Video Summarization

Ryan Rowe
University of Washington

Joseph Zhong
University of Washington

Preston Jiang
University of Washington

{rfrowe, josephz, prestonj} @cs.washington.edu

Abstract

The use of recurrent neural networks with attention for processing time-sequential information, from audio, natural language processing, signal processing, as well as video processing has gained significant recent popularity, as this framework can be generalized to generate sequential outputs, such as captions. In this paper, we propose a neural-based encoder-decoder method inspired with recent innovations incorporating attention mechanisms to re-weight the importance of encoded features across time. We deviate from this framework, instead, approaching discontinuities in input video contexts by directly detecting temporal boundaries in the encoding recurrent network.

We provide our source code repository below[1].

1 Introduction

Every minute over 300 hours of video content are uploaded to YouTube. This works out to nearly 50 years of new content each day. The sheer amount of video data available-thanks to the advent of the Internet-on YouTube, Vimeo, Twitter, and hundreds of other websites means it is impossible for human eyes to watch each frame. In order to quickly review massive amounts of video content, especially for long, multi-contextual videos such as surveillance videos, vlogs, or travel videos, one must first summarize each video.

In general, automatically describing video through human natural language is an important task in computer vision and machine learning: a similar and tangential task called video captioning extends towards a significant field of applications.

Many such video summarizers exist for various purposes. For example, Descriptive Video Service (DVS) (Dvs, 2019) produces video descriptions for the purpose of making visual media such as television or films more accessible to people of low-vision or other visual impairment (Adp, 2019).

Analogous to video summarization, the tangentially related problem of captioning has been applied to static images, or static visual captioning (Vinyals et al., 2014; Karpathy and Li, 2014; Xu et al., 2015; Vinyals et al., 2016). Such techniques in image captioning have provided a fundamental basis for approaching describing visual content in terms of natural language. As a result, this approach has been extended to deal with the temporal progression of video frames (Yao et al., 2015; Sutskever et al., 2014; Yu et al., 2015).

Overall, past methods for video captioning and summarization have primarily applied recurrent neural networks (Sherstinsky, 2018; Salehinejad et al., 2018; Lipton, 2015) or long short-term memory neural architectures (Hochreiter and Schmidhuber, 1997; Olah, 2015) which have been shown to be effective models for capturing the temporal relationship between sequences of information. Naively applied, recurrent neural networks and long short-term memory architectures can be effective for modelling the sequential nature in time series data. However in the case of video captioning, and in particular, complex edited videos with multiple segmented points of short scenes, can have high variance in visual appearance, while still maintaining contextual consistency (Baraldi et al., 2016).

Thus, in this paper we propose an architecture inspired by the work Baraldi et al, (Baraldi et al., 2016) to use an attention-mechanism inspired module to encode scene discontinuities in the video sequence. We report experimentation on the Microsoft Video-to-Text (MSRVTT) dataset (see Figure 2) (Brad and Rebedea, 2017) contain-

[1] https://github.com/joseph-zhong/VideoSummarization/

ing over 200K video clip sentence-pairs.

We organize our results as follows: we start our discussion with an overview of previous methods in § 2. In § 3, we discuss our overall pipeline and data representation. In § 4, we discuss the details of our architecture and the different neural models involved. In § 5, we summarize our experiment results, and finally in § 6 we summarize weaknesses of our approach and future work in video summarization.

2 Related Works

Early video captioning methods have entailed visual feature extraction to condition the natural language captioning output in terms of subjects, objects, and relational actions (Guadarrama et al., 2013; Krishnamoorthy et al., 2013; Thomason et al., 2014). As an baseline approach, these methods capture the basic, high-level content of an image, but cannot generalize to complex sentences or unseen data outside the training dataset. Thus, this motivates the application of more generalized decoders through recurrent neural networks to better satisfy the rich natural language inherent in describing and captioning complex video scenes (Sherstinsky, 2018; Salehinejad et al., 2018; Lipton, 2015; Hochreiter and Schmidhuber, 1997; Olah, 2015)

One early approach in video captioning by Venugopalan et al. (2014) motivated the use of recurrent neural networks to encode the sequence of low-dimensional convolutional neural network features extracted from single video frames, fed into a single LSTM layer. While effective in generally encoding the visual content of the video frames, this method did not effectively encode the sequential nature of the video input, framing video captioning data into a static image captioning problem. Future work followed by encoding the sequential transition using the *sequence-to-sequence* approach in recurrent neural architecture, encoding the sequential transitions in video through a stacked LSTM architecture where subsequent layers of LSTM cells were first conditioned by the outputs of the previous LSTM layers, inspired from work in machine translation (Sutskever et al., 2014).

As this general framework gained popularity, subsequent approaches incorporated attention mechanisms in the sentence decoding (Yao et al., 2015), or building visual-language semantic embeddings (Pan et al., 2015b), or encoding external prior knowledge from language models (Venugopalan et al., 2016).

More recently, research in video captioning has primarily focused on taking the existing input representation and attempting to better exploit the internal representation throughout the encoder-decoder neural architecture. Yu et al. (2015) proposed a hierarchal recurrent neural architecture, where in the sentence decoding, they introduced a sentence decoder as well as a paragraph generator using a gated recurrent unit (GRU) layer (Cho et al., 2014) while conditioning on contextual information generated by the sentence generator, combined with the encoded video features.

In contrast, Pan et al. (2015a) focused on video encoding, also using a hierarchical approach to the recurrent video encoder. Similar to a sliding-window approach, a LSTM is applied to overlapping video chunks of varying scales and granularities.

In this paper, we take inspiration from Baraldi et al. (2016) to combine the hierarchical architectures in both the encoder and decoder, while also incorporating an attention-mechanism inspired boundary detection module in the video encoder. The leveraging of segment-level features has been previously investigated in natural language processing (Chung et al., 2016), action recognition (Tang et al., 2012; Song et al., 2013; Pirsiavash and Ramanan, 2014; Lan et al., 2015) and event detection context (Wang et al., 2016). This technique was first introduced to video captioning by Baraldi et al. (2016). We referenced the implementations of the boundary detector from the Yugnaynehc (2017) repository.

3 Approach

The overall architecture of our processing pipeline is described in Figure 1. We uniformly sampled thirty frames for every video (to ensure this, we changed the sampling frequency with respect to each video's frame rate) to downsample the amount of repetitive frames in high refresh rate videos. First, for every video frame sampled, we used a pre-trained ResNet 50 He et al. (2015) network as the visual feature extractor and represented each video frame as the output of its last fully connected layer. Thus, we have out input $\mathbf{X}$ for each forward pass of our pipeline as

$$\mathbf{X} \in \mathbb{R}^{N \times T \times p}$$

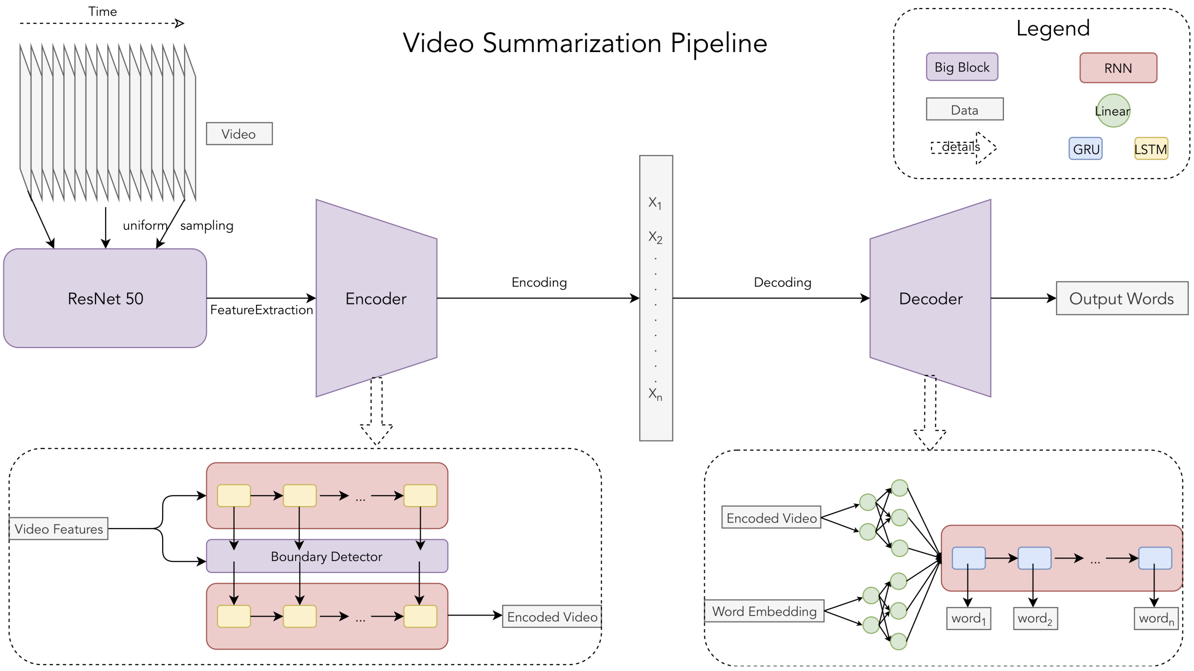

Figure 1: The video summarization pipeline. Video frames are uniformly sampled. ResNet50 (He et al., 2015) is used to extract visual information of the frames. An Encoder-Decoder framework is used to generate text summaries from frames (see text for details).

1. race cars are lined up in a track filled with cars
2. several race cars are lined up in two long rows about to race
3. indy cars are lined up at the start of a race
4. a video game race is about to begin
5. there is a is a red car is ready to move
6. there is a man is talking about a race
7. a man is explaining a video game before he plays it
8. the formula 1 car i race is ready for the event and the cars are very attractive and big also and it will ready for the future

Figure 2: Example sample frames and label captions from the MSRVTT dataset Xu et al. (2016)

where N is the batch size, T is the sequence length (30), and p is the dimensionality of the last fully connected layer of ResNet 50 (2048).

Next, we used an Encoder-Decoder (Cho et al., 2014) architecture to transform video frame sequences to text summarization. For the encoder part, we used a two-layer recurrent neural network (RNN) with Long Short Term Memory (LSTM) cells (Hochreiter and Schmidhuber, 1997), sandwiching a boundary detector network, to encode the sequence information of the frames. The decoder consists of two fully connected networks to project the encoded frames and word embeddings and a RNN with Gated Recurrent Units (GRUs) (Cho et al., 2014). The GRUs cells output the predicted words at such sequence step.

The pipeline was trained tsinghrough cross entropy losses at each word prediction step, backpropagated from the decoder to the encoder. Note that we did *not* adjust the weights of ResNet 50 using the cross entropy losses, as it only served as a feature extractor for the video frames.

We discuss each part of this pipeline in detail in the next section.

4 Models

4.1 Encoder: LSTM + Boundary Detector

For every input of the forward pass X, we first fed it into the first layer of the stacked LSTM. LSTM cells are defined through four "gates" which represent the short-term and long-term "memory" of how much information to forget and remember from the past. Specifically, the four gates are defined as:

$$
\begin{aligned}
i_t &= \sigma(U^{(i)}x_t + W^{(i)}h_{t-1} + b^{(i)}) \\
f_t &= \sigma(U^{(f)}x_t + W^{(f)}h_{t-1} + b^{(f)}) \\
o_t &= \sigma(U^{(o)}x_t + W^{(o)}h_{t-1} + b^{(o)}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t
\end{aligned}
$$

where $\tilde{c}_t = \tanh(U^{(c)}x_t + W^{(c)}h_{t-1} + b^{(c)})$, and $h_t = o_t \circ \tanh(c_t)$. Each c_t is called a "cell state" and each h_t is called a "hidden state". Assuming the hidden size of our LSTM network is H, we have our output from the first layer LSTM as $N \times T \times H$.

We took the output of every hidden state cell of the first layer, and fed them into the Boundary Detector. The boundary detectors serves as a binary gate which decides if the hidden state from the LSTM should be result before being fed into the next one. The boundary detector is defined as

$$
s_t = \tau(\mathbf{v}_s^T \cdot [W_{si}x_t + W_{sh}h_{t-1} + b_s])
$$

where

$$
\tau(x) = \begin{cases} 1, & \text{if } \sigma(x) > 0.5 \\ 0, & \text{otherwise} \end{cases}
$$

$\mathbf{v_s}, W_{si}, W_{sh}, b_s$ are learned parameters. In its essence, s_t creates a binary mapping of the original hidden state and cell state to either keep or forget them. The input for the second layer LSTM, aka the hidden state and cell state from the previous layer, is in turn defined as

$$
\begin{aligned}
h_{t-1} &= h_{t-1} \cdot (1 - s_t) \\
c_{t-1} &= c_{t-1} \cdot (1 - s_t)
\end{aligned}
$$

We took the *last* hidden output from the second layer LSTM as the encoding of the video frames. Therefore, our encoding has dimensionality $N \times H$.

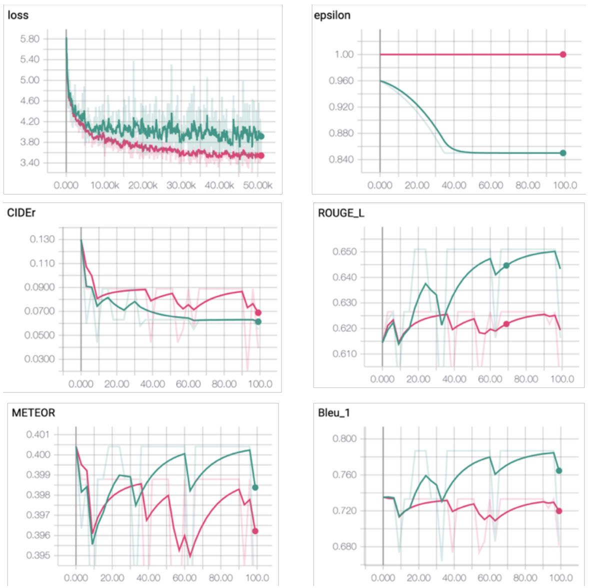

Figure 3: Results using a medium-size training set (25% of MST-VTT), $\arg\max$ sampling for generating captions. Green curves have Teacher Forcing ratio 0.85, and red curves have Teaching Forcing ratio 1.0.

4.2 Decoder: Fully connected layers + GRUs

We first fed the encoded video frame, and word embedding into two fully connected layers respectively, which projects the dimentionality to $N \times P$, where P is the output dimension. Then the output were added together to feed in the GRU cells. The GRU cells, similar to LSTM cells, are defined as

$$
\begin{aligned}
z_t &= \sigma(U^{(z)}x_t + W^{(z)}h_{t-1} + b^{(z)}) \\
r_t &= \sigma(U^{(r)}x_t + W^{(r)}h_{t-1} + b^{(r)}) \\
h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t
\end{aligned}
$$

where $\tilde{h_t} = \tanh(U^{(h)}x_t + W^{(h)}(r_t \circ h_{t-1}) + b^{(h)})$. The actual words are then restored either through the $\arg\max$ of the hidden output at each step, or sampled using a multinomial distribution parameterized by the hidden output as the probabilities. Note that we used scheduled sampling (Bengio et al., 2015) during training period within a minimum ratio.

For a complete summary of the dimension rundown through the pipeline, please refer to Table 1.

5 Experiments and Results

We report our results on the Microsoft Video-to-Text (MSRVTT) video-captioning dataset Xu et al. (2016).

In particular, we noticed that there were significant differences in the output based on the

Section	Component	Output Dimensionality
Feature Extraction	ResNet 50	$N \times T \times p$
Encoding	First Layer LSTM	$N \times T \times H$
	Boundary Detector	$N \times T \times 1$
	Second Layer LSTM	$N \times H$
Decoding	Fully connected - Video Encoding	$N \times P$
	Fully connected - Word Embedding	$N \times P$
	GRU	$N \times$ max words

Table 1: Each component in the processing pipeline and their output dimension. N: batch size; T: time steps (30 frames); p: the output dimension of ResNet 50; H: the hidden size of LSTM; P: the output dimension of the fully connected layers in the decoder; max words: the maximum length of the video caption prediction.

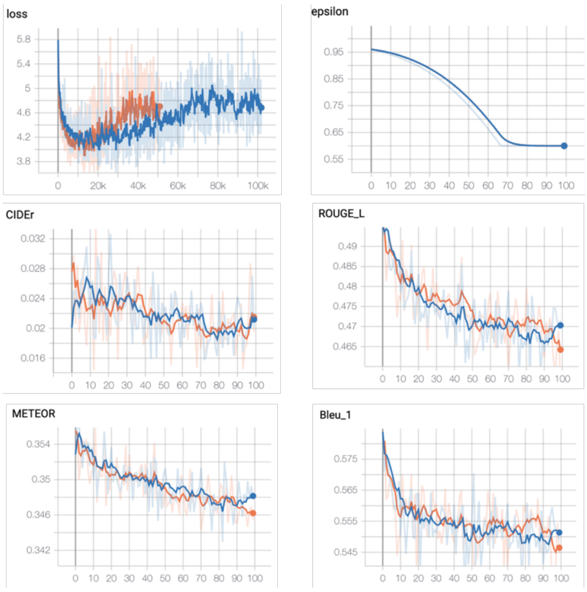

Figure 4: Results using multinomial sampling for generating words, and Teaching Forcing ratio of 0.6. Orange curves use a median-size dataset (25% of MSR-VTT), blue curves use a large-size dataset (50% of MSR-VTT).

sampling strategy strategy, as well as the scheduled sampling teacher-forcing ratio. From Figure 3, we report results on using our "medium-size" dataset, using about 25% of the total MSR-VTT dataset, using $\arg\max$ sampling during caption decoding. Compared to Figures 4 and 5, the quantitative results in terms of evaluation metrics "ROUGE_L", "CIDEr", "METEOR" and "Bleu" are significantly better and with seemingly better generalization, whereas the results from using "multinomial" or "top-k multinomial" sampling. However, this is counter intuitive, as multinomial sampling will produce much more diversified outputs. In fact, from our sample outputs in Figure 6, we notice that the $\arg\max$ sampling strategy consistently outputs a "averaged" output, where a significant portion of the dataset actually consists of labels for where it is simply "a man talking [...]" (See Figure 2). Thus, this actually informs us a significant issue with the MSR-VTT and its relationship with the evaluation metrics, where because a majority of the data samples contain such a general caption such as "there is a man talking about a race" (See Figure 2). As a result, the "multinomial" and "top-k" results, while quantitatively seem significantly worse, produce more specific and nuanced results which deviate the label of "a [wo]man talking about [...]". See more results from Figure 6.

6 Conclusions and Future Work

Overall, from the results we realize that there are two primary directions which simultaneously can significantly improve video summary generation. Firstly, we can tackle the neural architecture as motivated from previous works Baraldi et al. (2016); Guadarrama et al. (2013); Pan et al. (2015a), to better model the sequential nature of

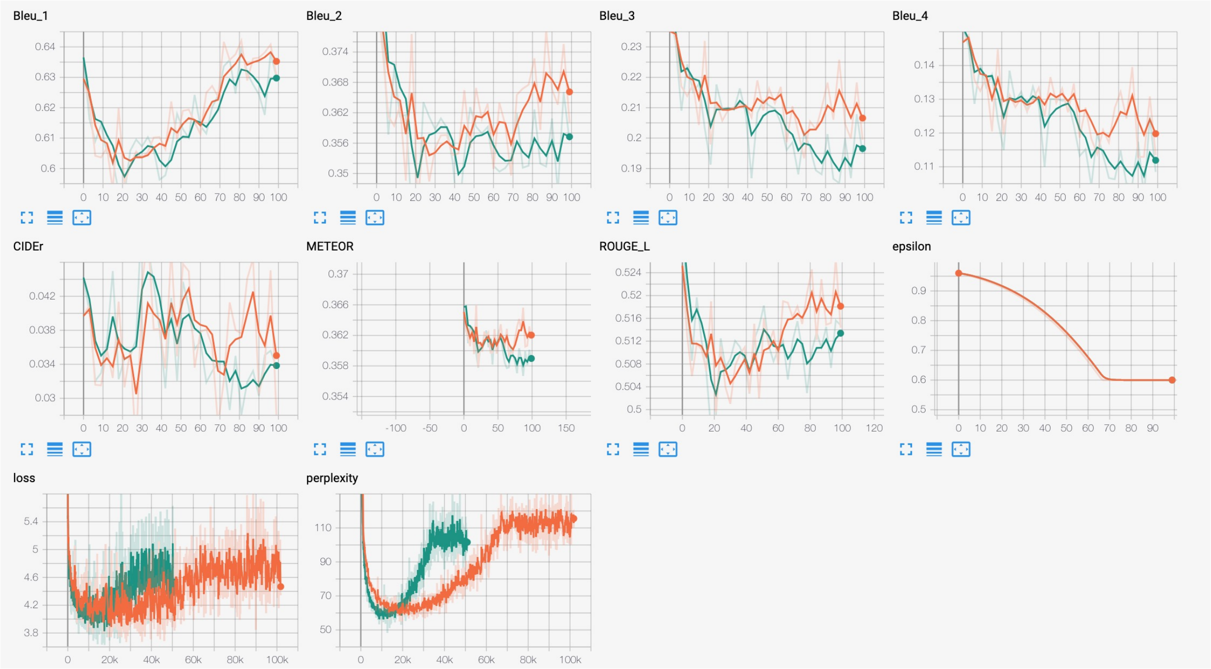

Figure 5: Results using top-k ($k = 20$ here) largest arguments as the multinomial probabilities for sampling to generate words, and Teaching Forcing ratio of 0.6. Green curves use a median-size dataset (25% of MSR-VTT), oragne curves use a large-size dataset (50% of MSR-VTT).

```
ground truth:
<start> a country music video <end>

top_k=20, min_ss=0.6:
<start> a woman is talking about the
    voice <end>

argmax, min_ss=0.85:
<start> a man is playing a song <end>

multinomial, min_ss=0.6:
<start> a scene of the wife s seeing <
    end>
```

Figure 6: Sample Output comparison between $\arg\max$, multinomial, and top-k multinomial sampling for video7586 from the test set.

the video and exploit the context from the underlying temporal visual context, (e.g. attention mechanisms, automatic scene boundary detection). However, ultimately we suspect from our preprocessing pipeline of only using a pre-trained ImageNet Deng et al. (2009) convolutional neural network (e.g. ResNet He et al. (2015)) to encode the features of the video fraem that there lies significant context to be extracted from significantly more semantically powerful and independent encoders. For example, we realized because the video sample from MSR-VTT have many captions which entirely deal in audio, that it would be a very fruitful future work to encode the audio context using both speech recognition encoders, as well as general audio scene classifiers, particularly with music videos with either singing or more culturally significant instrumental background music.

Furthermore, beyond simply using pre-trained ImageNet encoders, we could also encode human action detection using 3D-convolutions (Tran et al., 2015), using pretrained representations from video classification and spatio-temporal networks Qiu et al. (2017); Karpathy et al. (2014).

Lastly, recent works in language modeling have seen significant improvements through the evolution of attention mechanisms applied beyond recurrent neural architectures in the form of Transformer-based networks Devlin et al. (2018); Anonymous (2019), and could similarly enforce syntatically well-formed English captioning during decoding, rather than training the video caption decoder from scratch.

Acknowledgments

We'd like to thank Professor Yejin Choi, and the assistants who helped and contributed greatly towards the quality of our education, Hannah Rashkin, Max Forbes, and Rowan Zellers.

References

2019. The audio description project. http://www.acb.org/adp/.

2019. Media access group. http://main.wgbh.org/wgbh/pages/mag/.

Anonymous. 2019. Improving relation extraction by pre-trained language representations. In *Submitted to Automated Knowledge Base Construction*. Under review.

Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2016. Hierarchical boundary-aware neural encoder for video captioning. *CoRR*, abs/1611.09312.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1171–1179, Cambridge, MA, USA. MIT Press.

Florin Brad and Traian Rebedea. 2017. Neural paraphrase generation using transfer learning. pages 257–261.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2016. Hierarchical multiscale recurrent neural networks. *CoRR*, abs/1609.01704.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond J Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings - 2013 IEEE International Conference on Computer Vision, ICCV 2013*, Proceedings of the IEEE International Conference on Computer Vision, pages 2712–2719, United States. Institute of Electrical and Electronics Engineers Inc.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Andrej Karpathy and Fei-Fei Li. 2014. Deep visual-semantic alignments for generating image descriptions. *CoRR*, abs/1412.2306.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*.

Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond Mooney, Kate Saenko, and Sergio Guadarrama. 2013. Generating natural-language video descriptions using text-mined knowledge. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Tian Lan, Yuke Zhu, Amir Roshan Zamir, and Silvio Savarese. 2015. Action recognition by hierarchical mid-level action elements. *CoRR*, abs/1508.07654.

Zachary Chase Lipton. 2015. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019.

Christopher Olah. 2015. Understanding lstm networks.

Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. 2015a. Hierarchical recurrent neural encoder for video representation with application to captioning. *CoRR*, abs/1511.03476.

Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2015b. Jointly modeling embedding and translation to bridge video and language. *CoRR*, abs/1505.01861.

Hamed Pirsiavash and Deva Ramanan. 2014. Parsing videos of actions with segmental grammars. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA*, pages 612–619.

Zhaofan Qiu, Ting Yao, and Tao Mei. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. *CoRR*, abs/1711.10305.

Hojjat Salehinejad, Julianne Baarbe, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. 2018. Recent advances in recurrent neural networks. *CoRR*, abs/1801.01078.

Alex Sherstinsky. 2018. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314.

Yale Song, Louis-Philippe Morency, and Randall Davis. 2013. Action recognition by hierarchical sequence summarization.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Kevin Tang, Li Fei-Fei, and Daphne Koller. 2012. Learning latent temporal structure for complex event detection.

Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond Mooney. 2014. Integrating language and vision to generate natural language descriptions of videos in the wild. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1218–1227. Dublin City University and Association for Computational Linguistics.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4489–4497, Washington, DC, USA. IEEE Computer Society.

Subhashini Venugopalan, Lisa Anne Hendricks, Raymond J. Mooney, and Kate Saenko. 2016. Improving lstm-based video description with linguistic knowledge mined from text. *CoRR*, abs/1604.01729.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko. 2014. Translating videos to natural language using deep recurrent neural networks. *CoRR*, abs/1412.4729.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2016. Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge. *CoRR*, abs/1609.06647.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. *CoRR*, abs/1608.00859.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing Videos by Exploiting Temporal Structure. *arXiv e-prints*, page arXiv:1502.08029.

Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. 2015. Video paragraph captioning using hierarchical recurrent neural networks. *CoRR*, abs/1510.07712.

Yugnaynehc. 2017. banet. `https://github.com/Yugnaynehc/banet`.