LSTM cell to measure the importance of new information at a certain timestamp. The crucial element in charge of eliminating (forgetting) the data from the prior timestamp is a forget gate. Additionally, the output gate's goal is to choose the most important data from the active LSTM cell and send it out as the output.

$$i_t = \mathrm{x_i}U^i + h_{t-1}W^i \qquad (1)$$

$$f_t = (x_tU^f + h_{t-1}W^f) \qquad (2)$$

$$o_t = \ x_tU^o + h_{t-1}W^o \qquad (3)$$

$$C_t' = \tanh\ (\ x_tU^g + h_{t-1}W^g). \qquad (4)$$

$$C_t = \tanh\ (\ x_tU^g + h_{t-1}W^g). \qquad (5)$$

$$h_t \ = \ tanh(C_t) * o_t \qquad (6)$$

Equations 1 through 6 illustrated by Zaoad et al, 2022 [22], shows the LSTM architecture's mathematical formulas for corresponding gates. The reason LSTM is used in this study is due to its ability to retain patterns for extended periods with selectivity. Additionally, the LSTM design makes it easy to categorize, process, and forecast the right result from massive time-series data.

2.2 Gated Recurrent Unit

A basic recurrent neural network with a gating mechanism added is called a gated recurrent unit (GRU). Just like in LSTM, gates are used to regulate the information flow in GRU. It can train more quickly and has a simpler architecture than LSTM, with fewer parameters. The fundamental architecture of a single GRU unit is illustrated by Zaoad et al., 2022 is shown in figure, which includes an update gate (zt), reset gate (rt), current memory content (ht), and final memory at the current time step (ht). The GRU mathematical formulas are presented:

$$z_t = W_z x x_t + W_z h h_t + b_z \tag{7}$$

$$r_t =(W_r x x_t + W_r h h_{t-1} + b_r \tag{8}$$

$$h'_t = tanh(W_{\sigma x} x_t + W_{\sigma h} h_{t-1} + b_\sigma) \tag{9}$$

$$h_t = z_t h_{t-1} + (1 - z_t) h'_t \tag{10}$$

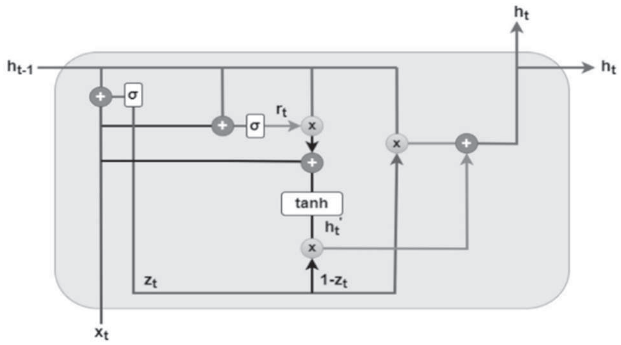

Figure 2.2: Basic Architecture of GRU [22]

2.3 Encoder

It is the objective of the encoder to analyze and understand the input sequence. The encoder examines the input, whether it’s a sentence in a natural language or a series of video frames, step by step, analyzing and extracting pertinent information at each time step. Each word in the phrase or each token in the input sequence is analyzed sequentially in text-based tasks. At each time step, the LSTM or GRU units of the encoder receive inputs and change their internal hidden states. These concealed states extract crucial data from the input sequence. The context vector, often referred to as the thought vector, is the encoder’s final hidden state and contains a compressed version of the whole input sequence. It’s important to note that in many applications, just the internal states are kept, and the encoder’s output is deleted.

Depending on the recurrent neural network (RNN) being used, the encoder cell architecture can change. The internal state configuration is one of the main characteristics that set LSTM (Long Short-Term Memory) apart from GRU (Gated Recurrent Unit). Each cell in an LSTM consists of two internal states: the hidden state, which contains information transferred from one time step to the next, and the cell state, which oversees managing long-term data. On the other hand, the architecture of GRU cells is simpler because they have just one hidden state.

Furthermore, an important change from conventional methods is seen when utilizing an attention mechanism. This technique considers the outputs from all 80 encoder states, rather than just the last step. This change in focus improves the model's ability to extract and apply pertinent information for the job at hand by enabling it to dynamically attend to different segments of the input sequence. The effciency of sequence-to-sequence models is greatly influenced by this architecture and design, especially in natural language processing and other fields where understanding and context are important.

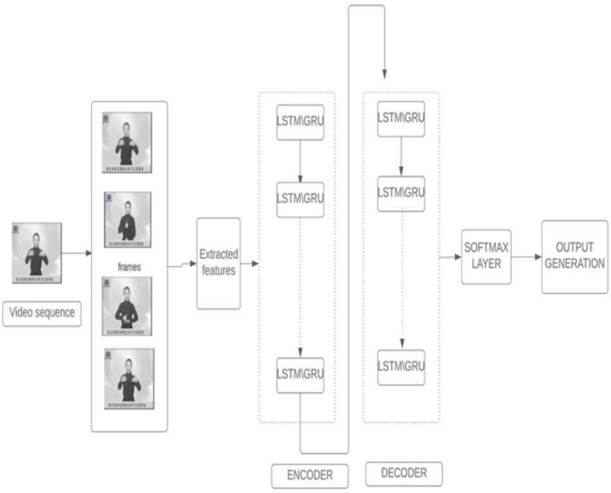

Figure 2.3: Basic Architecture of Encoder Decoder

2.4 Decoder

The output sequence is created by the decoder, which succeeds the encoder. The context vector from the encoder is used to initialize its initial LSTM or GRU cell. Making the output sequence step by step is the decoder's main responsibility. The decoder generates words one at a time for tasks involving text production, such as language translation or text summarization. It produces insightful captions for each frame of the video when used for captioning. The context vector is used to set the initial hidden state of the decoder's LSTM or GRU cell, which serves as the starting point for producing the output. The decoder creates an output token (such as a word) at every time step based on the previous output and the concealed state it is now in.

3 METHODOLOGY

An encoder-decoder paradigm for video captioning is a strong, adaptable technology with significant applications. It serves as an example of how AI has the potential to close gaps across various media types (visual and textual) and to produce more inclusive and educational digital experiences. It presents fascinating opportunities for enhancing how we interact with and comprehend video information in our increasingly digital world as the field continues to develop. A fascinating and interdisciplinary field of research that combines computer vision and natural language processing is video captioning utilizing an encoder-decoder architecture. This strategy has a wide range of real-world uses and is receiving a lot of attention since it has the potential to make videos easier to access, find, and use. It brings up new possibilities by allowing machines to comprehend and characterize the content of videos.

3.1 Dataset

This study utilized the MSVD dataset, developed by the University of Texas at Austin in collaboration with Microsoft Research's Natural Language Processing Group. This extensive dataset includes nearly 2,000 brief YouTube video clips, each ranging from 10 to 20 seconds in length and depicting a wide variety of activities and subjects. Comprising 1,970 distinct videos, the dataset is enriched with over 80,789 text descriptions, averaging around 40 captions per video, though the exact number of captions varies. Each text description includes a video ID and an English caption, which were carefully processed using Google Translate and manual post-editing for the study's purposes. The MSVD dataset stands out for its rich annotations and diverse content, making it a valuable resource for video captioning tasks.

3.2 Preprocessing

Initially, English captions were translated into Nepali using Google Translate, with subsequent manual editing to rectify errors, particularly in lengthy or ambiguous captions. Each translated Nepali caption was tokenized using a specialized Nepali tokenizer. To standardize the captions, a "start of sentence" token was added at the beginning and an "end of sentence" token at the end. The dataset was then systematically divided into training, validation, and testing sets to facilitate accurate model evaluation. A word list derived from the training captions enabled effective tokenization, converting textual data into numerical form necessary for machine learning algorithms. Captions were padded to a uniform maximum length of 10 words, thereby avoiding excessive padding and potential complications in model performance. This comprehensive preprocessing approach ensured that both textual and video data conformed to the required input specifications for the study.

3.3 Feature Extraction

The MSVD (Microsoft Research Video Description) dataset feature extraction for video captioning entails gathering both visual and temporal data from the video frames to provide meaningful captions. Frames from videos are extracted using the VGG16 model. To extract frames from a given video file frames function requires parameters like the video path and the desired number of extracted frames. The feature reads frames, distributing them uniformly throughout the film to guarantee that representative frames are recorded. a thorough pipeline for video preprocessing that pulls frames from a dataset of films, stores them as NumPy arrays, and stores the captions that go with them.

Extracting frames from Video

Frames from videos are extracted during the pre-processing stage, and any necessary adjustments are made. To extract frames from a given video file, use the extract frame's function. It requires parameters like the video path and the desired number of extracted frames. The feature reads frames, distributing them uniformly throughout the film to guarantee that representative frames are recorded. a thorough pipeline for video preprocessing that pulls frames from a dataset of films, stores them as NumPy arrays and stores the captions that go with them.

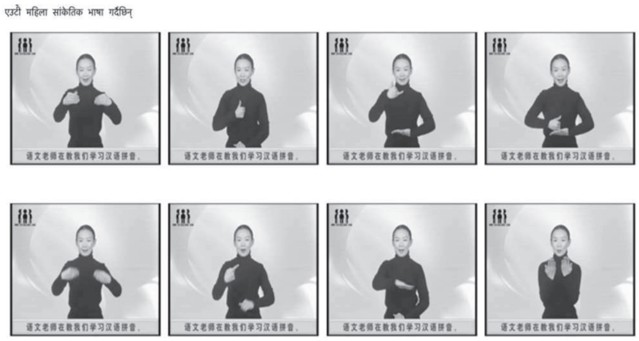
एउटी महिला सांकेतिक भाषा गर्दैछिन्
语文老师在教我们学习汉语拼音。

Fig 3.3: Extracted Frames from Video

The success of video caption creation is attributed to the pre-processing step that guarantees the video data is in an appropriate format for subsequent machine learning and deep learning activities. If the captioning model generates the same caption, then the model performance increases.

Features from Extracted frames

The pre-trained model performs feature extraction on each chosen frame. The use of a model like VGG16, which has been pre-trained on a sizable dataset including millions of images, produces a vector of features, frequently of dimensions 4096, that reflects the key visual qualities of each frame as it is processed by the model. Following the stacking of these feature vectors for each film, a structured NumPy array with dimensions (28, 4096) is produced.

3.4 Model Training

For various goals, the study on video captioning calls for several machine-learning architectures. There are various uses for each machine learning and deep learning model. Encoder and decoder models with LSTM and GRU, which aid in training a series of frames, are needed for training. A train set, a validation set, and a test set have been created from the preprocessed data. To the training model, the features data is sent. Where the LSTM performs best for sequence data as a reservoir, an encoder-decoder model with an LSTM and GRU is employed. The encoder's final output is supplied as input to the decoder model, which creates captions. Under this model, the training set is trained.

The suggested video captioning models are trained and assessed using the updated dataset. A split ratio of 85% is employed for training and validation, with 1450 and 100 video clips used for testing, respectively. A neural network model with the following parameters was employed: a latent dimension of 512, an encoder with 28-time steps, and 2048 unique tokens. The decoder was intended to produce output sequences consisting of ten-time steps and a 1500 token output vocabulary. The model was trained over 40 epochs using a batch size of 320 in our training procedure. The model's architecture and training schedule were shaped by these parameter values, which in turn affected the model's performance within the framework of study.

3.5 Evaluation metrics

There are four distinct automated evaluation measures that are employed: BLEU , METEOR, CIDEr , and ROUGE. Several research reports assert that because videos have a dynamic structure, different content pieces, events, and activities, the BLEU score may vary from the human assessment of the generated captions. However, after noting BLEU's recent success in MT output evaluation, we employed it in addition to METEOR for caption evaluation.

4 RESULT AND ANALYSIS:

While there has been a lot of research on Nepali picture captioning, there hasn’t been nearly as much done around video captioning for the language. Videos, which consist of a continuous series of images, can be related to images. The suggested models’ performances are compared as well with the results of research done on Nepali image captioning. The performance of various models compared to the suggested ones is shown in the figure below using various evaluation metrics

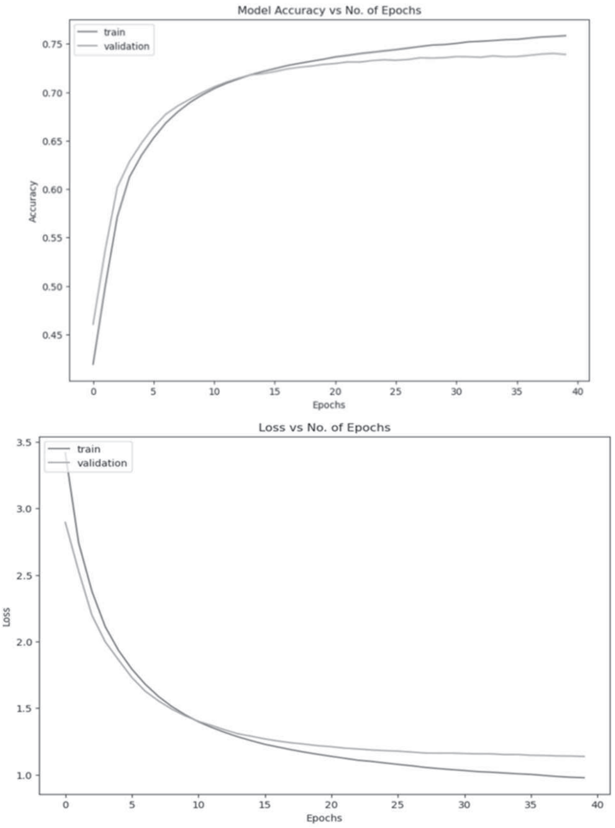

Fig 4.1: Accuracy loss plot encoder decoder with GRU

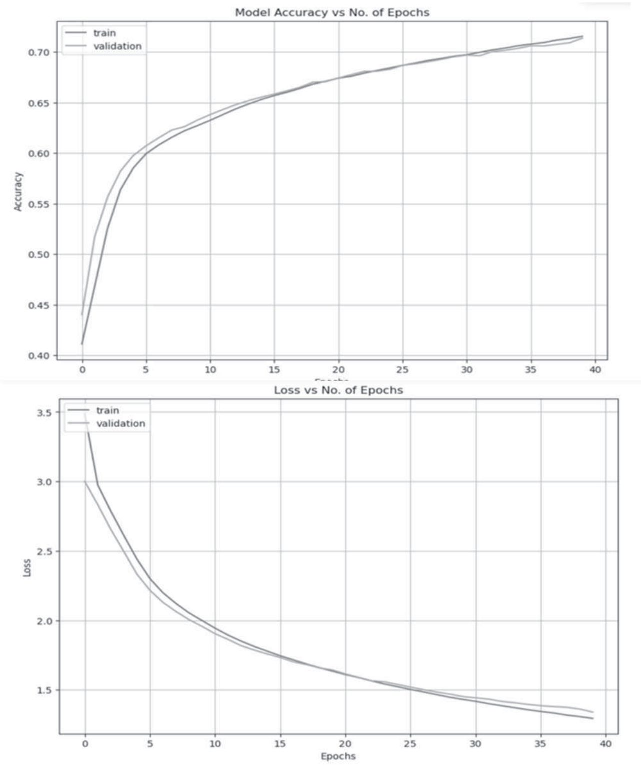

Fig 4.2: Accuracy loss plot of Encoder decoder LSTM

Table 4.3: Performance Evaluation on Greedy search

RNN	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE_L
LSTM	0.58	0.421	0.282	0.169	0.361	0.248
GRU	0.66	0.479	0.329	0.189	0.361	0.248

5 Conclusion

This study significantly advances video captioning by developing a syntactically and semantically consistent dataset from the MSVD. Among the evaluated models, the GRU outperformed others, achieving high scores in BLEU, METEOR, and ROUGE metrics, while the LSTM also demonstrated competitive performance. These findings underscore the effectiveness of GRU-based models in capturing temporal dynamics and contextual dependencies in video data, offering a robust solution for video captioning tasks. Furthermore, this research can be extended to the MSR-VTT dataset, a large-scale benchmark for video captioning that contains 10,000 video clips with diverse content and multiple reference captions. Extending the study to the MSR-VTT dataset will validate and enhance the model's generalizability, paving the way for broader applications in video captioning.

References

[1] L. Yan et al., "Video Captioning Using Global-Local Representation," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 10, pp. 6642-6656, Oct. 2022, doi:10.1109/TCSVT.2022.3177320

[2] Yan, Zhiwen, et al. "Multimodal feature fusion based on object relation for video captioning." CAAI Transactions on Intelligence Technology 8.1 (2023):247-259. https://dx.doi.org/10.1049/cit2.12071

[3] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17949-17958, 2022.

[4] Sandeep Samleti , Ashish Mishra , Alok Jhajhria , Shivam Kumar Rai, Gaurav Malik, 2021, Real Time Video Captioning Using Deep Learning, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) Volume 10, Issue 12 (December 2021)

[5] Im H, Choi Y-S. UAT: Universal Attention Transformer for Video Captioning. Sensors. 2022; 22(13):4817. https://doi.org/10.3390/s22134817

[6] P. Song, D. Guo, J. Cheng and M. Wang, "Contextual Attention Network for Emotional Video Captioning," in IEEE Transactions on Multimedia, 2022, doi: 10.1109/TMM.2022.3183402.

[7] Jin, T., Huang, S., Chen, M., Li, Y., Zhang, Z. (2020). SBAT: Video Captioning with Sparse Boundary-Aware Transformer. ArXiv, abs/2007.11888.https://doi.org/10.48550/arXiv.2007.11888

[8] W. Pei, J. Zhang, X. Wang, L. Ke, X. Shen, and Y.-W. Tai, "Memory- attended recurrent network for video captioning," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8339-8348, 2019.

[9] D. Liu, Y. Cui, Y. Chen, J. Zhang, and B. Fan, "Video object detection for autonomous driving: Motion-aid feature calibration," Neurocomputing, vol. 409, pp. 1-11, 2020

[10] N. Aafaq, N. Akhtar, W. Liu, S. Z. Gilani and A. Mian, "Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning", Proc. IEEE/CVF Conf. Computer. Vision. Pattern Recognition. (CVPR), pp. 12487- 12496, Jun. 2019.

[11] Y. Yang et al., "Video Captioning by Adversarial LSTM," in IEEE Trans- actions on Image Processing, vol. 27, no. 11, pp. 5600-5611, Nov. 2018, doi: 10.1109/TIP.2018.2855422.

[12] L. Yan et al., "Video Captioning Using Global-Local Representation," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 10, pp. 6642-6656, Oct. 2022, doi: 10.1109/TCSVT.2022.3177320

[13] P. Song, D. Guo, J. Cheng and M. Wang, "Contextual Attention Network for Emotional Video Captioning," in IEEE Transactions on Multimedia, 2022, doi: 10.1109/TMM.2022.3183402.

[14] Q. Zheng, C. Wang, and D. Tao, "Syntax-aware action targeting for video captioning," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13 093-13 102, 2020.

[15] Dhir, R., Mishra, S.K., Saha, S., Bhattacharyya, P.: A deep attention-based framework for image caption generation in Hindi lan- guage. Computaci ÃÅon y Sistemas 23(3) (2019)

[16] S. Li, Z. Tao, K. Li and Y. Fu, "Visual to Text: Survey of Image and Video Captioning," in IEEE Transactions on Emerging Topics in Computational Intel- ligence, vol. 3, no. 4, pp. 297-312, Aug. 2019, doi: 10.1109/TETCI.2019.2892755.

[17] Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., Saenko, K.: Translating videos to natural language using deep recurrent neural networks. arXiv:1412.4729 (2014)

[18] Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence-video to text. In: Proceed- ings of the IEEE international conference on computer vision, pp. 4534–4542 (2015)

[19] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational lin- guistics, pp. 311–318. Association for Computational Linguistics (2002)

[20] Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consen- sus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4566–4575 (2015)

[21] Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out, pp. 74–81 (2004)

[22] Zaoad, M. S., Mannan, M. R., Mandol, A. B., Rahman, M., Islam, M. A., Rahman, M. M. (2022). An attention-based hybrid deep learning approach for bengali video captioning. Journal of King Saud University - Computer and Information Sciences, 35(1), 257-269. https://doi.org/10.1016/j.jksuci.2022.11.015

[23] Y. Yang et al., ”Video Captioning by Adversarial LSTM,” in IEEE Trans-actions on Image Processing, vol. 27, no. 11, pp. 5600-5611, Nov. 2018, doi: 10.1109/TIP.2018.2855422.